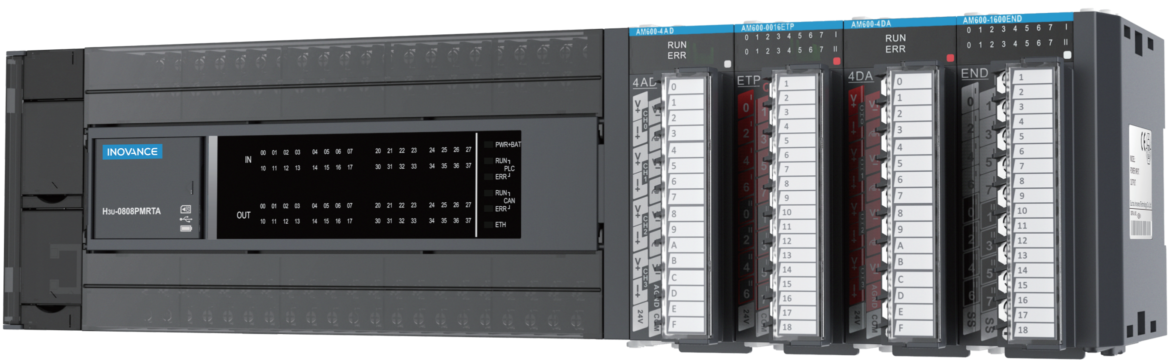


H3U Series

PLC Programming Manual

User Guide

A01

Data code 19010575

Introduction

The H3U series programmable logic controller (PLC) is a third-generation high-performance compact-sized PLC developed by Inovance Technology based on the MCU and FPGA architecture. The product has a high-speed input frequency of 8 x 200 kHz, and a high-speed output frequency of 5 x 200kHz, supporting more and faster high-speed pulse output ports. It supports S-curve acceleration/deceleration, and multiple positioning methods (for example, interrupt positioning and multi-speed positioning). Motion control PLCs support 3 x 500 kHz high-speed differential output, 2- or 3-axis linear interpolation, 2-axis arc interpolation, helix interpolation, and 3-axis electronic cam or G-code input.

The main module has Ethernet communication functions for seamless integration with automation and information. It supports CANlink and CANopen bus for CAN communication and allows easy networking through graphical configuration. It supports USB communication featuring quick commissioning.

Based on the advanced programming environment AutoShop, the product supports programming languages such as ladder chart, instruction list, and sequential function chart (SFC). It supports user programs of up to 64,000 steps and 40,000 power-failure storage word elements. Data and user programs are stored in the flash drive with no battery. It provides a variety of commissioning methods, including online modification and oscilloscope functions.

This product is applicable in the automatic equipment industry, mainly in advanced manufacturing sectors, including the production line automation, wood-working machinery, glass machinery, transportation, loading/ unloading, and electronical customization.

 Inovance will continuously optimize and improve our products, and upgrade this manual accordingly. The manual may be updated without notification. You can visit our website to download the latest version.

Your feedback on this manual is welcomed through any of the following methods.

- ☆ Website: www.inovance.cn
- ☆ Email: plc@inovance.cn

Related Manuals

This manual describes information about the H3U series software, mainly the differences between H2U-XP and H3U, including the scope of elements, special elements, trajectory control, electronic cams, G-code, extension modules, and CANopen.

You can download it from the website www.inovance.cn.

Revision History

Version	Date	Remarks
A00	July 2017	Released the first issue.
A01	May 2019	Updated the cover.

Contents

Instructions Lookup Sheet 1

Note: Instructions in gray background in the table only support H3U-PM motion control models.

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
A	ABSD	Absolute cam control method	√	√	328
	ABST	Absolute position modal instruction	×	√	501
	ACOS	Binary floating-point ARCCOS operation	√	√	124
	ADD	Binary data addition	√	√	102
	ALT	Alternate output	√	√	62
	ANB	Serial connection of circuit blocks	√	√	58
	AND	Serial connection of NO contacts	√	√	50
	AND&	LD logical AND operation	×	√	89
	AND^	AND logical XOR operation	×	√	89
	AND|	AND logical OR operation	×	√	89
	AND<	AND contact comparison smaller than	√	√	86
	AND<=	AND contact comparison smaller than or equal to	√	√	86
	AND<>	AND contact comparison not equal to	√	√	86
	AND=	AND contact comparison equal to	√	√	88
	AND>	AND contact comparison larger than	√	√	6
	AND>=	AND contact comparison larger than or equal to	√	√	86
	ANDF	Serial connection for AND falling edge pulse (F) detection	√	√	86
	ANDP	Serial connection for AND rising edge pulse detection	√	√	50
	ANDZ<	Conductive when absolute value < comparative AND state contact |S1 – S2| < |S3|	×	√	50
	ANDZ<=	Conductive when absolute value <= comparative AND state contact |S1 – S2| ≦ |S3|	×	√	95
	ANDZ<>	Conductive when absolute value <> comparative AND state contact |S1 – S2| ≠ |S3|	×	√	95
	©ANDZ=	Conductive when absolute value = comparative AND state contact |S1 – S2| = |S3|	×	√	95
	ANDZ>	Conductive when absolute value > comparative AND state contact |S1 – S2| > |S3|	×	√	95
	ANDZ>=	Conductive when absolute value >= comparative AND state contact |S1 – S2| ≧ |S3|	×	√	95
	ANI	Serial connection of NC contacts	√	√	95
	ANR	Signal alarm reset	√	√	50
	ANS	Signal alarm setting	√	√	214
	ARWS	Direction switch	√	√	213
	ASC	ASCII code conversion	√	√	327
	ASCI	Conversion from HEX to ASCII format	√	√	322
	ASIN	Binary floating-point ARCSIN operation	√	√	164
	ATAN	Binary floating-point ARCTAN operation	√	√	123
	AXISALMRST	Axis alarm reset	×	√	125
	AXISDRVA	Axis absolute positioning	×	√	294
	AXISENAB	Axis enabling	×	√	288
	AXISESTOP	Axis emergency stop (used for stopping the servo in case of exceptions)	×	√	285
	AXISJOGA	Axis jog	×	√	287
	AXISSTOP	Axis stop positioning	×	√	292
	AXISZRN	Axis return to zero	×	√	286

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
B	BAND	AND bit contact of bit data	√	√	54
	BANI	ANI bit contact of bit data	√	√	55
	BCD	Conversion from binary to BCD format	√	√	150
	BIN	Conversion from BCD to binary format	√	√	151
	BINDA	Conversion from binary format to decimal ASCII format	×	√	157
B	BK-	Matrix subtraction operation	×	√	217
	BK+	Matrix addition operation	×	√	215
	BKCMP<	Matrix smaller than comparison (S1 < S2)	×	√	227
	BKCMP<=	Matrix smaller than or equal to comparison (S1 ≤ S2)	×	√	227
	BKCMP<>	Matrix not equal to comparison (S1 ≠ S2)	×	√	227
	BKCMP=	Matrix equal to comparison (S1 ＝ S2)	×	√	227
	BKCMP>	Matrix larger than comparison (S1 ＞ S2)	×	√	227
	BKCMP>=	Matrix larger than or equal to comparison (S1 ≥ S2)	×	√	227
	BLD	Bit contact of bit data	√	√	52
	BLDI	Inverse bit contact of bit data	√	√	53
	BMOV	Batch data transfer	√	√	172
	BON	ON bit judgment	√	√	209
	BOR	OR bit contact of bit data	√	√	56
	BORI	ORI bit contact of bit data	√	√	57
	BOUT	Bit data output	√	√	63
	BRST	Bit data reset	√	√	65
	BSET	Bit data setting	√	√	64
	BTOW	Data combination by byte	×	√	160
	BZAND	Dead zone control	×	√	135
C	CALL	Call subprogram	√	√	70
	CAMRD	Electronic cam data reading	×	√	580
	CAMWR	Electronic cam data writing	×	√	576
	CANC	Motion compensation cancellation	×	√	521
	CCD	Verification code	√	√	301
	CCW	Counterclockwise arc interpolation	×	√	489
	CJ	Conditional jump	√	√	76
	CJEND	Conditional jump to the program end	√	√	78
	CML	Inverted data transfer	√	√	174
	CMP	Data comparison	√	√	97
	CNTC	Arc center compensation	×	√	519
	COS	Floating-point COS operation	√	√	121
	COSH	Binary floating-point COSH operation	√	√	129
	CRC	CRC verification code calculation	×	√	303
	CW	Clockwise arc interpolation	×	√	489

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
D	DABIN	Conversion from decimal ASCII format to BIN format	×	√	155
	DEC	Binary data decremented by 1	√	√	113
	DECO	Data decoding	√	√	317
	DEG	Binary floating-point radian-to-angle conversion	√	√	127
	DHSOR	High-speed interruption comparison reset	×	√	372
	DHSOS	High-speed interruption comparison setting	×	√	370
	DI	Disable interrupt	√	√	72
	DINTR	Double-speed interruption positioning	×	√	513
	DIS	4-bit separation of 16-bit data	×	√	163
	DIV	Binary data division	√	√	106
	DRV	High-speed positioning	√	√	480
	DRVA	Absolute position positioning	√	√	410
	DRVI	Relative position positioning	√	√	415
	DRVR	Electrical zero return	×	√	511
	DRVZ	Mechanical zero return reset	×	√	506
	DSW	Digital switch	√	√	316
	DSZR	DOG search zero return	√	√	398
	DUTY	Timing pulse generation	×	√	280
	DVIT	Interruption positioning (extension)	×	√	420
E	EADD	Binary floating-point addition	√	√	107
	EBCD	Conversion from binary to decimal floating-point number format	√	√	153
	EBIN	Conversion from decimal to binary floating-point number format	√	√	154
	ECAMRD	Electronic cam floating-point data reading	×	√	581
	ECAMWR	Electronic cam floating-point data writing	×	√	578
	ECMP	Binary floating point comparison	√	√	98
	EDIV	Binary floating-point division	√	√	110
	EI	Enable interrupt	√	√	73
	EMOV	Binary floating point transfer	√	√	170
	EMUL	Binary floating-point multiplication	√	√	109
	ENCO	Data encoding	√	√	318
	END	End of all programs	√	√	68
	ENEG	Binary floating-point notation negation	×	√	119
	ESQR	Binary floating-point square root operation	√	√	145
	ESTR	Conversion from binary floating-point number to string format	×	√	245
	ESUB	Binary floating-point subtraction	√	√	108
	EVAL	Conversion from string to binary floating-point number format	×	√	248
	EXP	Binary floating-point exponent operation	√	√	142
	EZCP	Binary floating point interval comparison	√	√	100

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
F	FAND<	Conductive when floating point number < comparative AND state contact S1 < S2	×	√	92
	FAND<=	Conductive when floating point number <= comparative AND state contact S1 ≦ S2	×	√	92
	FAND<>	Conductive when floating point number <> comparative AND state contact S1 ≠ S2	×	√	92
	FAND=	Conductive when floating point number = comparative AND state contact S1 = S2	×	√	92
	FAND>	Conductive when floating point number > comparative AND state contact S1 > S2	×	√	92
	FAND>=	Conductive when floating point number >= comparative AND state contact S1 ≧ S2	×	√	92
	FDEL	Deletion of data from data tables	×	√	187
	FEND	End of the main program	√	√	66
	FINS	Insertion of data to data tables	×	√	189
	FLD<	Conductive when floating point number < comparative state contact S1 < S2	×	√	91
	FLD<=	Conductive when floating point number <= comparative state contact S1 ≦ S2	×	√	91
	FLD<>	Conductive when floating point number <> comparative state contact S1 ≠ S2	×	√	91
	FLD=	Conductive when floating point number = comparative state contact S1 = S2	×	√	91
	FLD>	Conductive when floating point number > comparative state contact S1 > S2	×	√	91
	FLD>=	Conductive when floating point number >= comparative state contact S1 ≧ S2	×	√	91
	FLT	Conversion from binary number format to binary floating-point number format	√	√	152
	FMOV	One-to-multiple data transfer	√	√	173
	FOR	Start of a loop	√	√	78
	FOR<	Conductive when floating point number < comparative OR state contact S1 < S2	×	√	93
	FOR<=	Conductive when floating point number <= comparative OR state contact S1 ≦ S2	×	√	93
	FOR<>	Conductive when floating point number <> comparative OR state contact S1 ≠ S2	×	√	93
	FOR=	Conductive when floating point number = comparative OR state contact S1 = S2	×	√	93
F	FOR>	Conductive when floating point number > comparative OR state contact S1 > S2	×	√	93
	FOR>=	Conductive when floating point number >= comparative OR state contact S1 ≧ S2	×	√	93

	Instruction	Description	Applicable Model H2U-XP	Applicable Model H3U	Page
G	G00	High-speed positioning	×	√	530
	G01	Linear interpolation	×	√	534
	G02	Clockwise arc interpolation	×	√	538
	G03	Counterclockwise arc interpolation	×	√	538
	G04	Delay waiting	×	√	547
	G17	Selection of the XY-plane modal instruction	×	√	549
	G18	Selection of the ZX-plane modal instruction	×	√	549
	G19	Selection of the YZ-plane modal instruction	×	√	549
	G90	Absolute position modal	×	√	548
	G90G01	2-axis linear absolute position interpolation	×	√	431
	G90G02	2-axis arc-forward absolute position interpolation	×	√	440
	G90G03	2-axis arc-back absolute position interpolation	×	√	452
	G91	Relative position modal	×	√	548
	G91G01	2-axis linear relative position interpolation	×	√	436
	G91G02	2-axis arc-forward relative position interpolation	×	√	446
	G91G03	2-axis arc-back relative position interpolation	×	√	457
	GBIN	Gray code inverse conversion	√	√	333
	GRY	Gray code conversion	√	√	332
H	HEX	Conversion from ASCII to HEX format	√	√	167
	HKY	16-key input	√	√	314
	HOUR	Hour meter	√	√	275
	HSCR	(High-speed counter) Comparison reset	√	√	**362 / 364**
	HSCS	(High-speed counter) Comparison setting	√	√	**358 / 361**
	HSZ	(High-speed counter) Range comparison	√	√	**364 / 369**
	HTOS	Conversion from hours:minutes:seconds format to seconds	×	√	270
I	INC	Binary data incremented by 1	√	√	112
	INCD	Cam control incremental mode	√	√	329
	INCT	Relative position modal instruction	×	√	501
	INSTR	String retrieving	×	√	255
	INT	Conversion from binary floating-point number format to BIN integer format	√	√	149
	INTR	Linear interpolation	√	√	485
	INV	Operation result inversion	√	√	50
	IRET	Interrupt return	√	√	72

	Instruction	Description	Applicable Model H2U-XP	Applicable Model H3U	Page
L	LBL	Marker instruction	√	√	77
	LD	Loading of NO contacts	√	√	50
	LD&	LD logical AND operation	×	√	88
	LD^	LD logical XOR operation	×	√	88
	LD|	LD logical OR operation	×	√	88
	LD<	LD contact comparison smaller than	√	√	88
	LD< =	LD contact comparison smaller than or equal to	√	√	88
	LD<>	LD contact comparison not equal to	√	√	88
	LD=	LD contact comparison equal to	√	√	88
	LD>	LD contact comparison larger than	√	√	88
	LD>=	LD contact comparison larger than or equal to	√	√	88
	LDF	Use of falling edge pulse	√	√	50
	LDI	Loading of NC contacts	√	√	50
	LDP	Use of rising edge pulse	√	√	50
L	LDZ<	Conductive when absolute value < comparative state contact |S1 – S2| < |S3|	×	√	94
	LDZ<=	Conductive when absolute value <= comparative state contact |S1 – S2| ≦ |S3|	×	√	94
	LDZ<>	Conductive when absolute value <> comparative state contact |S1 – S2| ≠ |S3|	×	√	94
	LDZ=	Conductive when absolute value = comparative state contact |S1 – S2| = |S3|	×	√	94
	LDZ>	Conductive when absolute value > comparative state contact |S1 – S2| > |S3|	×	√	94
	LDZ>=	Conductive when absolute value >= comparative state contact |S1 – S2| ≧ |S3|	×	√	94
	LEFT	Start to read from the left of the string	×	√	258
	LEN	Detect the string length	×	√	254
	LIMIT	Upper/Lower limit control	×	√	133
	LIN	Linear interpolation	×	√	485
	LOG	Binary floating-point logarithm operation with a base of 10	√	√	144
	LOGE	Binary floating-point natural logarithm operation	√	√	143
	LRC	LRC verification code calculation	×	√	304

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
M	M02	End of the main program with auxiliary function code 0xxxx	×	√	550
	M30	End of the main program of the auxiliary function code 0xxxx	×	√	550
	M98	Call of the subprogram of the auxiliary function code 0xxxx	×	√	550
	M99	Return of the subprogram of the auxiliary function code 0xxxx	×	√	551
	MADD	(Integer/Floating point) addition operation	×	√	524
	MAND	Matrix AND operation	×	√	218
	MBC	Matrix bit status count operation	×	√	225
	MBR	Matrix bit cyclic replacement operation	×	√	235
	MBRD	Matrix bit read operation	×	√	230
	MBS	Matrix bit shift operation	×	√	233
	MBWR	Matrix bit write operation	×	√	232
	MC	Coil instruction for serial contacts used by the main control	√	√	66
	MCALL	Motion control program call/return instruction	×	√	499
	MCMP	Matrix comparison operation	×	√	228
	MCR	Release instruction for serial contacts used by main control reset	√	√	66
	MDIV	(Integer/Floating point) division operation	×	√	524
	MEAN	Mean value calculation	√	√	132
	MEF	Generation of pulsed operation results			59
	MEP	Generation of pulsed operation results	×	√	59
	MIDR	Any character read from a string	×	√	260
	MIDW	Any character replacement in a string	×	√	262
	MINV	Matrix inverse operation	×	√	224
	MMOV	Data transfer	×	√	524
	MMUL	(Integer/Floating point) multiplication operation	×	√	524
	MODBUS	MODBUS communication	√	√	300
	MOR	Matrix OR operation	×	√	220
	MOV	Value transfer	√	√	169
	MOVC	Linear displacement compensation	×	√	518
	MPP	Stack read	√	√	57
	MPS	Stack-based storage	√	√	57
	MRD	Stack read (current pointer unchanged)	√	√	57
	MRET	Conditional subprogram return	×	√	500
	MRST	Reset	×	√	523
	MSET	Setting	×	√	523
	MSUB	(Integer/Floating point) subtraction operation	×	√	524
M	MTR	Matrix input	√	√	324
	MUL	Binary data multiplication	√	√	105
	MXNR	Matrix XNR operation	×	√	223
	MXOR	Matrix XOR operation	×	√	221
	Mxxxx	Setting function code of the Mxxxx element	×	√	551
	NEG	Binary data negation	√	√	117
	NEXT	End of a loop	√	√	79
	NOP	No action	√	√	68

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
O	OR	Parallel connection of NO contacts	√	√	50
	OR&	OR logical AND operation	×	√	90
	OR^	OR logical XOR operation	×	√	90
	OR|	OR logical OR operation	×	√	90
	OR<	OR contact comparison smaller than	√	√	87
	OR<=	OR contact comparison smaller than or equal to	√	√	87
	OR<>	OR contact comparison not equal to	√	√	87
	OR=	OR contact comparison equal to	√	√	87
	OR>	OR contact comparison larger than	√	√	87
	OR>=	OR contact comparison larger than or equal to	√	√	87
	ORB	Parallel connection of circuit blocks	√	√	58
	ORF	Parallel connection for OR falling edge pulse (F) detection	√	√	50
	ORI	Parallel connection of NC contacts	√	√	50
	ORP	Parallel connection for OR rising edge pulse detection	√	√	50
	ORZ<	Conductive when absolute value < comparative OR state contact |S1 – S2| < |S3|	×	√	96
	ORZ<=	Conductive when absolute value <= comparative OR state contact |S1 – S2| ≦ |S3|	×	√	96
	ORZ<>	Conductive when absolute value <> comparative OR state contact |S1 – S2| ≠ |S3|	×	√	96
	ORZ=	Conductive when absolute value = comparative OR state contact |S1 – S2| = |S3|	×	√	96
	ORZ>	Conductive when absolute value > comparative OR state contact |S1 – S2| > |S3|	×	√	96
	ORZ>=	Conductive when absolute value >= comparative OR state contact |S1 – S2| ≧ |S3|	×	√	96
	OUT	Coil drive	√	√	61
P	PID	PID calculation	√	√	307
	PLF	Falling edge pulse (F) detection coil instruction	√	√	61
	PLS	Rising edge pulse detection coil instruction	√	√	61
	PLSN	Multi-speed pulse output	×	√	426
	PLSR	Pulse output with acceleration/ deceleration	√	√	405
	PLSV	Variable-speed pulse output		√	385
	PLSV2	Variable-speed pulse output with acceleration/deceleration	×	√	389
	PLSY	Pulse output	√	√	380
	POP	Subsequent data read	×	√	191
	POW	Floating-point weight instruction	×	√	147
	PR	ASCII code printing	√	√	323
	PRUN	Octal bit transfer	√	√	326
	PWM	Pulse-width modulation output	√	√	462
R	RAD	Binary floating-point angle -> radian conversion	√	√	126
	RADC	Arc radius compensation	×	√	520
	RAMP	Ramp instruction	√	√	193
	RCL	Carry-included cyclic left-shift	√	√	198
	RCR	Carry-included cyclic right-shift	√	√	197
	REF	Input/output refreshing	√	√	295
	REFF	Input filtering time adjustment	√	√	296

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
R	RET	Program return to the primary bus	√	√	80
	RIGHT	Start to read from the right of the string	×	√	257
	RND	Random data generation	×	√	211
	ROL	Cyclic left-shift	√	√	196
	ROR	Cyclic right-shift	√	√	195
	ROTC	Rotation table control	√	√	331
	RS	Serial data transfer (see the MODBUS instruction)	√	√	298
	RST	Contact or cache clearance	√	√	61
S	S	Auxiliary function code used for setting the rotational speed for the spindle	×	√	552
	SCL	Coordinate determination (coordinate data of different points)	×	√	138
	SCL2	Coordinate determination 2 (X and Y coordinate data)	×	√	140
	SEGD	Seven-segment code decoding	√	√	319
	SEGL	Seven-segment hour-minute display	√	√	320
	SER	Data search	√	√	185
	SET	SET action storage coil instruction	√	√	61
	SETR	Electrical zero setting	×	√	505
	SETT	Current position setting	×	√	503
	SFL	Left-shift of 16-bit data by n bits (carry included)	×	√	207
	SFR	Right-shift of 16-bit data by n bits (carry included)	×	√	205
	SFRD	First in first out data read	√	√	204
	SFTL	Bit left-shift	√	√	200
	SFTR	Bit right-shift	√	√	199
	SFWR	First in first out data write	√	√	203
	SIN	Floating point SIN operation	√	√	120
	SINH	Floating point SINH operation	√	√	128
	SINTR	Single-speed interruption positioning	×	√	513
	SMOV	Shifted transfer	√	√	171
	SORT	Data sorting	√	√	181
	SORT2	Data sorting 2	×	√	183
	SPD	Pulse density detection	√	√	351
	SQR	Binary data square root operation	√	√	146
	SRET	Subprogram return	√	√	71
	SSRET	Conditional subprogram return	×	√	72
	STL	Program jump to the secondary bus	√	√	80
	STMR	Special timer	√	√	278
	STOH	Conversion from seconds to hours:minutes:seconds format	×	√	272
	STR	Conversion from BIN to string format	×	√	237
	SUB	Binary data subtraction	√	√	103
	SUM	Total number of ON bits	√	√	210
	SWAP	Upper/Lower byte exchange	√	√	208

	Instruction	Description	Applicable Model		Page
			H2U-XP	H3U	
T	T	Auxiliary function code	×	√	552
	TADD	Clock data addition operation	√	√	268
	TAN	Floating-point TAN operation	√	√	122
	TANH	Binary floating-point TANH operation	√	√	130
	TCMP	Clock data comparison	√	√	266
	TIM	Delay waiting	×	√	498
	TKY	10-key input	√	√	313
	TRD	Clock data read	√	√	273
	TSUB	Clock data subtraction operation	√	√	269
	TTMR	Training timer	√	√	277
	TWR	Clock data write	√	√	274
	TZCP	Clock data range comparison	√	√	267
U	UNI	4-bit combination of 16-bit data	×	√	162
V	VAL	Conversion from string to BIN format	×	√	241
W	WAND	Binary data logical AND	√	√	114
	WDT	Watchdog timer reset	√	√	69
	WOR	Binary data logical OR	√	√	115
	WSFL	Word left-shift	√	√	202
	WSFR	Word right-shift	√	√	201
	WSUM	Sum of calculated values	×	√	131
	WTOB	Data separation by byte	×	√	159
	WXOR	Binary data logical XOR	√	√	116
X	XCH	Data exchange	√	√	212
	XYP	Setting the XY-plane modal instruction	×	√	502
Y	YZP	Setting the YZ-plane modal instruction	×	√	502
Z	ZCP	Interval comparison	√	√	99
	ZONE	Zone control	×	√	136
	ZPOP	Batch recovery of indexed address registers	×	√	178
	ZPUSH	Batch storage of indexed address registers	×	√	175
	ZRN	Zero return	√	√	394
	ZRST	Full data reset	√	√	180
	ZXP	Setting the ZX-plane modal instruction	×	√	502
	$+	String combination	×	√	252
	$MOV	String transfer	×	√	264

Instructions Lookup Sheet 2

Note: Instructions in gray background in the table only support H3U-PM motion control models.

1 Program Logic Instructions

Type		Model Supported		Page
		H2U-XP	H3U	
Contact Instructions				
LD	Loading of NO contacts	√	√	50
LDI	Loading of NC contacts	√	√	50
AND	Serial connection of NO contacts	√	√	50
ANI	Serial connection of NC contacts	√	√	50
OR	Parallel connection of NO contacts	√	√	50
ORI	Parallel connection of NC contacts	√	√	50
LDP	Use of rising edge pulse	√	√	50
LDF	Use of falling edge pulse	√	√	50
ANDP	Serial connection for AND rising edge pulse detection	√	√	50
ANDF	Serial connection for AND falling edge pulse (F) detection	√	√	50
ORP	Parallel connection for OR rising edge pulse detection	√	√	50
ORF	Parallel connection for OR falling edge pulse (F) detection	√	√	50
INV	Operation result inversion	√	√	50
BLD	Bit contact of bit data	√	√	52
BLDI	Inverse bit contact of bit data	√	√	53
BAND	AND bit contact of bit data	√	√	54
BANI	ANI bit contact of bit data	√	√	55
BOR	OR bit contact of bit data	√	√	56
BORI	ORI bit contact of bit data	√	√	57
Combined instructions				
ANB	Serial connection of circuit blocks	√	√	58
ORB	Parallel connection of circuit blocks	√	√	58
MPS	Stack-based storage	√	√	57
MRD	Stack read (current pointer unchanged)	√	√	57
MPP	Stack read	√	√	57
MEP MEF	Current edge control, generation of pulsed operation results	×	√	58
Output instructions				
OUT	Coil drive	√	√	61
SET	SET action storage coil instruction	√	√	61
RST	Contact or cache clearance	√	√	61
PLS	Rising edge pulse detection coil instruction	√	√	61
PLF	Falling edge pulse (F) detection coil instruction	√	√	61
ALT	Alternate output	√	√	62
BOUT	Bit data output	√	√	63
BSET	Bit data setting	√	√	64
BRST	Bit data reset	√	√	65
Main control instructions				
MC	Coil instruction for serial contacts used by the main control	√	√	66
MCR	Release instruction for serial contacts used by main control reset	√	√	66
End instructions				
FEND	End of the main program	√	√	67
END	End of all programs	√	√	68
Other processing instructions				
NOP	No action	√	√	68
WDT	Watchdog timer reset	√	√	69

2 Program Flow Instructions

Type		Model Supported		Page
		H2U-XP	H3U	
Subprogram				
CALL	Call subprogram	√	√	70
SRET	Subprogram return	√	√	71
SSRET	Conditional subprogram return	×	√	72
IRET	Interrupt return	√	√	72
Interrupt				
EI	Enable interrupt	√	√	73
DI	Disable interrupt	√	√	73
Jump				
CJ	Conditional jump	√	√	76
LBL	Marker instruction	√	√	77
CJEND	Conditional jump to the program end	√	√	78
Loop				
FOR	Start of a loop	√	√	78
NEXT	End of a loop	√	√	79
Step-by-step sequential control				
STL	Program jump to the secondary bus	√	√	80
RET	Program return to the primary bus	√	√	80

3 Data Comparison

Type		Model Supported		Page
		H2U-XP	H3U	
Contact comparison				
LD=	LD contact comparison equal to	√	√	84
LD>	LD contact comparison larger than	√	√	84
LD<	LD contact comparison smaller than	√	√	84
LD<>	LD contact comparison not equal to	√	√	84

Type		Model Supported		Page
		H2U-XP	H3U	
LD>=	LD contact comparison larger than or equal to	√	√	84
LD<=	LD contact comparison smaller than or equal to	√	√	84
AND=	AND contact comparison equal to	√	√	86
AND>	AND contact comparison larger than	√	√	86
AND<	AND contact comparison smaller than	√	√	86
AND<>	AND contact comparison not equal to	√	√	86
AND>=	AND contact comparison larger than or equal to	√	√	86
AND< =	AND contact comparison smaller than or equal to	√	√	
OR=	OR contact comparison equal to	√	√	87
OR>	OR contact comparison larger than	√	√	87
OR<	OR contact comparison smaller than	√	√	87
OR<>	OR contact comparison not equal to	√	√	87
OR>=	OR contact comparison larger than or equal to	√	√	87
OR<=	OR contact comparison smaller than or equal to	√	√	87
LD&	LD logical AND operation	×	√	88
LD|	LD logical OR operation	×	√	88
LD^	LD logical XOR operation	×	√	883
AND&	LD logical AND operation	×	√	89
AND|	AND logical OR operation	×	√	89
AND^	AND logical XOR operation	×	√	89
OR&	OR logical AND operation	×	√	90
OR|	OR logical OR operation	×	√	90
OR^	OR logical XOR operation	×	√	90
FLD>	Conductive when floating point number > comparative state contact S1 > S2	×	√	91
FLD>=	Conductive when floating point number >= comparative state contact S1 ≧ S2	×	√	91
FLD <	Conductive when floating point number < comparative state contact S1 < S2	×	√	91
FLD<=	Conductive when floating point number <= comparative state contact S1 ≦ S2	×	√	91
FLD =	Conductive when floating point number = comparative state contact S1 = S2	×	√	91
FLD<>	Conductive when floating point number <> comparative state contact S1 ≠ S2	×	√	91

Type		Model Supported		Page
		H2U-XP	H3U	
FAND >	Conductive when floating point number > comparative AND state contact S1 > S2	×	√	92
FAND>=	Conductive when floating point number >= comparative AND state contact S1 ≧ S2	×	√	92
FAND<	Conductive when floating point number < comparative AND state contact S1 < S2	×	√	92
FAND<=	Conductive when floating point number <= comparative AND state contact S1 ≦ S2	×	√	92
FAND=	Conductive when floating point number = comparative AND state contact S1 = S2	×	√	92
FAND<>	Conductive when floating point number <> comparative AND state contact S1 ≠ S2	×	√	92
FOR>	Conductive when floating point number > comparative OR state contact S1 > S2	×	√	93
FOR>=	Conductive when floating point number >= comparative OR state contact S1 ≧ S2	×	√	93
FOR<	Conductive when floating point number < comparative OR state contact S1 < S2	×	√	93
FOR<=	Conductive when floating point number <= comparative OR state contact S1 ≦ S2	×	√	93
FOR =	Conductive when floating point number = comparative OR state contact S1 = S2	×	√	93
FOR<>	Conductive when floating point number <> comparative OR state contact S1 ≠ S2	×	√	93
LDZ>	Conductive when absolute value > comparative state contact |S1 – S2| > |S3|	×	√	94
LDZ>=	Conductive when absolute value >= comparative state contact |S1 – S2| ≧ |S3|	×	√	94
LDZ<	Conductive when absolute value < comparative state contact |S1 – S2| < |S3|	×	√	94
LDZ<=	Conductive when absolute value <= comparative state contact |S1 – S2| ≦ |S3|	×	√	94
LDZ=	Conductive when absolute value = comparative state contact |S1 – S2| = |S3|	×	√	94
LDZ<>	Conductive when absolute value <> comparative state contact |S1 – S2| ≠ |S3|	×	√	94
ANDZ>	Conductive when absolute value > comparative AND state contact |S1 – S2| > |S3|	×	√	95

Type		Model Supported		Page
		H2U-XP	H3U	
ANDZ >=	Conductive when absolute value >= comparative AND state contact $\lvert S1 - S2\rvert \geqq \lvert S3\rvert$	×	√	95
ANDZ<	Conductive when absolute value < comparative AND state contact $\lvert S1 - S2\rvert < \lvert S3\rvert$	×	√	95
ANDZ<=	Conductive when absolute value <= comparative AND state contact $\lvert S1 - S2\rvert \leqq \lvert S3\rvert$	×	√	95
©ANDZ=	Conductive when absolute value = comparative AND state contact $\lvert S1 - S2\rvert = \lvert S3\rvert$	×	√	95
ANDZ<>	Conductive when absolute value <> comparative AND state contact $\lvert S1 - S2\rvert \neq \lvert S3\rvert$	×	√	95
ORZ>	Conductive when absolute value > comparative OR state contact $\lvert S1 - S2\rvert > \lvert S3\rvert$	×	√	96
ORZ>=	Conductive when absolute value >= comparative OR state contact $\lvert S1 - S2\rvert \geqq \lvert S3\rvert$	×	√	96
ORZ<	Conductive when absolute value < comparative OR state contact $\lvert S1 - S2\rvert < \lvert S3\rvert$	×	√	96
ORZ<=	Conductive when absolute value <= comparative OR state contact $\lvert S1 - S2\rvert \leqq \lvert S3\rvert$	×	√	96
ORZ=	Conductive when absolute value = comparative OR state contact $\lvert S1 - S2\rvert = \lvert S3\rvert$	×	√	96
ORZ<>	Conductive when absolute value <> comparative OR state contact $\lvert S1 - S2\rvert \neq \lvert S3\rvert$	×	√	96
Comparative output				
CMP	Data comparison	√	√	97
ECMP	Binary floating point comparison	√	√	98
ZCP	Interval comparison	√	√	99
EZCP	Binary floating point interval comparison	√	√	100

4 Data Operation

Type		Model Supported		Page
		H2U-XP	H3U	
Four arithmetic operations				
ADD	Binary data addition	√	√	102
SUB	Binary data subtraction	√	√	103
MUL	Binary data multiplication	√	√	105
DIV	Binary data division	√	√	106
EADD	Binary floating-point addition	√	√	107
ESUB	Binary floating-point subtraction	√	√	108
EMUL	Binary floating-point multiplication	√	√	109
EDIV	Binary floating-point division	√	√	110
INC	Binary data incremented by 1	√	√	112
DEC	Binary data decremented by 1	√	√	113

Type		Model Supported		Page
		H2U-XP	H3U	
Data logical operations				
WAND	Binary data logical AND	√	√	114
WOR	Binary data logical OR	√	√	115
WXOR	Binary data logical XOR	√	√	116
NEG	Binary data negation	√	√	117
ENEG	Binary floating-point notation negation	×	√	119
Trigonometric functions				
SIN	Floating-point SIN operation	√	√	120
COS	Floating-point COS operation	√	√	121
TAN	Floating-point TAN operation	√	√	122
ASIN	Binary floating-point ARCSIN operation	√	√	123
ACOS	Binary floating-point ARCCOS operation	√	√	124
ATAN	Binary floating-point ARCTAN operation	√	√	125
RAD	Binary floating-point angle -> radian conversion	√	√	126
DEG	Binary floating-point radian-to-angle conversion	√	√	127
SINH	Binary floating-point SINH operation	√	√	128
COSH	Binary floating-point COSH operation	√	√	129
TANH	Binary floating-point TANH operation	√	√	130
Table operations				
WSUM	Sum of calculated values	×	√	131
MEAN	Mean value calculation	√	√	132
LIMIT	Upper/Lower limit control	×	√	133
BZAND	Dead zone control	×	√	135
ZONE	Zone control	×	√	136
SCL	Coordinate determination (coordinate data of different points)	×	√	138
SCL2	Coordinate determination 2 (X and Y coordinate data)	×	√	140
Exponent operation				
EXP	Binary floating-point exponent operation	√	√	142
LOGE	Binary floating-point natural logarithm operation	√	√	143
LOG	Binary floating-point logarithm operation with a base of 10	√	√	144
ESQR	Binary floating-point square root operation	√	√	145
SQR	Binary data square root operation	√	√	146
POW	Floating-point weight instruction	×	√	147

5 Data Processing

Type		Model Supported H2U-XP	H3U	Page
Data conversion				
INT	Conversion from binary floating-point number format to BIN integer format	√	√	149
BCD	Conversion from binary to BCD format	√	√	150
BIN	Conversion from BCD to binary format	√	√	151
FLT	Conversion from binary number format to binary floating-point number format	√	√	152
EBCD	Conversion from binary to decimal floating-point number format	√	√	153
EBIN	Conversion from decimal to binary floating-point number format	√	√	154
DABIN	Conversion from decimal ASCII format to BIN format	×	√	155
BINDA	Conversion from binary format to decimal ASCII format	×	√	157
WTOB	Data separation by byte	×	√	159
BTOW	Data combination by byte	×	√	160
UNI	4-bit combination of 16-bit data	×	√	162
DIS	4-bit separation of 16-bit data	×	√	163
ASCI	Conversion from HEX to ASCII format	√	√	164
HEX	Conversion from ASCII to HEX format	√	√	167
Data transfer				
MOV	Value transfer	√	√	169
EMOV	Binary floating point transfer	√	√	170
SMOV	Shifted transfer	√	√	171
BMOV	Batch data transfer	√	√	172
FMOV	One-to-multiple data transfer	√	√	173
CML	Inverted data transfer	√	√	174
ZPUSH	Batch storage of indexed address registers	×	√	175
ZPOP	Batch recovery of indexed address registers	×	√	178
Table operations				
ZRST	Full data reset	√	√	180
SORT	Data sorting	√	√	181
SORT2	Data sorting 2	×	√	183
SER	Data search	√	√	185
FDEL	Deletion of data from data tables	×	√	187
FINS	Insertion of data to data tables	×	√	189
POP	Subsequent data read	×	√	191
RAMP	Ramp instruction	√	√	193
Data shift				
ROR	Cyclic right-shift	√	√	195
ROL	Cyclic left-shift	√	√	196
RCR	Carry-included cyclic right-shift	√	√	197
RCL	Carry-included cyclic left-shift	√	√	198
SFTR	Bit right-shift	√	√	199
SFTL	Bit left-shift	√	√	200
WSFR	Word right-shift	√	√	201
WSFL	Word left-shift	√	√	202
SFWR	First in first out data write	√	√	203
SFRD	First in first out data read	√	√	204
SFR	Right-shift of 16-bit data by n bits (carry included)	×	√	205
SFL	Left-shift of 16-bit data by n bits (carry included)	×	√	207
Other data processing				
SWAP	Upper/Lower byte exchange	√	√	208
BON	ON bit judgment	√	√	209
SUM	Total number of ON bits	√	√	210
RND	Random data generation	×	√	211
XCH	Data exchange	√	√	212
ANS	Signal alarm setting	√	√	213
ANR	Signal alarm reset	√	√	214

6 Matrix Instructions

Type		Model Supported H2U-XP	H3U	Page
Matrix operation				
BK+	Matrix addition operation	×	√	215
BK-	Matrix subtraction operation	×	√	217
MAND	Matrix AND operation	×	√	218
MOR	Matrix OR operation	×	√	220
MXOR	Matrix XOR operation	×	√	221
MXNR	Matrix XNR operation	×	√	223
MINV	Matrix inverse operation	×	√	224
MBC	Matrix bit status count operation	×	√	225
Matrix comparison				
BKCMP=	Matrix equal to comparison (S1 = S2)	×	√	227
BKCMP>	Matrix larger than comparison (S1 > S2)	×	√	227
BKCMP<	Matrix smaller than comparison (S1 < S2)	×	√	227
BKCMP<>	Matrix not equal to comparison (S1 ≠ S2)	×	√	227
BKCMP<=	Matrix smaller than or equal to comparison (S1 ≤ S2)	×	√	227
BKCMP>=	Matrix larger than or equal to comparison (S1 ≥ S2)	×	√	227
MCMP	Matrix comparison operation	×	√	228

Type		Model Supported		Page
		H2U-XP	H3U	
Matrix read/write				
MBRD	Matrix bit read operation	×	√	230
MBWR	Matrix bit write operation	×	√	232
Matrix replacement				
MBS	Matrix bit shift operation	×	√	233
MBR	Matrix bit cyclic replacement operation	×	√	235

7 String Instructions

Type		Model Supported		Page
		H2U-XP	H3U	
STR	Conversion from BIN to string format	×	√	237
VAL	Conversion from string to BIN format	×	√	241
ESTR	Conversion from binary floating-point number to string format	×	√	245
EVAL	Conversion from string to binary floating-point number format	×	√	248
$+	String combination	×	√	252
LEN	Detect the string length	×	√	254
INSTR	String retrieving	×	√	255
RIGHT	Start to read from the right of the string	×	√	257
LEFT	Start to read from the left of the string	×	√	258
MIDR	Any character read from a string	×	√	260
MIDW	Any character replacement in a string	×	√	262
$MOV	String transfer	×	√	264

8 Clock Instructions

Type		Model Supported		Page
		H2U-XP	H3U	
Clock comparative output				
TCMP	Clock data comparison	√	√	266
TZCP	Clock data range comparison	√	√	267
Clock operation				
TADD	Clock data addition operation	√	√	268
TSUB	Clock data subtraction operation	√	√	269
Clock conversion				
HTOS	Conversion from hours:minutes:seconds format to seconds	×	√	270
STOH	Conversion from seconds to hours:minutes:seconds format	×	√	272
Clock read/write				
TRD	Clock data read	√	√	273
TWR	Clock data write	√	√	274
Timing				
HOUR	Hour meter	√	√	275
TTMR	Training timer	√	√	277
STMR	Special timer	√	√	278
DUTY	Timing pulse generation	×	√	280

9 High-speed Input, Pulse Positioning, and Communication Positioning

Type		Model Supported		Page
		H2U-XP	H3U	
High-speed comparison				
HSCS	(High-speed counter) comparison setting	√	√	358
HSCR	(High-speed counter) Comparison reset	√	√	362
HSZ	(High-speed counter) Range comparison	√	√	364
DHSOS	High-speed interruption comparison setting	×	√	370
DHSOR	High-speed interruption comparison reset	×	√	372
Pulse input				
SPD	Pulse density detection	√	√	351
Pulse output				
PWM	Pulse-width modulation output	√	√	462
PLSY	Pulse output	√	√	380
PLSR	Pulse output with acceleration/deceleration	√	√	405
Pulse positioning				
PLSV	Variable-speed pulse output		√	385
PLSV2	Variable-speed pulse output with acceleration/deceleration	×	√	389
PLSN	Multi-speed pulse output	×	√	426
DVIT	Interruption positioning (extension)	×	√	420
DRVI	Relative position positioning	√	√	415
DRVA	Absolute position positioning	√	√	410
ZRN	Zero return	√	√	394
DSZR	DOG search zero return	√	√	398

Type		Model Supported		Page
		H2U-XP	H3U	
Communication positioning (robot)				
AXISENAB	Axis enabling	×	√	285
AXISSTOP	Axis stop positioning	×	√	286
AXISESTOP	Axis emergency stop (used for stopping the servo in case of exceptions)	×	√	287
AXISDRVA	Axis absolute positioning	×	√	288
AXISZRN	Axis return to zero	×	√	290
AXISJOGA	Axis jog	×	√	292
AXISALMRST	Axis alarm reset instruction	×	√	294
Refreshing				
REF	Input/output refreshing	√	√	295
REFF	Input filtering time adjustment	√	√	296

10 Motion Control

Type		Model Supported		Page
		H2U-XP	H3U	
Interpolation for the H3U model				
G90G01	2-axis linear absolute position interpolation	×	√	431
G91G01	2-axis linear relative position interpolation	×	√	436
G90G02	2-axis arc-forward absolute position interpolation	×	√	440
G91G02	2-axis arc-forward relative position interpolation	×	√	446
G90G03	2-axis arc-back absolute position interpolation	×	√	452
G91G03	2-axis arc-back relative position interpolation	×	√	457
Instructions for MC-motion control in the PM model				
DRV	High-speed positioning	√	√	480
LIN	Linear interpolation	×	√	485
INTR	Linear interpolation	√	√	485
CW	Clockwise arc interpolation	×	√	489
CCW	Counterclockwise arc interpolation	×	√	489
TIM	Delay waiting	×	√	498
MCALL	Call the subprogram	×	√	499
MRET	Conditional subprogram return	×	√	500
ABST	Absolute position modal instruction	×	√	501
INCT	Relative position modal instruction	×	√	501
XYP	Setting the XY-plane modal instruction	×	√	502
YZP	Setting the YZ-plane modal instruction	×	√	502
ZXP	Setting the ZX-plane modal instruction	×	√	502
SETT	Current position setting	×	√	503

Type		Model Supported		Page
		H2U-XP	H3U	
SETR	Electrical zero setting	×	√	505
DRVZ	Mechanical zero return reset	×	√	506
DRVR	Electrical zero return	×	√	511
SINTR	Single-speed interruption positioning	×	√	493
DINTR	Double-speed interruption positioning	×	√	493
MOVC	Linear displacement compensation	×	√	518
CNTC	Arc center compensation	×	√	519
RADC	Arc radius compensation	×	√	520
CANC	Motion compensation cancellation	×	√	521
Other MC- instructions for the PM model				
MSET	Setting	×	√	523
MRST	Reset	×	√	523
MMOV	Data transfer	×	√	524
MADD	(Integer/Floating point) addition operation	×	√	524
MSUB	(Integer/Floating point) subtraction operation	×	√	524
MMUL	(Integer/Floating point) multiplication operation	×	√	524
MDIV	(Integer/Floating point) division operation	×	√	524
G-code motion control instructions for the PM model				
G00	High-speed positioning; moving to the specified position at the highest speed set. The three axises run separately.	×	√	530
G01	Linear interpolation	×	√	534
G02	Clockwise arc interpolation	×	√	538
G03	Counterclockwise arc interpolation	×	√	538
G04	Delay waiting	×	√	547
G90	Absolute position modal	×	√	548
G91	Relative position modal	×	√	548
G17	Selection of the XY-plane modal instruction	×	√	549
G18	Selection of the ZX-plane modal instruction	×	√	549
G19	Selection of the YZ-plane modal instruction	×	√	549
M02	End of the main program with auxiliary function code 0xxxx	×	√	550
M30	End of the main program with auxiliary function code 0xxxx	×	√	550
M98	Call of the subprogram of the auxiliary function code 0xxxx	×	√	550
M99	Return of the subprogram of the auxiliary function code 0xxxx	×	√	551
Mxxxx	Setting function code of the Mxxxx element	×	√	551

Type		Model Supported		Page
		H2U-XP	H3U	
S	Auxiliary function code used for setting the rotational speed for the spindle	×	√	552
T	Auxiliary function code used for selecting the tool	×	√	552

11 Communication

Type		Model Supported		Page
		H2U-XP	H3U	
Communication instructions				
RS	Serial data transfer (see the MODBUS instruction)	√	√	298
MODBUS	MODBUS communication (see the MODBUS instructions)	√	√	300
Verification				
CCD	Verification code	√	√	301
CRC	CRC verification code calculation	×	√	303
LRC	LRC verification code calculation	×	√	304

12 Peripheral Instructions

Type		Model Supported		Page
		H2U-XP	H3U	
PID calculation				
PID	PID calculation	√	√	307
Bit switch access				
TKY	10-key input	√	√	313
HKY	16-key input	√	√	314
DSW	Digital switch	√	√	316
DECO	Data decoding	√	√	317
ENCO	Data encoding	√	√	318
LED				
SEGD	Seven-segment code decoding	√	√	319
SEGL	Seven-segment hour-minute display	√	√	320
Other peripheral instructions				
ASC	ASCII code conversion	√	√	322
PR	ASCII code printing	√	√	323
MTR	Matrix input	√	√	324
PRUN	Octal bit transfer	√	√	326
ARWS	Direction switch	√	√	327
ABSD	Absolute cam control method	√	√	328
INCD	Cam control incremental mode	√	√	329
ROTC	Rotation table control	√	√	331
GRY	Gray code conversion	√	√	332
GBIN	Gray code inverse conversion	√	√	333

13 Electronic Cam Instructions

Type		Model Supported		Page
		H2U-XP	H3U	
CAMRD	Electronic cam data reading	×	√	580
CAMWR	Electronic cam data writing	×	√	576
ECAMRD	Electronic cam floating-point data reading	×	√	581
ECAMWR	Electronic cam floating-point data writing	×	√	578

Guide on Quick Reference

The following table is for your quick reference if you have doubts on any of the following items:

No.	Item	Page
1	High-speed counter and input interrupt	See “5.1 H3U Standard Model” on page 336 and “5.3 H3U-PM Motion Control Model” on page 353
2	Use of high-speed comparison instructions	See “5.2 High-speed Pulse Comparison Instructions of H3U Standard Model” on page 341 and “5.4 High-speed Comparison Instructions for H3U-PM Motion Control Model” on page 357
3	Use of special elements with high-speed output	See “6.1 Overview” on page 375
4	High-speed output positioning instructions	See “6.2 Positioning Instruction” on page 379
5	Interpolation instructions for standard models	See “6.3 Interpolation Instruction” on page 430
6	Introduction to subprograms for motion control models	See “7.3 Execution and Call of Motion Control Subprogram” on page 468
7	Use of motion control instructions	See “7.7 Format and Use of MC Subprograms” on page 476
8	Use of motion control G-code instructions	See “7.9 Format and Use of G-code Subprograms” on page 527
9	Use of the electronic cam	See “Chapter 8 Electronic Cam” on page 563
10	Introduction to MODBUS communication and protocols	See “9.4 Modbus Protocol” on page 598 and “9.5 Modbus Configuration and Usage” on page 613
11	CANlink communication	See “9.6 CANlink Communication” on page 616
12	CANopen communication	See “9.7 CANopen Communication” on page 635
13	Ethernet communication	See “9.8 Ethernet Communication” on page 651
14	Application of extension modules	See “Chapter 10 Extension Modules” on page 659
15	Introduction to interrupts	See “Chapter 11 Interrupt” on page 686
16	Introduction to subprograms	See “Chapter 12 Subprogram” on page 694
17	Description on special relays and registers	See “Appendix A Allocation of Soft Elements SM, SD, D8000, and M8000” on page 704

1 Overview

1

Chapter 1 Overview

1.1 Introduction to the H3U Series PLC

The H3U series PLC is a third-generation high-performance PLC developed by Inovance. Thanks to the latest industrial-strength CPU and FPGA hardware structure and embedded software with independent intelligent property rights, the product performance and capacity has been substantially improved. In addition, it provides a wide range of functions including positioning, trajectory tacking control, and network communication. The series consists of the H3U standard model and the H3U-PM motion control model.

Main performance specifications of the H3U standard model and the H3U-PM motion control model are introduced below:

Item	H3U Standard Model	H3U-PM Motion Control Model
Program capacity	64K	
Basic instruction rate	100ns	
High-speed input	200K (8 inputs)	200K (3 inputs)[1]
High-speed output[2]	200K (5 inputs)	500K (3 inputs)[3]
Storage capacity in case of a power failure	48K	
COM serial communication	COM0: RS422 COM1: RS485	
CAN communication	Equipped with a CAN communication port CANlink (up to 256 items for a primary station, and up to 16 items for a secondary station) CANopen (up to 64 items for a primary station, and up to 8 items for a secondary station)	
Ethernet communication	Equipped with an Ethernet port Program uploading/ downloading, and MODBUS TCP Up to 16 links	Equipped with an Ethernet port Program uploading/downloading, and MODBUS TCP Up to 8 links
USB	Supported	
Expansion module type	AM600 extension module, CANlink remote module, and AM600 remote module[4]	
Interpolation	2-axis arc and 2-axis straight line	2-axis arc, 3-axis straight line and spiral line[5]
Positioning instruction	Multiple positioning types added	
S-curve acceleration/ deceleration	Supported	
Special element	SM, SD and R added	
Subprogram with parameters	FC supported	
Electronic cam	Not supported	3-axis electronic cam[6]
Motion control and G-code	Not supported	Supported

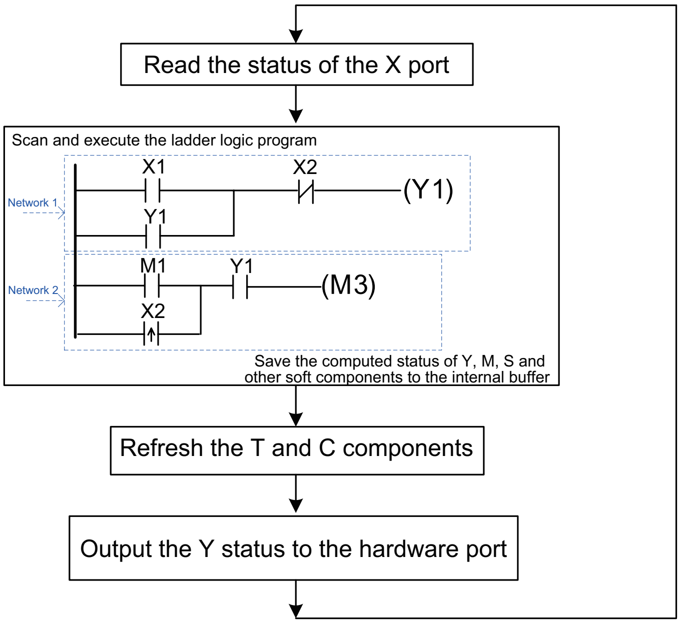
NOTE

- [1]: The three channels of AB phase counters support differential or single-ended inputs.
- [2] High-speed outputs are only applicable to models with transistor outputs. H3U standard models use open-drain outputs. Motion models use differential outputs.
- [3]: 3-axis output is defined. Each axis includes two groups of differential outputs, which can serve as AB phase outputs in the CW/CCW output format or pulse plus direction mode.
- [4]: The H3U local extension module does not support the H2U-XP extension module, but supports the AM600 local extension module.
- [5]: For the motion control functions of the H3U-PM model, see "Chapter 7 Motion Control" on page 466.
- [6]: For application of the electronic cam of the H3U-PM model, see "Chapter 8 Electronic Cam" on page 563.

1.2 PLC Operating Principle

After a programmer downloads a designed and compiled ladder chart program to the PLC memory, the PLC scans the user program.

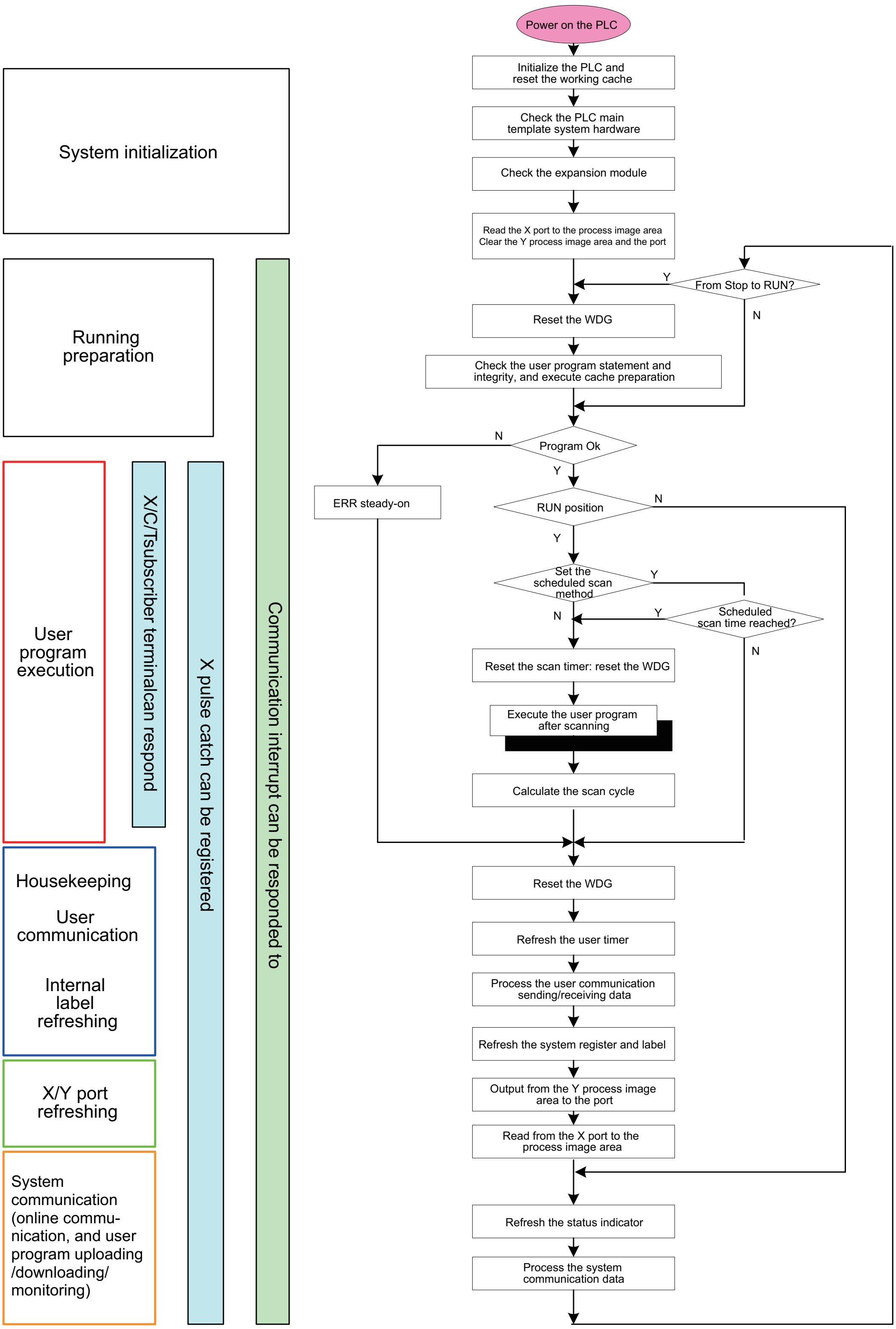
When running, the PLC mainly detects the X input, scans and operates the user program, refreshes the status of other elements, and outputs the Y cache status to the Y hardware port in cycles. User program scanning is the core task. The process is shown in the following figure.



Before executing a user program, the PLC reads and stores the X hardware port status to the X variable buffer.

The user program is scanned based on network blocks. A network block is a group of elements connected through wires. See the two networks in the preceding figure. The calculation is executed from the first network to the last one. The "contact" statuses of elements in each network are logically calculated and synthesized one after another from the left to the right. The PLC outputs the result to the "coil" of the element, or determines whether to perform an operation based on the logic.

In the ladder chart, the part on the left now serves as a "live wire" with the default (potential) state of ON. Each time an element is passed, the logic operation result transitory status is refreshed. The intermediate operation result transitory status is sometimes called a "flow". The intermediate logic operation result status is ON; that is, the "flow" is valid. The output status of this network is the status of the flow that outputs electricity. If the right most indicates an operation and the flow is valid, the operation is performed; otherwise, the operation is not performed.


System initialization
Running preparation
User program execution
X/C/Tsubscriber terminalcan respond
X pulse catch can be registered
Communication interrupt can be responded to
Housekeeping
User communication
Internal label refreshing
X/Y port refreshing
System communication (online communication, and user program uploading /downloading/ monitoring)
Power on the PLC
Initialize the PLC and reset the working cache
Check the PLC main template system hardware
Check the expansion module
Read the X port to the process image area Clear the Y process image area and the port
From Stop to RUN?
Y
N
Reset the WDG
Check the user program statement and integrity, and execute cache preparation
Program Ok
N
Y
ERR steady-on
RUN position
N
Y
Set the scheduled scan method
Y
N
Scheduled scan time reached?
Y
N
Reset the scan timer: reset the WDG
Execute the user program after scanning
Calculate the scan cycle
Reset the WDG
Refresh the user timer
Process the user communication sending/receiving data
Refresh the system register and label
Output from the Y process image area to the port
Read from the X port to the process image area
Refresh the status indicator
Process the system communication data

After all the networks of the main program are scanned from top to the bottom, all timers are refreshed, and routine communication and other data are processed, the PLC system program outputs the status of the variable in the Y register buffer to the Y hardware port. Then the PLC starts the user program scanning again until the "RUN/STOP" switch for controlling execution of user programs is toggled to STOP.

In addition, running preparation, system communication, and interrupt processing shall be finished for the PLC system software. The system software running process is shown in the preceding figure. When scanning a complex user program, the system can use the "interrupt" processing method to respond to the "user interrupt" signal to timely process important signals (also called important "events").

"Interrupt" processing means that, after detecting a specific signal, the CPU immediately stops (or interrupts) the current routine action, executes the specific subprogram, and resumes the routine action which is previously stopped after the subprogram is executed. It is a main characteristic of the "interrupt" function that the interrupt signal request can receivie timely response and processing.

In the PLC, interrupts are divided into user interrupts (interrupt of high-speed signal input (X0~X7), high-speed counting, and timing) and communication interrupts (including system communication and communication launched by user programs). In the PLC, the priorities of all interrupts are the same but their allowable intervals are slightly different (see the preceding illustration).

Memo NO. ____________

Date / /

2 Pre-programming Precautions

Chapter 2 Pre-programming Precautions

2.1 Programming Software

The H3U series PLC uses AutoShop V2.50 or later versions. The minimum configuration of AutoShop is as follows:

Operating system: Windows XP, Windows 7, Windows 8, or Windows 10

Memory: 1 GB or above

Hard disk: 1 GB or above available space

CPU: Intel i3 or later versions or AMD-equivalent CPUs

You can download AutoShop from the website www.inovance.cn.

2.2 Communication Cable

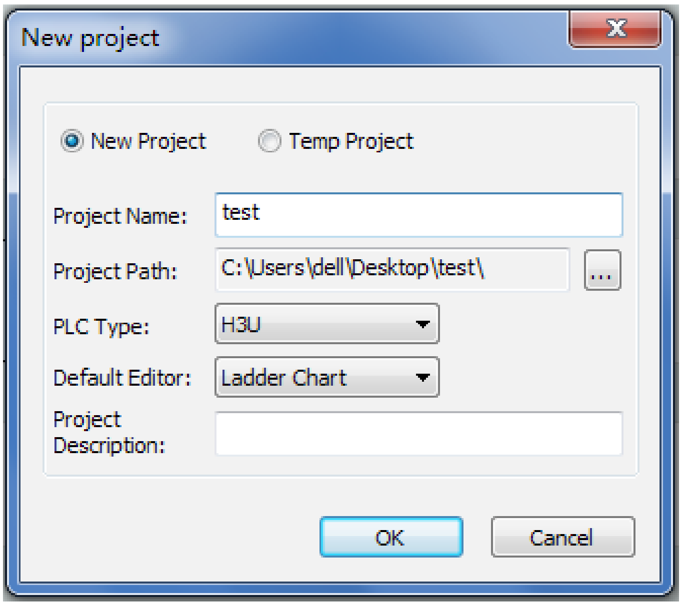
The commercially available RS232-Mini DIN8 plug cable dedicated for PLC project downloading or the Mini USB download cable can be used for user project download, commissioning, monitoring, and HMI connection.

If the computer does not have the DB9 RS232 serial port, the USB-Mini DIN8 dedicated download cable can also be used.

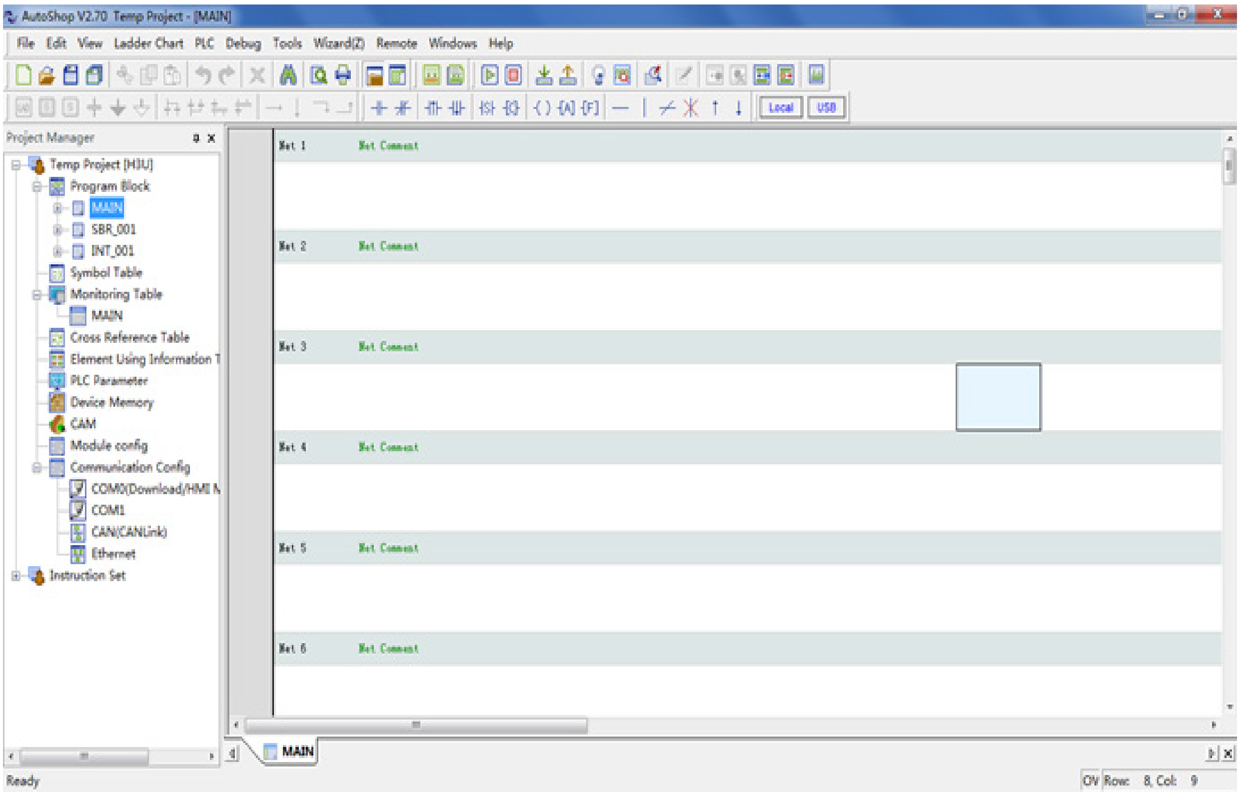
2.3 Introduction to AutoShop

1) Project creating

Choose **File** > **New Project**. The following dialog box is displayed.



Enter the relevant information as prompted, and click **OK**. A new H3U project is created.



2) Project input

AutoShop supports three programming languages: instruction list, ladder chart, and SFC. The ladder chart is the default programming language. The following describes how to input an instruction in ladder chart mode:

a) Click an instruction icon on the toolbar, and the **Application Instruction** window is displayed. Add an instruction to the current position.

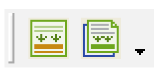
b) Select an instruction in **Instruction Set** in the **Project Manager** window.

- Drag the instruction to the window and add it to the current position through the **Application Instruction** window.
- Double-click the instruction, and the **Application Instruction** window is displayed. Add an instruction to the current position.

c) Select an instruction in **Ladder Chart** in the menu, and open the **Application Instruction** window to add the instruction to the current position.

d) If you are familiar with application instructions, you can use a keyboard to manually input a project.

3) Project compiling

a) Use the shortcut buttons for project compiling.  The first button is used to compile the current project. The second button is used to compile all projects.

b) Choose **PLC** > **Compile/Compile All** for compiling.

c) If no error is prompted during project compiling, the project can be downloaded.

4) Project uploading/downloading

a) Use a programming cable to connect the PLC to a PC.

b) Choose **Tool** > **Communication Setting** to set the communication mode.

c) Choose **PLC** > **Download/Upload** or click the shortcut buttons to upload or download the project.

NOTE

- A non-compiled project will be automatically compiled before downloading. An application project that cannot be compiled cannot be downloaded.

5) Project debugging

Monitoring mode: Choose **Debug** > **Monitor** from the menu or click the shortcut button to enter monitoring mode. In this mode, you can monitor the input/output status and current value of elements.

Online modification mode: Choose **PLC** > **Online Edit Mode** from the menu or click the shortcut button to enter the online modification mode. In this mode, you can modify a user project without stopping the PLC. After modifying a project, you can directly download the program without stopping the PLC.

Element monitoring: Double-click a subdirectory under Monitoring Table in the **Project Manager** window. The monitoring table interface is displayed. In monitoring mode, you can check the current element value in real time.

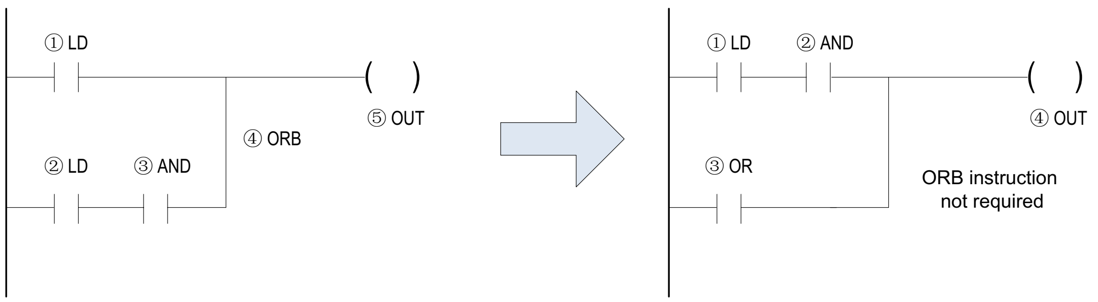
Using AutoShop to start/stop the PLC: Choose **PLC** > **Run/Stop** from the menu or click the shortcut button to control the PLC running status.

6) Project execution steps and optimization recommendations

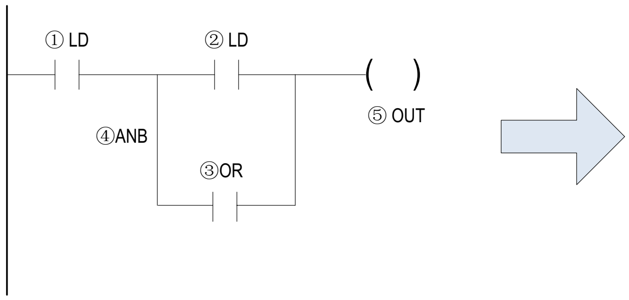
Project execution steps: User projects are scanned from top to bottom and from left to right in cycles.

Project execution sequence and optimization solution:

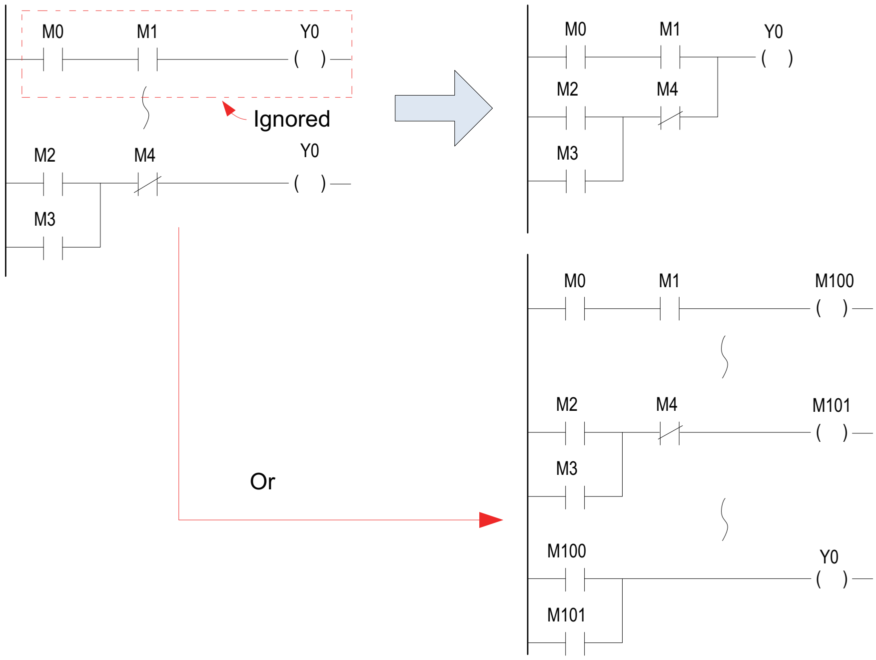
a) It is recommended that ladder chart with multiple serial contacts be placed above the project. (There is one step fewer in the right diagram than in the left diagram.)



b) It is recommended that ladder chart with multiple parallel contacts be placed at the left of the project. (There is one step fewer in the right diagram than in the left diagram.)



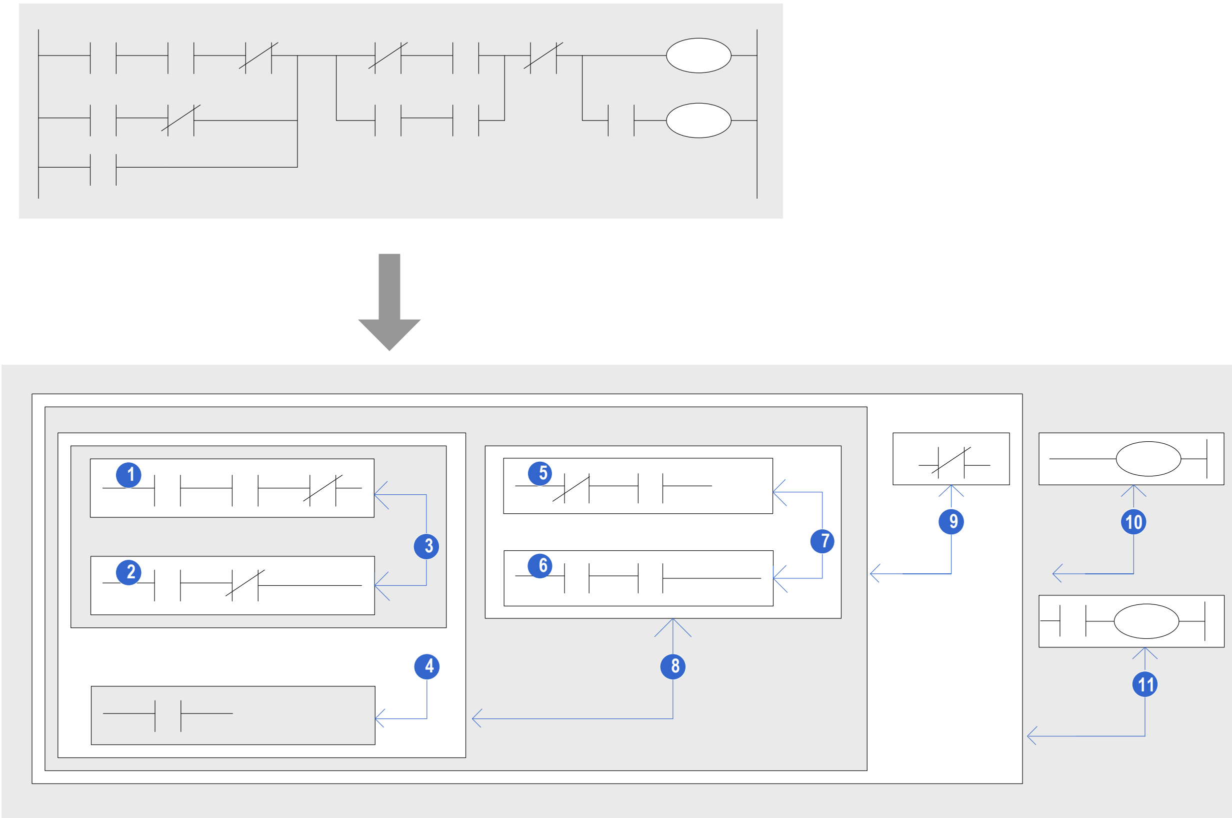
c) Dual-coil solution



The dual-coil design does not go against the project editing rule, but the output status may not be what the user expects. Because the actual port I/O status of the PLC is refreshed when the project is finished, only the last status of the project is refreshed, and intermediate status changes cannot be shown. To show I/O status change in the same scan cycle, you need to use the REF instruction.

Execution sequence and steps of a PLC project

The project is processed from top to bottom and from left to right.


1
2
3
4
5
6
7
8
9
10
11

3 Elements

3

Chapter 3 Elements

The following table lists types of elements supported by H3U.

No.	Type	Function and Classification
1	Input/Output relay	Corresponding to the PLC's bit element X for hardware boolean input
		Corresponding to the PLC's bit element Y for output control
2	Auxiliary relay	Ordinary auxiliary relay M
		System special auxiliary relay M
		System special auxiliary relay SM
3	Status relay	Status flag bit element S for step control
4	Timer	16-bit timers T with a step length of 1 ms, 10 ms, and 100 ms
5	Counter	16-bit/32-bit incremental/decremental counter C
		32-bit high-speed counter C
6	Data register	Ordinary data register D
		System special data register D
		System special data register SD
		Indirect addressing data registers V and Z
7	File register	File register R
8	Marker	Marker/Jump pointer P
9	Subprogram	Subprogram SBR
		Interrupt subprogram I
		Motion control subprogram MC
		G-code subprogram CNC
10	Nested pointer	Nested pointer N
11	Constant	Decimal constant K
		Hexadecimal constant H
		Floating-point number E

3.1 Table of Elements

Name	Description		
Input/Output relay			
Input relay	X0 to X377	256 points	Numbered with an octal number, 512 input and output points in total
Output relay	Y0 to Y377	256 points	
Auxiliary relay			
For general purpose	M0 to M499	500 points	Retained upon power failure
For retention purpose	M500 to M1023	524 points	
For retention purpose	M1024 to M7679	6656 points	Retained upon power failure
For special system use	M8000 to M8511	512 points	Retained upon power failure
For special system use	SM0 to SM1023	1024 points	Retained upon power failure
Status relay			
Initial status	S0 to S9	10 points	Retained upon power failure
For general purpose	S10 to S499	490 points	
For retention purpose	S500 to S899	400 points	
For alarm purpose	S900 to S999	100 points	Retained upon power failure
For retention purpose	S1000 to S4095	3096 points	Retained upon power failure
Timer			
100 ms	T0 to T191	192 points	0.1 to 3276.7 s
100 ms	T192 to T199	8 points	0.1 to 3276.7 s, used for subprograms and interrupt subprograms
10 ms	T200 to T245	46 points	0.01s to 327.67s
1 ms accumulation type	T246 to T249	4 points	0.001s to 32.767s
100 ms accumulation type	T250 to T255	6 points	0.1 to 3276.7 s
1 ms	T256 to T511	256 points	0.001s to 32.767s
Counter			
General-purpose incremental counter (16-bit)	C0 to C99	100 points	0 to 32,767, retained upon power failure
Incremental counter for retention purpose (16-bit)	C100 to C199	100 points	
General-purpose dual-direction (32-bit)	C200 to C219	20 points	-2,147,483,648 to +2,147,483,647, retained upon power failure
Dual-direction for retention purpose (32-bit)	C220 to C234	15 points	
High-speed counter			
Single-phase single-counting input dual-direction (32-bit)	C235 to C245	11 points	-2,147,483,648 to +2,147,483,647, retained upon power failure
Single-phase dual-counting input dual-direction (32-bit)	C246 to C250	5 points	
Dual-phase dual-counting input dual-direction (32-bit)	C251 to C255	5 points	
Data register			
For general purpose (16-bit)	D0 to D199	200 points	Retained upon power failure
For retention purpose (16-bit)	D200 to D511	312 points	
For retention purpose (16-bit)	D512 to D7999	7488 points	Retained upon power failure
For special purpose (16-bit)	D8000 to D8511	512 points	Retained upon power failure

3

Name	Description		
For special purpose (16-bit)	SD0 to SD1023	1024 points	Retained upon power failure
For address indexing (16-bit)	V0 to V7, 0 to Z7	16 points	Retained upon power failure
File register			
Extended register (16-bit)	R0 to R32767	32,768 points	Retained upon power failure
Marker			
For the CJ instruction	P0 to P511	512 points	Used in combination with the LBL instruction
Subprogram			
For the CALL instruction	/	512 points	Configurable as a general subprogram, an encrypted subprogram, a subprogram with parameters, or an encrypted subprogram with parameters
Output interrupts X000 to X007	I00 □ , I10 □ , I20 □ , I30 □ , I40 □ , I50 □ , I56 □ , I57 □	8 points	□ 0 indicates a falling edge interrupt, and 1 indicates a rising edge interrupt. When the edge interrupt disabling flag bit register is set to ON, the corresponding input interrupt is disabled.
Timing interrupt	I6 □□ to I8 □□	3 points	□□ = 01 to 99, and time base = 1 ms
Counting completion interrupt	I010 to I080	8 points	For the DHSCS instruction
Pulse completion interrupt	I502 to I506	5 points	
Motion control subprogram	MC00 to MC63	64	
G-code subprogram	CNC00	1	Corresponding to the MC10000. Each G-code subprogram supports up to 16 Oxxxx subprograms.
Nested pointer			
For the main control loop	N0 to N7	8 points	For the MC instruction
Constant			
Decimal constant K	16-bit	-32,768 to +32,767	
	32-bit	-2,147,483,648 to +2,147,483,647	
Hexadecimal constant H	16-bit	0 to FFFF	
	32-bit	0 to FFFFFFFF	
Real number E	32-bit	0, -1.0*2e128 to -1.0*2e-126, 1.0*2e-126 to 1.0*2e128 (32-bit)	

3.2 Input/Output Relay

3.2.1 Input Relay X

The input relay X is a PLC external input signal status element that uses the X port to detect the status of an external signal. 0 indicates an open-circuit external signal (OFF) and 1 indicates a closed external signal (ON).

The input relay status cannot be modified using a project instruction, and the contact signals (normally ON or normally OFF) can be used infinitely in the user project.

Relay signals are marked with X0, X1, …X7, X10, and X11, and numbered with an octal number.

When a local extension module is connected, the X ports on the extension module are numbered immediately after the X ports on the main module. For example, if the main module is the H3U standard model, when an AM600-1600END extension module is connected, because the last X port of the main module is X37, the X ports of the extension module are numbered X40 to X47, and X50 to X57 during programming.

NOTE

- The ports of extension modules are always numbered from an octal number starting with 0.

3.2.2 Output Relay Y

The output relay is an element that is directly associated with the hardware ports of external user control devices, and logically corresponds to the PLC's physical output port in one-to-one manner. Each time after the PLC finishes scanning a user project, it transfers the element status of the relay Y to its hardware ports, with 0 indicating an open-circuit output port (OFF), and 1 indicating a closed output port (ON).

The relays Y are marked with Y0, Y1, ···Y7, Y10, and Y11, and numbered with an octal number. The relays Y can be used infinitely in the user project. The hardware can be divided into relays and transistors. If output extension module ports exist, all the ports are numbered in sequence starting from the ports of the main module. When a local extension module is connected, the Y ports on the extension module are numbered immediately after the X ports on the main module. For example, if the main module is the H3U standard model, when an AM600-0016ETN extension module is connected, because the last Y port of the main module is Y37, the Y ports of the extension module are numbered Y40 to Y47, and Y50 to Y57 during programming.

Note: The ports of extension modules are always numbered from an octal number starting with 0.

3.3 Auxiliary Relay

Used as an intermediate variable during execution of the user project, the auxiliary relay M element functions like the auxiliary relay in an actual electronic control system. It transfers the status information and combines multiple M variables into a word variable. The M variables have no direct relation to external ports, but can copy X to M through a project statement or copy M to Y to establish contact with external ports. One M variable can be used infinitely.

The auxiliary relays are marked with M0, M1, ······ and M8511, and numbered with a decimal number. M8000 and subsequent variables are dedicated to the system for interaction of the PLC user project and system status. Part of the M variables can be retained upon power failure.

The special auxiliary relays SM are variables dedicated to the system for interaction of the PLC user project and system status. They are marked with SM0, SM1, ······ and SM1023, and numbered with a decimal number.

For general purpose	For retention purpose	For retention purpose	For special purpose	For special purpose
M0 to M499 500 points[1]	M500 to M1023 524 points[2]	M1024 to M7679 6656 points[3]	M8000 to M8511 512 points	SM0 to SM1023 1024 points

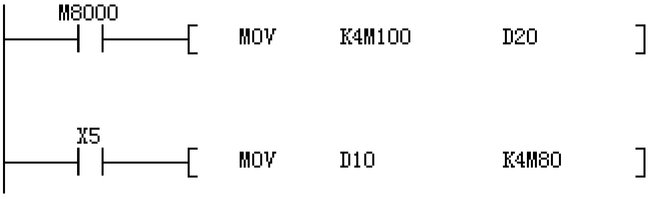
NOTE

- [1] It cannot be retained upon power failure. It can be set to be retained upon power failure through parameter setting.
- [2] It can be retained upon power failure. It can be set to be not retained upon power failure through parameter setting.
- [3] It can be retained upon power failure and the status cannot be modified through parameter setting.

You can use parameters to adjust the regions for the PLC's general auxiliary relays and auxiliary relays which can be retained upon power failure.

The PLC has many special auxiliary relays with different functions. For details, see “Appendix A Allocation of Soft Elements SM, SD, D8000, and M8000” on page 704. Note: Undefined special auxiliary relays are unusable.

Usage tips: You can access consecutive M variables by bytes or words. For example,



K4M100 indicates reading the M100, M101, M102……M115 (16 units in total) as one unit (using M100 as bit 0 of the word …… and M115 as bit 15 of the word), which can improve the programming efficiency.

3.4 Status Relay

The status relay S is used to design and execute stepping projects. It uses the STL stepping instruction to control the transfer of the stepping status S and thus simplifies the project design.

If the STL programming method is not used, the S can be used as a general bit element, like the M variable. The status S variables are marked with S0, S1, ……and S999, and numbered with a decimal number. Some of the S variables can be retained upon power failure.

For general purpose		For retention purpose	For alarm purpose	For retention purpose
S0 to S9 10 points[1]	S10 to S499 490 points[1]	S500 to S899 400 points[2]	S900 to S999 100 points	S1000 to S4095 3096 points[3]

NOTE

- [1] It cannot be retained upon power failure. It can be set to be retained upon power failure through parameter setting.
- [2] It can be retained upon power failure. It can be set to be not retained upon power failure through parameter setting.
- [3] It can be retained upon power failure and the status cannot be modified through parameter setting.

3.5 Timer

Timers are used for timing. Each timer consists of a coil, contacts, and a counting value register. When the timer coil is "energized" (the flow is valid), the timer starts timing. If the timing value reaches the preset time value, contact a (NO contact) is closed, and contact b (NC contact) is opened. When the coil is "de-energized" (the flow is invalid), the timer contacts restore to the initial state, and the timing value is automatically cleared. The timing values of some timers can be accumulated or retained upon power failure. The timing value before power failure is kept after the timer is energized again.

The timers are marked with T0, T1, …… and T255, and numbered with a decimal number. Timers have different timing steps, for example, 1 ms, 10 ms, and 100 ms. Some timers can be retained upon power failure.

100 ms		10 ms	1 ms accumulation-type	100 ms accumulation-type	1ms
T0 to T191 192 points ※1	T192 to T199 8 points ※2	T200 to T245 46 points ※1	T246 to T249 4 points ※3	T250 to T255 6 points ※4	T256 to T511 256 points ※1

NOTE

- [1] It cannot be retained upon power failure.
- [2] It cannot be retained upon power failure, and is used for subprograms and interrupt subprograms.
- [3] It can be retained upon power failure, and the timing value can be accumulated.
- [4] It can be retained upon power failure, and the timing value can be accumulated.

An unoccupied timer number can be used as a data register for numerical value storage.

The timer accumulates the clock pulses of 1 ms, 10 ms, 100 ms, and so on in the PLC. When the timing value reaches the preset value, the contacts take action only during execution of the coil instruction or END instruction.

You can use the constant (K) of the program memory as the preset value, or use the content of the data register (D) to specify the preset value.

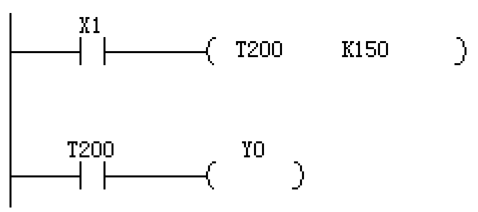
NOTE

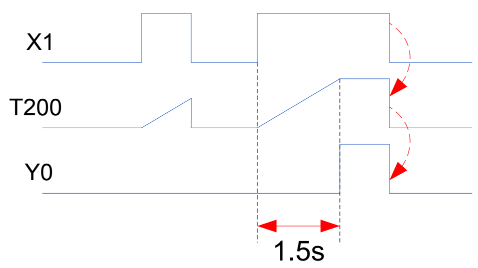
- The content of D must be set before timing starts. After timing starts, changes to the data of D can only take effect when timing starts next time.

The following describes the possible timing length from timer coil driving to timer contact action:

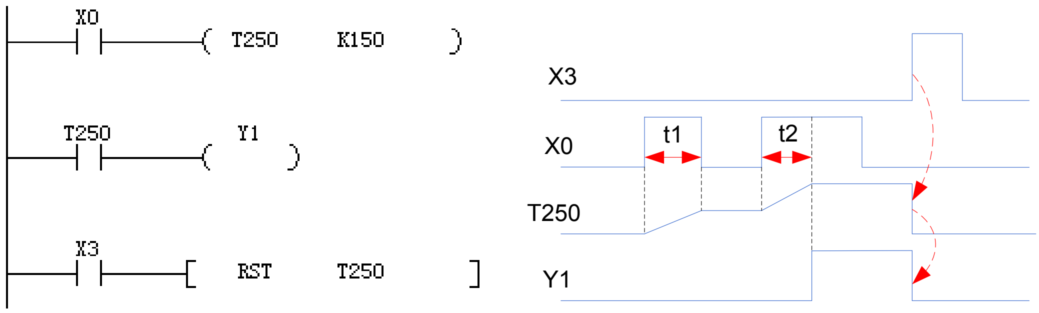
- The longest timing length is (T + T0 + a), wherein T is the preset timing value, T0 is project scanning time, and a is the timing step of the timer.
- The shortest timing length is (T – a).
- When the timer contact instruction is before the coil instruction, the timing length is (T + 2T0), which is the least ideal condition.
- The b contact of the timer can be used for delay-disconnected and self-oscillating output signals.
- The PLC also provides special timer instructions, for example, TTMR, and STMR. For details, see the descriptions on these instructions.

Example 1: The general timer T200 has a step of 10 ms, and the actual action delay is 150 × 10 ms ＝ 1500 ms (1.50s). The action principle is as follows:

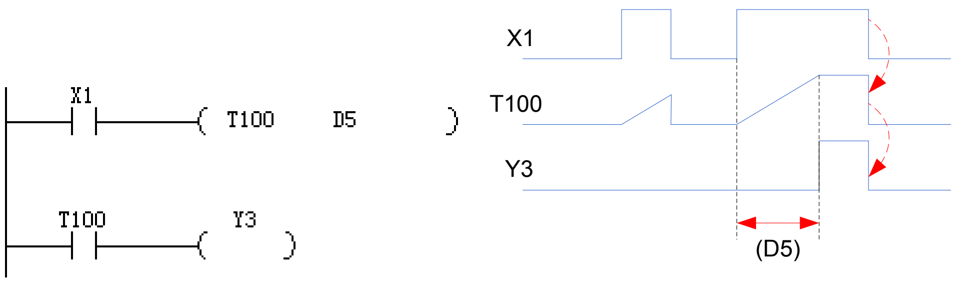




Example 2: The driving signal of a cumulative timer T250 which can be retained upon power failure is OFF, or the next driving signal is ON when the PLC is powered off and the internal timing value is retained. Timing continues until the preset value is reached, and the output contact is closed. When the timer coil is reset, the timing value is cleared, and the output contact is opened, as shown in the following figure. Because the step of the timer T250 is 100 ms, th actual cumulative action delay is 150 ms × 100 ms ＝ 15,000 ms (15.0s), that is, the t1 ＋ t2 in the figure.



Example 3: You can use the register D to set the action value for the timer, as shown in the figure below. (Changes to values in the register D during timing take effect when the timer is started next time.)



3.6 Counter

Counters are used for counting. Each counter consists of a coil, contacts, and a timing data value register. When the driving signal of the counter coil switches from OFF to ON, the counter reading is incremented by 1. When the counting value reaches the preset value, contact a (NO contact) is closed, and b contact b (NC contact) is opened. When the counting value is cleared, output contact a is opened, and contact b (NC contact) is closed. The values of some counters can be accumulated or retained upon power failure. The value before power failure is kept after the counter is energized again.

Counters are marked with C0, C1, ……and C255, and numbered with a decimal number.

For general purpose, 16-bit	For retention purpose, 16-bit	For general purpose, 32-bit	For retention purpose, 32-bit	For retention purpose, 32-bit
C0 to C99 100-point incremental counting[1]	C100 to C199 100-point incremental counting[2]	C200 to C219 20-point dual-direction counting[1]	C220 to C234 15-point dual-direction counting[2]	C235 to C255 21-point high-speed counting[2]

NOTE

- [1] It cannot be retained upon power failure. It can be set to be retained upon power failure through parameter setting.
- [2] It can be retained upon power failure. It can be set to be not retained upon power failure through parameter setting.

An unoccupied counter number can be used as a data register for data retention.

In 32-bit counters C200 to C234, the special auxiliary relays M8200 to M8234 are used to control switch between incremental and decremental counters, as shown in the following table.

Counter NO.	Direction Switch	Counter NO.	Direction Switch	Counter NO.	Direction Switch	Counter NO.	Direction Switch
C200	M8200	C209	M8209	C218	M8218	C226	M8226
C201	M8201	C210	M8210	C219	M8219	C227	M8227
C202	M8202	C211	M8211	N/A	N/A	C228	M8228
C203	M8203	C212	M8212	C220	M8220	C229	M8229
C204	M8204	C213	M8213	C221	M8221	C230	M8230
C205	M8205	C214	M8214	C222	M8222	C231	M8231
C206	M8206	C215	M8215	C223	M8223	C232	M8232
C207	M8207	C216	M8216	C224	M8224	C233	M8233
C208	M8208	C217	M8217	C225	M8225	C234	M8234

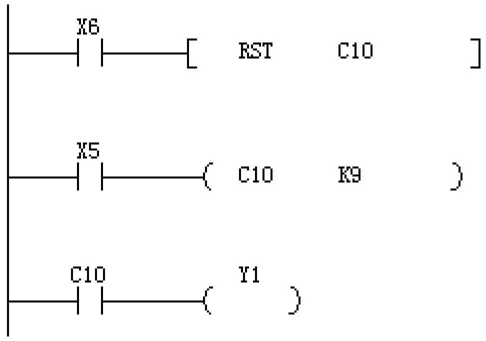
Characteristics of the 16-bit and 32-bit counters are listed in the following table. The counters can be used separately based on counting direction switch and use conditions of the counting scope.

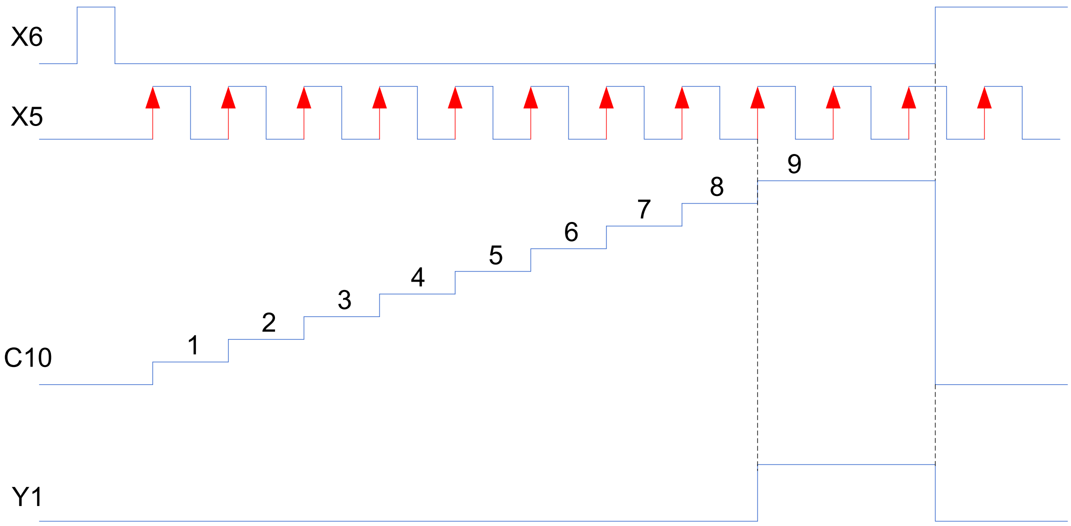
Item	16-bit Counter	32-bit Counter
Counting direction	Incremental	Switch between incremental and decremental counting (see the preceding table)
Set value	1 to 32,767	-2,147,483,648 to +2,147,483,647
Specified set value	The constant K or a data register	The constant K or two D data registers
Change to the current value	No change after counting on	Change after counting on (loop counter)
Output contact	Action retained after counting on	Action retained after counting on, and reset after counting back
Reset action	When the RST command is executed, the current value of the counter is zero, and the output contact is reset.	
Current value register	16-bit	32-bit

3.6.1 16-bit Counter

You can use the system parameter configurations to set and change allocation of counters for general purpose use and retention purpose use.

The valid value of a 16-bit counter ranges from K1 to K32,767 (a decimal constant). The value K0 and K1 have the same effect; that is, the output contact takes actions when the first counting starts, as shown in the following example.





Each time the counting input X5 drives the C10 coil, the current value of the counter increments, and the output contact takes action when the coil instruction is executed at the ninth time. After that, the current value of the counter remains unchanged even the counting input X5 continues to drive the coil. If the input X6 is set to **ON**, the RST instruction is executed, the current value of the counter is cleared, and the output contact is reset.

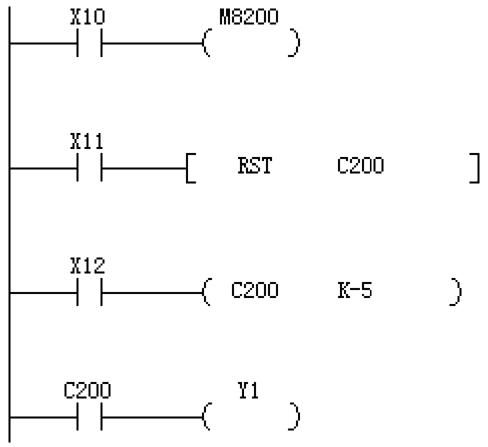
You can use the constant K or the data register to set the value of the counter. In the example above, you can use D20 to replace K9. If D20 is set to 9, the effect is the same as that when the value is K9.

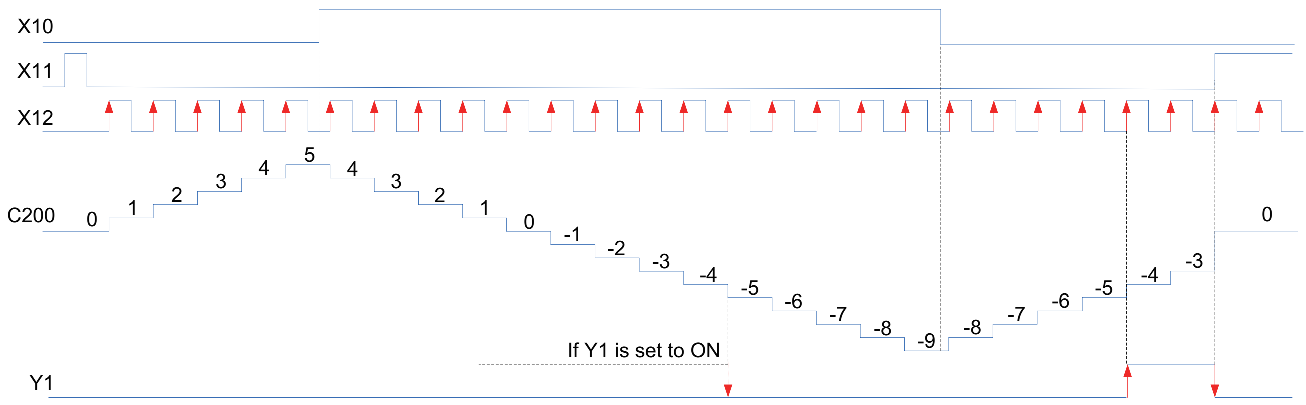
When a value larger than the preset value is written into the counter (for example, C10) through MOV or other instructions, the current value of the counter will change to the preset value and the corresponding output coil will be switched on during next counting.

If the PLC is powered off, the counting value of a counter for general purpose is cleared, while the counting value before power failure of a counter which can be retained during power failure can be stored, and the counter continues the counting following the previous counting value.

3.6.2 32-bit Counter

The value range for a 32-bit incremental/decremental counter is -2,147,483,648 to +2,147,483,647 (a decimal constant). You can use the constant K or the content of the data register D to set the value. You can use the special auxiliary relays M8200 to M8234 to specify the counting incremental/decremental direction. If the M8 △△△ (corresponding to C △△△) is set to **1**, the counter changes to be decremental counter. Otherwise, the counter is incremental counter.





The increment/decrement of the current value is unrelated to the action of the output contact. However, if counting increments from 2,147,483,647, the reading changes to -2,147,483,648 after another pulse is input. Similarly, if counting decrements from -2,147,483,648, the reading changes to 2,147,483,647 after another pulse is input. Such an action is called ring counting. If the input X11 is set to **ON**, the RST instruction is executed, the current value of the counter is cleared, and the output contact is reset.

When a counter which can be retained upon power failure is used, the current value of the counter, the output contact action, and reset status are retained upon power failure.

A 32-bit counter can be used as a 32-bit data register. However, the 32-bit counter cannot be used as an element in the 16-bit application instructions.

When a value larger than the set value is written into the counter by using the DMOV or other instructions, the counting continues upon subsequent counting input, and the contact does not change.

The higher bit (bit 15) of a 16-bit counter is the sign bit. The data processed is within the range 0 to 32,767; that is, only positive numbers can be processed.

The higher bit (bit 31, the higher bit of the high-order byte) of a 32-bit counter is the sign bit. The data processed is within the range -2,147,483,648 to +2,147,483,647.

3.6.3 High-speed Counter

High-speed counters can be used for counting external input signals and supports single-phase single-counting, single-phase dual-counting, and A/B-phase fundamental or quadruplicated frequency. For details about use of high-speed counters, see "Chapter 5 High-speed Input" on page 336.

3.7 Register

Registers are used for data computation and storage of parameters of timers, counters, and analog values. The width of each register is 16-bit. When a 32-bit instruction is used, the two neighboring registers are automatically combined into a 32-bit register, with the lower address being used as the low-order byte, and the higher address being used as the high-order byte.

The H3U supports the data register D, address indexing data registers V and Z, and file register R.

For general purpose	For retention purpose	For retention purpose	For special purpose	For address indexing purpose	For retention purpose	For special purpose
D0 to D199 200 points[1]	D200 to D511 312 points[2]	D512 to D7999 7488 points[3]	D8000 to D8511 512 points	V0 to V7 Z0 to Z7	R0 to R32767 32,768 points[3]	SD0 to SD1023 1024 points

NOTE

- [1] It cannot be retained upon power failure. It can be set to be retained upon power failure through parameter setting.
- [2] It can be retained upon power failure. It can be set to be not retained upon power failure through parameter setting.
- [3] It can be retained upon power failure and the status cannot be modified through parameter setting.

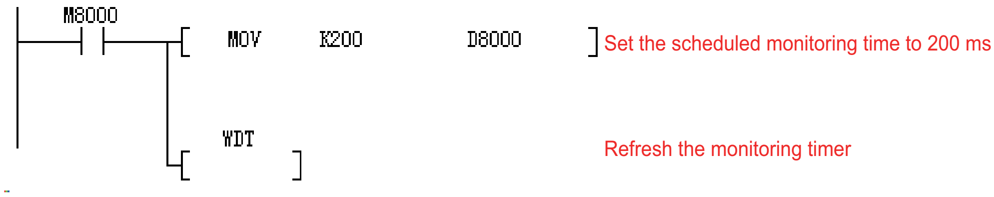
3.7.1 Data Register D

The width of each data register D is 16-bit. When 32-bit data is used, the two neighboring data registers D are combined to demonstrate 32-bit data. (The bigger D register is the higher 16 bits, and the smaller D register is the lower 16 bits. In an address indexing register, V is the higher bit, and Z is the lower bit.) When the lower bit (for example, D0) of the 32-bit register is specified, the higher-bit number (for example, D1) following the lower bit is automatically occupied. You can use an odd or even number of any element to specify the lower bit. Considering the monitoring function of peripheral devices, it is suggested using an even number of an element to specify the lower bit.

For a data register which cannot be retained upon power failure, no change occurs as long as no other data is written into the register after initial data writing. However, when the slide switch of PLC switches from RUN to STOP or PLC power off, all data will be cleared. (The data can be retained if the special auxiliary relay M8033 is driven.) If the data register is of power failure retain type, the data can be retained when the slide switch of PLC switches from RUN to STOP or PLC power off.

You can modify the allocation of D registers for general purpose use or retention purpose use by setting system parameters. When data registers dedicated for retention upon power failure are used for general purpose, use the RST or ZRST instruction to clear their content when starting the project.

The special register is used to realize some special functions of PLC by specific values are written into. It is a special unit used for data interaction between the user project and the PLC system project. For example, in D8000, the time of the monitoring timer is initially set through the system ROM. To change the time, you need to use the MOV instruction to write the target time in D8000.



Besides, some special D registers are used to cache the system running status parameters. You can query such registers to check the running parameters.

For characteristics of special data registers which can be retained upon power failure, see "Appendix A Allocation of Soft Elements SM, SD, D8000, and M8000" on page 704.

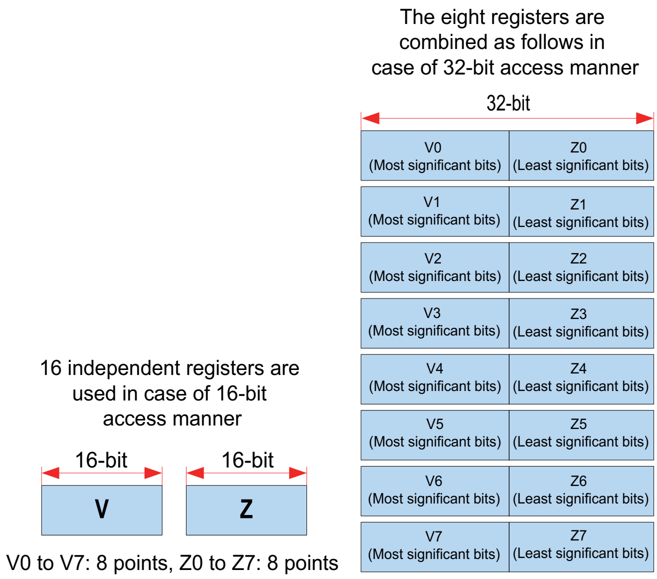
The data register can be used to process numerical values of various types for control purpose. For example, it can be used for defining the value of a timer or a counter, data computation and so on. The instructions supporting D registers will be described in detail later.

3.7.2 Address Indexing Registers V and Z

Same as the general data register, the address indexing registers V and Z are 16-bit data registers for numerical value reading and writing. There are 16 address indexing registers, namely, V0 to V7 and Z0 to Z7.

The address indexing registers can be used in the same way as that of general data registers. Besides, they can also be used in combination with the numbers or numerical values of other elements in the operands of application instructions. Note that the numbers of elements of basic sequential control instructions (such as LD, AND, and OUT) or STL instructions cannot be used together with the address indexing registers.

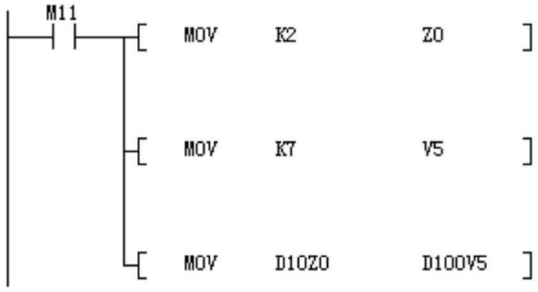
The following figure shows how to use the V and Z registers to access data in 16-bit and 32-bit manners.



Conventionally, to process elements in a 32-bit application instruction or a numerical value more than 16 bits (in 32-bit register manner), the V (higher bit) and Z (lower bit) registers are accessed simultaneously, and the names of the specified registers must be Z0 to Z7. The address cannot be indexed even when the higher bits of V0 to V7 are specified.

3

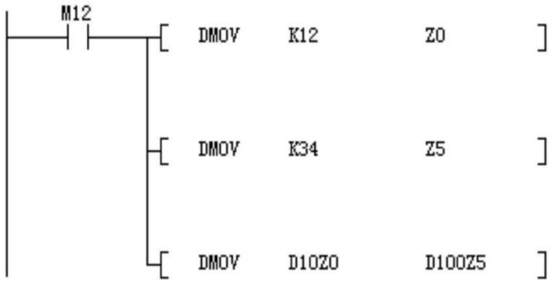
1) **Example of 16-bit address indexing applications:**



Because: D10Z0 = D (10 + 2) = D12
D100V5 = D (100 + 7) = D107

Which is equivalent to:
MOV D12 D107

2) **Example of 32-bit address indexing applications:**



Because: D10Z0=D[10+(V0,Z0)]=D[10+12]=D22
D100Z5=D[100+(V5,Z5)]=D[100+34]=D134

Which is equivalent to:
DMOV D22 D134

3) **Special example of constant-based address indexing**

Z0 plus K5, and the result is sent to D10 while the Z0 value remains unchanged

When the V and Z indirect addressing method is used in the loop instructions (V and Z change with the loop variables) for operation on data zones in batches or table query, the programming is simplified and the instruction efficiency is improved.

3.7.3 File Register R

The H3U supports 32,768-point 16-bit file registers. The file register R is used in the same way as that of the data register D. For details, see "3.7.1 Data Register D" on page 42.

3.8 Marker and Subprogram

The marker/jump pointer (P) is used to mark the portal address of the jump program. The subprogram SBR is used to mark the starting address of a subprogram. The motion control subprogram is marked with MC. The interrupt subprogram (I) is used to mark the starting address of an interrupt program, and is numbered with a decimal number.

Label	Subprogram	Overview	
P	For the CJ instruction	Used in combination with the LBL instruction The marker is used in each program block and cannot jump outside the current program block. Up to 512 jump pointers are supported in all program blocks.	
L	For the CJ instruction	Equivalent to P.	
SBR	For the CALL instruction	Up to 512 subprograms are supported. The subprograms can be set to general subprograms, encrypted subprograms, subprograms with parameters, and encrypted subprograms with parameters. The capacities of encrypted subprograms, subprograms with parameters, and general subprograms are not restricted, Such three types of subprograms share the 64K-step capacity of the system.	
I	Interrupt subprogram	External interrupt	X000-X007 input interrupt numbered I00 □ , I10 □ , I20 □ , I30 □ , I40 □ , I50 □ , I56 □ , and I57 □ . 8 points (□ indicates a falling edge interrupt, and 1 indicates a rising edge interrupt.) When the edge interrupt disabling flag bit register is set to ON, the corresponding input interrupt is disabled.
		Timing interrupt	I6 □□ , I7 □□ , I8 □□ , 3 points (□□ = 1 to 99, time base = 1 ms)
		Counting completion interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points (used by the DHSCS instruction)
		Pulse completion interrupt	I502 to I506, 5 points
MC	Motion control subprogram (Only supported by the H3U-PM series)	Up to 64 motion control subprograms are supported and numbered from MC0 to MC63. In addition, one G-code subprogram numbered MC10000 is supported. The G-code subprogram file supports up to Oxxxx codes numbered from O0000 to O9999. The capacities of motion subprograms and other subprograms are not restricted. They share the 64K-step capacity of the system.	

For details about the use method of interrupt and subprogram pointers, see “Chapter 11 Interrupt” on

page 686.

3.9 Constant

Five types of numerical values can be used based on the use and purpose of the PLC. Their roles and functions are listed in the following table below.

Type	Description of Application in Programming
Decimal, DEC	It is used to set the value of the timer and counter (constant K). It is used to specify the number of the auxiliary relay (M), timer (T), counter (C), and status S (the element number). It is used to specify the numerical value of the application instruction operand and instruction action (constant K).
Hexadecimal, HEX	Similar to DEC, it is used to specify the numerical value of the application instruction operand and instruction action (constant H).
Binary, BIN	The DEC or HEX can be used to specify numerical values for the timer, counter or data register, but the numerical values are processed as binary numbers inside the PLC. Besides, when used on a peripheral device for monitoring purpose, these elements will automatically convert to decimal or hexadecimal numbers.
Octal, OCT	The element numbers of input relays and output relays are allocated as octal values. Therefore, 0-7, 10-17……70-77, 100-107 are allowed, but 8, 9 are not allowed.
BCD	BCD is a method in which 4-bit binary values are used to represent decimal numbers 0 to 9. The bits can be easily processed, and can be used to control the display of the digital switch or 7-segment codes of the BCD output parameters.
Binary floating-point number	The PLC provides the high-precision floating-point operation function using BIN floating-point numbers.
Decimal flowing-point number	Decimal flowing-point values are used only for monitoring and easy reading purpose.

[K] is a symbol indicating a decimal integer. It is mainly used to specify the value of the timer or counter or the numerical value in the application instruction operand. In a 16-bit instruction, the value range of the constant K is -32768 to +32767. In a 32-bit instruction, the value range of the constant K is -2,147,483,648 to +2,147,483,647.

[H] is a symbol indicating a hexadecimal value. It is mainly used to specify the numerical value in the application instruction operand. In a 16-bit instruction, the value range of the constant H is 0000 to FFFF. In a 32-bit instruction, the value range of the constant H is 0x0 to 0xFFFFFFFF.

4 Instructions

Chapter 4 Instructions

This chapter describes the instructions supported by H3U.

4.1 Program Logic Instructions

4.1.1 Contact Instructions	LD	Loading of normally open (NO) contacts
	LDI	Loading of normally closed (NC) contacts
	AND	Serial connection of NO contacts
	ANI	Serial connection of NC contacts
	OR	Parallel connection of NO contacts
	ORI	Parallel connection of NC contacts
	LDP	Use of rising edge pulse
	LDF	Use of falling edge pulse
	ANDP	Serial connection for AND rising pulse detection
	ANDF	Serial connection for AND falling pulse (F) detection
	ORP	Parallel connection for OR rising pulse detection
	ORF	Parallel connection for OR falling pulse (F) detection
	INV	Operation result inversion
	BLD	Bit contact of bit data
	BLDI	Inverse bit contact of bit data
	BAND	AND bit contact of bit data
	BANI	ANI bit contact of bit data
	BOR	OR bit contact of bit data
	BORI	ORI bit contact of bit data
4.1.2 Combined Instructions	ANB	Serial connection of circuit blocks
	ORB	Parallel connection of circuit blocks
	MPS	Stack-based storage
	MRD	Stack read (flow pointer unchanged)
	MPP	Stack read
	MEP	Flow edge control, generate falling/rising edge pulse for operation results
	MEF	
4.1.3 Output Instructions	OUT	Coil drive
	SET	SET action storage coil instruction
	RST	Contact or cache clearance
	PLS	Rising pulse detection coil instruction
	PLF	Falling pulse (F) detection coil instruction
	ALT	Alternate output
	BOUT	Bit data output
	BSET	Bit data setting
	BRST	Bit data reset
4.1.4 Main Control Instructions	MC	Coil instruction for serial contacts used by the main control
	MCR	Release instruction for serial contacts used by main control reset
4.1.5 End Instructions	FEND	End of the main program
	END	End of all programs
4.1.6 Other Processing Instructions	NOP	No action
	WDT	Watchdog timer reset

4.1.1 Contact Instructions

Contact instructions	LD	Loading of NO contacts
	LDI	Loading of NC contacts
	LDP	Use of rising edge pulse
	LDF	Use of falling edge pulse
	AND	Serial connection of NO contacts
	ANI	Serial connection of NC contacts
	ANDP	Serial connection for AND rising pulse detection
	ANDF	Serial connection for AND falling pulse (F) detection
	OR	Parallel connection of NO contacts
	ORI	Parallel connection of NC contacts
	ORP	Parallel connection for OR rising pulse detection
	ORF	Parallel connection for OR falling pulse (F) detection
	INV	Operation result inversion
	BLD [Note]	Bit contact of bit data
	BLDI [Note]	Inverse bit contact of bit data
	BAND [Note]	AND bit contact of bit data
	BANI [Note]	ANI bit contact of bit data
	BOR [Note]	OR bit contact of bit data
	BORI [Note]	ORI bit contact of bit data

Note: Use the contact instructions to selectively take one bit of a word element or double word element for the operation. A word instruction occupies five steps, whereas a double word instruction occupies nine steps.

Word instructions and double word instructions have the same set of operands. The first operand indicates a word element or double word element; the second operand indicates the bit of the element for the operation. The second operand ranges from 0 to 15 for word instructions and 0 to 31 for double word instructions.

4

LD, LDI, LDP, LDF, AND, ANI, ANDP, ANDF, OR, ORI, ORP, ORF, and INV

LD	Loading of NO contacts	Operand types: S, X, Y, M, T, and C Number of steps: 1 step
LDI	Loading of NC contacts	
LDP	Use of rising edge pulse	
LDF	Use of falling edge pulse	

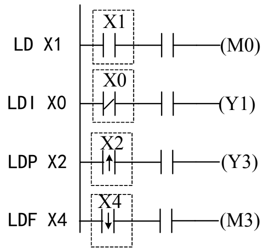
Instruction	Operand						
LD LDI LDP LDF	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
	√	√	√	√	√	√	√

The LD, LDI, LDP, and LDF instructions are used by the contacts starting from the left-hand bus.

Use the LD and LDI instructions to store the current flow statuses of contacts A and B and store the acquired contact status in a cumulative cache.

Use the LDP instruction to acquire the rising edge of a contact signal. If rising edge jump is scanned in a signal, the contact is active, but it becomes inactive during the next scan operation.

Use the LDF instruction to acquire the falling edge of a contact signal. If falling edge jump is scanned in a signal, the contact is active, but it becomes inactive during the next scan operation.



AND	Serial connection of NO contacts	Operand types: S, X, Y, M, T, and C Number of steps: 1 step
ANI	Serial connection of NC contacts	
ANDP	Serial connection for AND rising pulse detection	Number of steps: 3 steps
ANDF	Serial connection for AND falling pulse (F) detection	

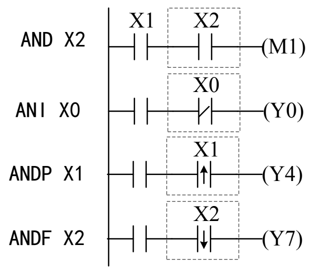
Instruction	Operand						
AND ANI ANDP ANDF LDF	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
	√	√	√	√	√	√	√

Use the AND, ANI, ANDP, and ANDF instructions to perform a status operation on serial contacts. These instructions read the status of the designated serial contact and perform an AND operation on the contact status and the contact's logical operation result. The AND result is stored in the cumulative cache.

Use the AND and ANI instructions to acquire the statuses of contacts A and B for an AND operation.

Use the ANDP instruction to acquire the rising edge jump status of a contact for an AND operation.

Use the ANDF instruction to acquire the falling edge jump status of a contact for an AND operation.



OR	Parallel connection of NO contacts	Operand types: S, X, Y, M, T, and C Number of steps: 1 step
ORI	Parallel connection of NC contacts	
ORP	Parallel connection for OR rising pulse detection	Number of steps: 3 steps
ORF	Parallel connection for OR falling pulse (F) detection	

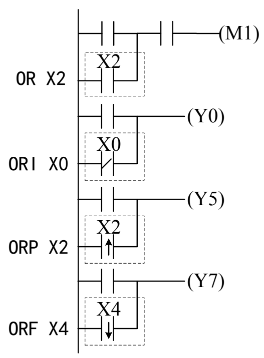
Instruction	Operand						
OR ORI ORP ORF	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
	√	√	√	√	√	√	√

Use the OR and ORI instructions to perform a status operation on parallel contacts. These instructions read the status of the designated parallel contact and perform an OR operation on the contact status and the contact's logical operation result. The OR result is stored in the cumulative cache.

Use the OR and ORI instructions to acquire the statuses of contacts A and B for an OR operation.

Use the ORP instruction to acquire the rising edge jump status of a contact for an OR operation.

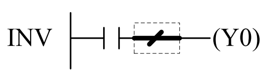
Use the ORF instruction to acquire the falling edge jump status of a contact for an OR operation.



INV	Operation result inversion	Number of steps: 1 step

Instruction	Operand
INV	None

The INV instruction performs phase inversion of the logical operation result prior to this instruction. The result is stored in the cumulative cache. After the INV instruction is executed, the flow status switches from ON to OFF, or vice versa.



BLD: Bit contact of bit data

◆ Overview

The execution result (ON or OFF) of the BLD instruction is determined based on the status (ON or OFF) of the designated bit of the source data (node A directly connected to the left-hand bus).

BLD S n		**Bit contact of bit data**	**Applicable model:** H3U	
S	Source data	Element number of the source data	16-bit instruction (5 steps) BLD: Continuous execution	32-bit instruction (9 steps) DBLD: Continuous execution
n	Loaded bit	Designated loaded bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)		

4

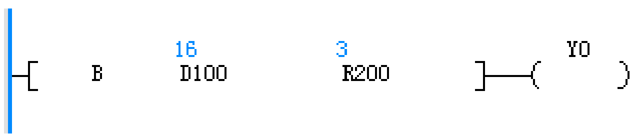
◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Application

n = 3:



n = 4:

BLDI: Inverse bit contact of bit data

◆ Overview

The execution result (ON or OFF) of the BLDI instruction is determined based on the status (ON or OFF) of the designated bit of the source data (node B directly connected to the left-hand bus).

BLDI S n			Inverse bit contact of bit data	Applicable model: H3U	
S	Source data	Element number of the source data		16-bit instruction (5 steps) BLDI: Continuous execution	32-bit instruction (9 steps) DBLDI: Continuous execution
n	Loaded bit	Designated loaded bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)			

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Application

n = 3:

n = 4:

BAND: AND bit contact of bit data

◆ Overview

The execution result (ON or OFF) of the BAND instruction is determined based on the status (ON or OFF) of the designated bit of the source data (node A connected to another node in series).

BAND S n			AND bit contact of bit data	Applicable model: H3U
S	Source data	Element number of the source data	16-bit instruction (5 steps) BAND: Continuous execution	32-bit instruction (9 steps) DBAND: Continuous execution
n	Loaded bit	Designated loaded bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Application

BANI: ANI bit contact of bit data

◆ Overview

The execution result (ON or OFF) of the BANI instruction is determined based on the status (ON or OFF) of the designated bit of the source data (node B connected to another node in series).

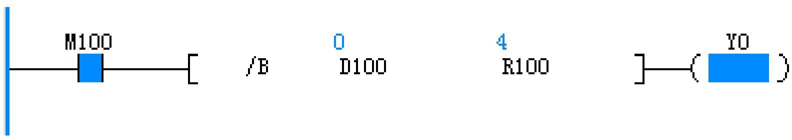
BANI S n			ANI bit contact of bit data	Applicable model: H3U
S	Source data	Element number of the source data	16-bit instruction (5 steps) BANI: Continuous execution	32-bit instruction (9 steps) DBAND: Continuous execution
n	Loaded bit	Designated loaded bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Application



4

BOR: OR bit contact of bit data

◆ Overview

The execution result (ON or OFF) of the BOR instruction is determined based on the status (ON or OFF) of the designated bit of the source data (node A connected to another node in parallel).

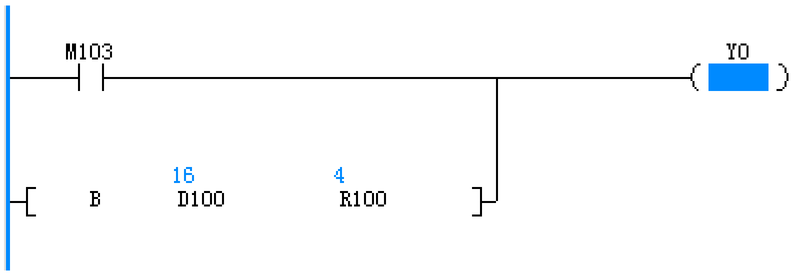
BOR S n		OR bit contact of bit data	Applicable model: H3U	
S	Source data	Element number of the source data	16-bit instruction (5 steps) BOR: Continuous execution	32-bit instruction (9 steps) DBOR: Continuous execution
n	Loaded bit	Designated loaded bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Application



BORI: ORI bit contact of bit data

◆ Overview

The execution result (ON or OFF) of the BORI instruction is determined based on the status (ON or OFF) of the designated bit of the source data (node B connected to another node in parallel).

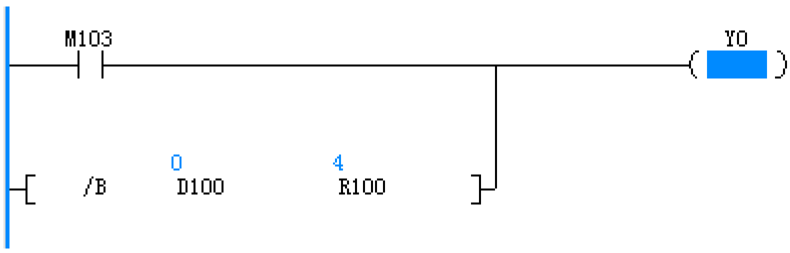
BORI S n			**ORI bit contact of bit data**	**Applicable model: H3U**	
S	Source data	Element number of the source data		16-bit instruction (5 steps) BORI: Continuous execution	32-bit instruction (9 steps) DBORI: Continuous execution
n	Loaded bit	Designated loaded bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Application



4.1.2 Combined Instructions

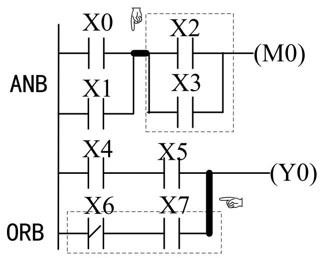
Combined instructions	ANB	Serial connection of circuit blocks
	ORB	Parallel connection of circuit blocks
	MPS	Stack-based storage
	MRD	Stack read (flow pointer unchanged)
	MPP	Stack read
	MEP	Flow edge control, generate falling/rising edge pulse for operation results
	MEF	

MPS: Stack-based storage; MRD: Stack read; MPP: Stack read

ANB	Serial connection of circuit blocks	Number of steps: 1 step
ORB	Parallel connection of circuit blocks	Number of steps: 1 step

Instruction	Operand
ANB	None
ORB	The objects of a block operation are the computing flows within the latest two LD (or LDI, LDP, or LDF) ranges.

Use the ANB and ORB instructions to perform an AND or OR operation on the previously saved logical operation result and the content of the cumulative cache.




```
☛LD X0
 OR X1
☛LD X2
 OR X3
☞ANB
 OUT M0
```

```
☛LD X4
 AND X5
☛LDI X6
 AND X7
☞ORB
 OUT Y0
```

MPS stack-based storage, MRD stack read, and MPP stack read

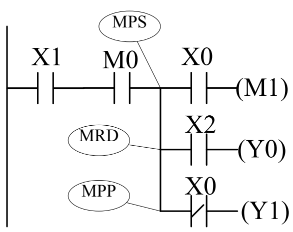
MPS	Stack-based storage	Number of steps: 1 step
MRD	Stack read (flow pointer unchanged)	Number of steps: 1 step
MPP	Stack read	Number of steps: 1 step

Instruction	Operand
MPS, MRD, and MPP	None

Use the MPS instruction to store the content of the cumulative cache in the stack. (The stack pointer is incremented by 1.)

Use the MRD instruction to read stack content and store the content in the cumulative cache. (The stack pointer remains unchanged.)

Use the MPP instruction to retrieve the previously saved logical operation result from the stack and store it in the cumulative cache. (The stack pointer is decremented by 1.)




```
LD X1
AND M0
MPS
AND X0
OUT M1
MRD
```

```
AND X2
OUT Y0
MPP
ANI X0
OUT Y1
```

MEP and MEF: Generate falling/rising edge pulse for operation results

◆ Overview

The MEP and MEF instructions generate falling/rising edge pulse for operation results. No element number needs to be specified.

1) MEP

The operation result until the MEP instruction is enabled upon OFF-to-ON switching.

Use the MEP instruction to enable simple pulsed processing when multiple contacts are connected in series.

2) MEF

The operation result until the MEF instruction is enabled upon ON-to-OFF switching.

Use the MEF instruction to enable simple pulsed processing when multiple contacts are connected in series.

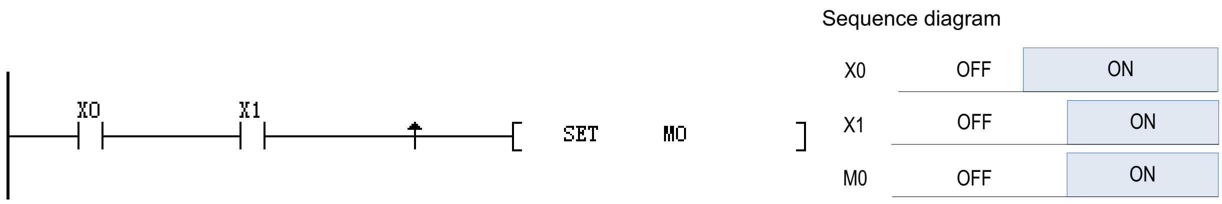
MEP and MEF	Generate falling/ rising edge pulse for operation results	Applicable model: H3U	
			1 (step)

◆ Operands

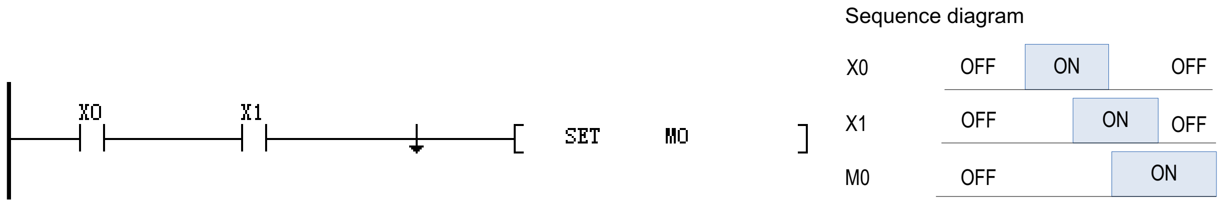
Instruction	Bit Element	Word Element				
	System·User	System·User	Bit Designation	Indexed Address	Constant	Real Number
MEP	No target element					
MEF	No target element					

◆ Application

1) MEP instruction (operation result: rising edge = ON)



2) MEF instruction (operation result: falling edge = ON)



4.1.3 Output Instructions

Output instructions	OUT	Coil drive
	SET	SET action storage coil instruction
	RST	Contact or cache clearance
	PLS	Rising pulse detection coil instruction
	PLF	Falling pulse (F) detection coil instruction
	ALT	Alternate output
	BOUT [Note]	Bit data output
	BSET [Note]	Bit data setting
	BRST [Note]	Bit data reset

Note: Use the output instructions to selectively take one bit of a word element or double word element for the operation. A word instruction occupies five steps, whereas a double word instruction occupies nine steps.

Word instructions and double word instructions have the same set of operands. The first operand indicates a word element or double word element; the second operand indicates the bit of the element for the operation. The second operand ranges from 0 to 15 for word instructions and 0 to 31 for double word instructions.

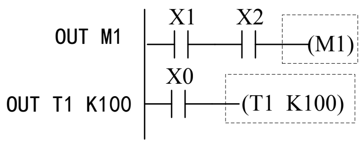
OUT, SET, RST, PLS, and PLF

OUT	Coil drive	Operand types: S, Y, and M Number of steps: 1 step
SET	SET action storage coil instruction	
RST	Contact or cache clearance	Operand types: S, Y, M, T, C, and D Number of steps: 3 steps
PLS	Rising pulse detection coil instruction	
PLF	Falling pulse (F) detection coil instruction	

4

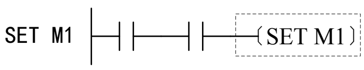
Instruction	Operand						
OUT	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
		√	√	√	√	√	√

The OUT instruction outputs the logical operation result prior to this instruction to the designated element.



Instruction	Operand						
SET	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
		√	√	√	√		

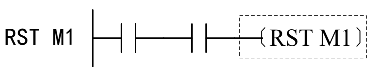
When the SET instruction is driven, the element designated by this instruction is set to ON and remains so regardless of whether the instruction is still driven. Use the RST instruction to set the element to OFF.



Instruction	Operand						
RST	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
		√	√	√	√	√	√
			D0–D8511	R0–R32767	SD0–SD1023		
			√	√	√		

When the RST instruction is driven, the element designated by this instruction is set to OFF and remains so regardless of whether the instruction is still driven. Use the SET instruction to set the element to ON.

Use the RST instruction to reset the D, V, and Z variables. That is, the values of the D, V, and Z elements are cleared.



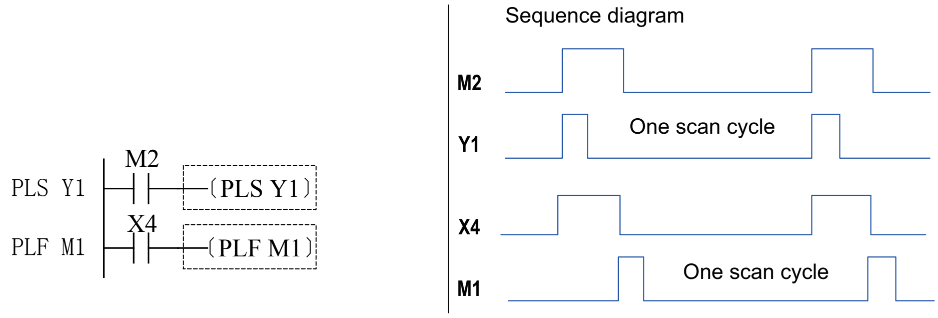
Element	Operation Result
S, M, and Y	Coils and contacts are set to OFF.
T and C	The timer or counter in use is set to 0, and coils and contacts are set to OFF.
D, V, and Z	The values of the elements are cleared.

Instruction	Operand						
PLS PLF	X0–X377	Y0–Y377	M0–M7679 M8000–M8511	S0–S4095	SM0–SM1023	T0–T511	C0–C255
		√	√		√		

When the PLS instruction is driven by the rising edge, the element designated by this instruction is set to ON and remains so within only one scan cycle.

When the PLF instruction is driven by the falling edge, the element designated by this instruction is set to ON and remains so within only one scan cycle.

Example:



ALT: Alternate output

◆ Overview

When driving conditions are met, the ALT instruction executes ON-OFF switching for the bit element D.

ALT D			Alternate output	Applicable model: H3U	
D	Execution operand	Bit element		16-bit instruction (3 steps) ALT: Continuous execution ALTP: Pulse execution	

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

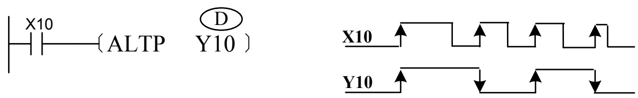
Note: The elements in gray background are supported.

◆ Function

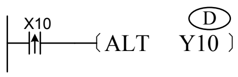
The status of the D element is switched when the flow is active. The D element is a bit variable element.

The ALTP instruction of the pulse execution type is usually used.

Example 1:

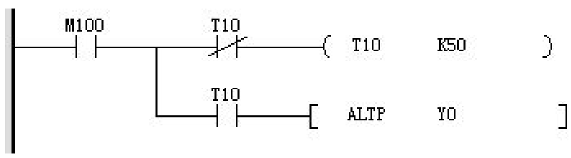


The action generated by the following instruction is the same as that generated by the ALTP instruction:



Example 2:

Introduce a timer to the instruction flow for convenient output of an oscillator. (The STMR instruction achieves the same result). See the following figure.



BOUT: Bit data output

◆ Overview

The BOUT instruction outputs the logical operation result prior to this instruction to the designated element.

BOUT D n			Bit data output	Applicable model: H3U	
D	Output data	Element number of the output data		16-bit instruction (5 steps) BOUT: Continuous execution	32-bit instruction (9 steps) DBOUT: Continuous execution
n	Output bit	Designated output bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)			

◆ Operands

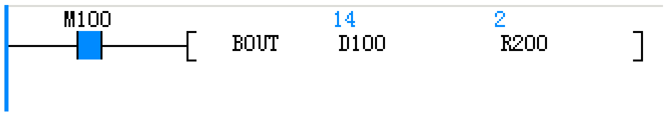
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

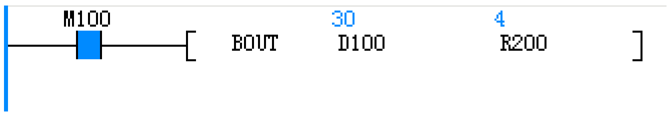
◆ Application

Initial value of D100 = 2#1010 (decimal K10)

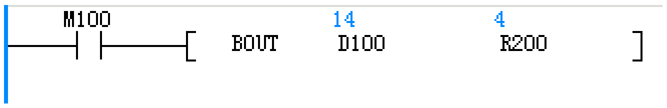
When R200 = 2 and M100 = ON, bit 2 of D100 is set to get the result D100 = 2#1110 (decimal K14).



When R200 = 4 and M100 = ON, bit 4 of D100 is set to get the result D100 = 2#11110 (decimal K30).



When M100 = OFF, bit 4 of D100 is reset to get the result D100 = 2#1110 (decimal K14).



BSET : Bit data setting

◆ Overview

When the BSET instruction is driven, the element designated by this instruction is set to ON and remains so. The BRST instruction sets this element to OFF regardless of whether the BSET instruction is still driven.

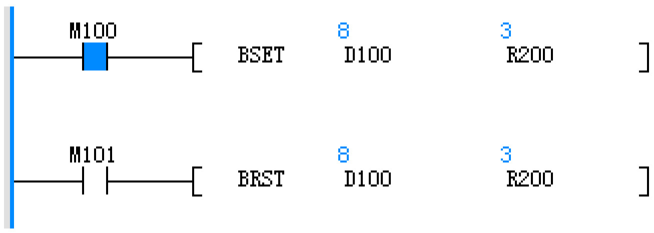
BSET D n		Bit data setting	Applicable model: H3U	
D	Output data	Element number of the output data	16-bit instruction (5 steps) BSET: Continuous execution	32-bit instruction (9 steps) DBSET: Continuous execution
n	Output bit	Designated output bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)		

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

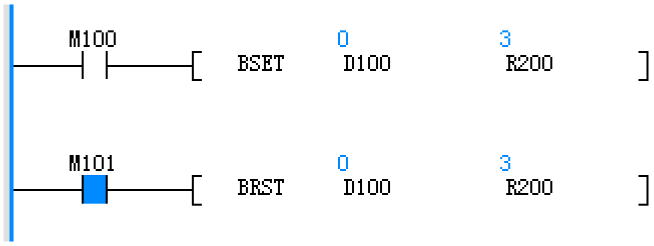
Note: The elements in gray background are supported.

◆ Application

When M100 = ON:



When M100 = OFF:



BRST: Bit data reset

◆ Overview

When the BRST instruction is driven, the bit designated by this instruction is set to OFF.

BRST D n			Bit data reset	Applicable model: H3U	
D	Output data	Element number of the output data		16-bit instruction (5 steps) BRST: Continuous execution	32-bit instruction (9 steps) DBRST: Continuous execution
n	Output bit	Designated output bit; value range: 0 to 15 (16-bit instruction) or 0 to 31 (32-bit instruction)			

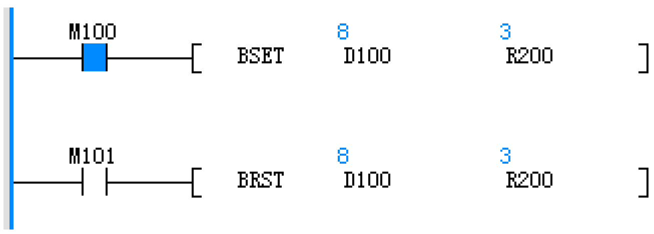
◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

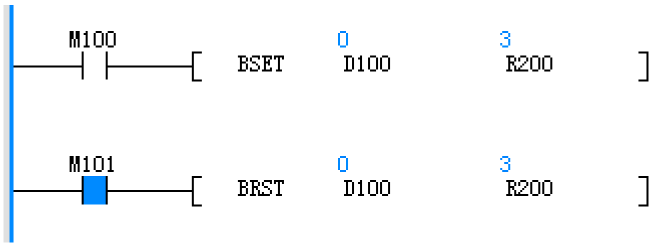
Note: The elements in gray background are supported.

◆ Application

When M100 = ON:



When M100 = OFF:



4.1.4 Main Control Instructions

Main control instructions	MC	Coil instruction for serial contacts used by the main control
	MCR	Release instruction for serial contacts used by main control reset

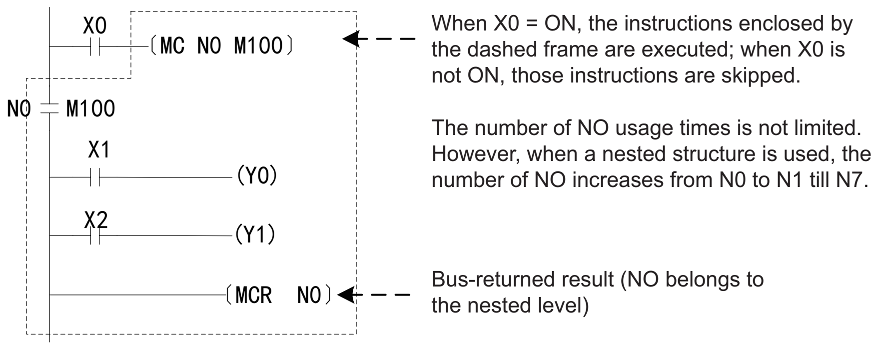
Instruction	Operand
MC MCR	N0–N7 markers (word parameters), 3 steps

MC is the main control starting instruction. When it is executed, the instructions between the MC and MCR instructions are executed normally. When the MC instruction is OFF, the instructions between the MC and MCR instructions trigger the following actions:

Timer	The timer is cleared, coils are de-energized, and contacts are inactive.
Counter	Coils are de-energized, whereas the counter and contacts remain unchanged.
Coils driven by the OUT instruction	No coils are energized.
Elements driven by the SET and RST instructions	The elements remain unchanged.
Application commands	All commands are inactive.

MCR is the main control ending instruction and located at the end of the main control program. This instruction must not be preceded by contact instructions.

The MC and MCR instructions of the main control program support the nested program structure with a maximum of eight layers, which are numbered from N0 to N7 and used in sequence.



Note: The main control instructions support the Y0–Y377 and M0–M7679 operands. The instructions do not support the M8000–M8511 operands and the S, SM, T, and C elements.

4.1.5 End Instructions

End instructions	FEND	End of the main program
	END	End of all programs

FEND: End of the main program

◆ Overview

The FEND instruction ends the main program.

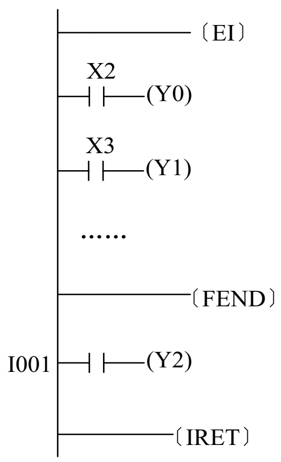
FEND	End of the main program	Applicable model: H3U	
An independent instruction without operands and contact driving		16-bit instruction (1 step) FEND: Continuous execution	

◆ Function

The FEND instruction is appended to the main program to indicate the end of the program. The FEND statement is located at the end of the main program. After this instruction is executed, the PLC ends a user program scan, returns to the 0-step program, and then scans the program again.

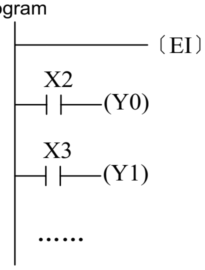
Compile the subprogram called by the CALL command after the FEND instruction, and add the SRET instruction at the end of the subprogram. Compile an interrupt subprogram after the FEDN instruction, and add the IRET instruction at the end of the interrupt subprogram. Compile a subprogram or interrupt program in an independent window in AutoShop. The FEDN instruction does not need to be added at the end of the main program, and the SRET or IRET instruction does not need to be added at the end of the subprogram or interrupt program.

Example:

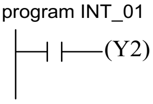


The process of compiling the left-hand program in AutoShop is as follows: Right-click the interrupt program INT_01 in the program block and choose Property, and then name 01 as needed.

Main program



Interrupt program INT_01



Right-click the interrupt program INT_01 in the program block, choose Property > Interrupt Event, and set it to I001.

END: End of all programs

◆ Overview

The END instruction ends all programs.

END	End of all programs	Applicable model: H3U	
An independent instruction without operands and contact driving		16-bit instruction (1 step) END: Continuous execution	

◆ Function

The END instruction is added only at the end of a ladder chart program or instruction program. The PLC scans the END instruction based on the address 0 of the user program, executes this instruction, and then returns to the address 0 for a new scan. The program space after the END instruction is not processed. In AutoShop, the FEND or END instruction is automatically added during the downloading process. Manual instruction input is not required.

4.1.6 Other Processing Instructions

Other processing instructions	NOP	No action
	WDT	Watchdog timer reset

Instruction	Operand
NOP	None

The NOP instruction does not perform any operation in a program; therefore, the original logical operation result is retained after this instruction is executed. This instruction is automatically deleted during the AutoShop programming process to reduce the waste of program space and increase the running speed.

WDT: Watchdog timer reset

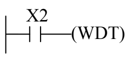
◆ Overview

The WDT instruction resets the watchdog timer.

WDT	Watchdog timer reset	**Applicable model:** H3U	
An independent instruction without operands		16-bit instruction (1 step) WDT: Continuous execution WDTP: Pulse execution	

◆ Function

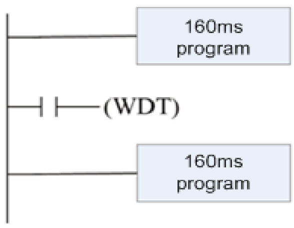
The PLC has a timer used to monitor the duration of a user program execution. If the execution times out, it is stopped and an alarm is generated. Use the WDT instruction to reset the watchdog timer, allowing it to start timing again and avoiding the timeout error.



A running timeout error may occur when a user program executes a complex operation (for example, an operation with many loop computations). To avoid this error, use the WDT instruction (for example, insert it between the FOR and NEXT instructions) when necessary during the programming process.

If the program scan duration exceeds the value in D8000 (default: 200 ms), insert the WDT instruction into the program to divide it into segments, each with a scan duration less than 200 ms, or change the value in D8000.

Example:



The scan duration of the program is 320 ms. Use the WDT instruction to divide the program into two segments, each with a scan duration less than 200 ms.

4.2 Program Flow Instructions

Subprogram	CALL	Subprogram call
	SRET	Subprogram return
	SSRET	Conditional subprogram return
	IRET	Interrupt return
Interrupt	EI	Enable interrupt
	DI	Disable interrupt
Jump	CJ	Conditional jump
	LBL	Marker instruction
	CJEND	Conditional jump to the program end
Loop	FOR	Start of a loop
	NEXT	End of a loop
Step sequential control	STL	Program jump to the secondary bus
	RET	Program return to the primary bus

4.2.1 Subprogram

CALL: Subprogram call

◆ Overview

The CALL instruction calls a subprogram.

CALL P000–P511			Subprogram call	Applicable model: H3U	
P	Pointer P	Target pointer number for subprogram call		16-bit instruction (3 steps) CALL: Continuous execution CALLP: Pulse execution	

◆ Function

When the flow is active, the program calls the subprogram designated by P***. After the subprogram is executed, the program returns to the next instruction of the CALL (or CALLP) statement to execute the subsequent statement.

The requirements for the P*** address pointer are as follows:

- The subprogram starting from P*** must be located after the end of the main program (which is ended by the FEND instruction).

- A subprogram must end with the SRET statement.
- The subprogram starting from P*** can be called in multiple locations or by another subprogram, but the number of nested layers cannot exceed five.
- A subprogram cannot be called within itself; otherwise, an infinite loop or program running timeout occurs.
- Subprograms support the T192–T199 or T246–T249 timers.

Subprograms are programmed in an independent window in AutoShop, which eliminates the problems of the FEND and SRET instructions. The names (including Chinese characters) of subprograms can be modified according to your need.

SRET: Subprogram return

◆ Overview

The SRET instruction forces the PLC to return to a subprogram.

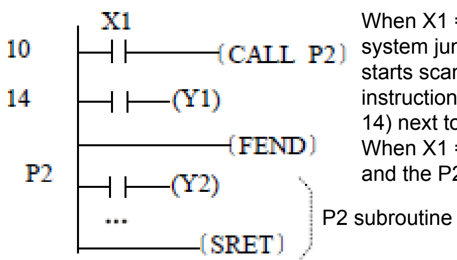
SRET	Subprogram return	Applicable model: H3U	
An independent instruction without operands and contact driving		16-bit instruction (1 step) SRET: Continuous execution	

◆ Function

The SRET statement is located at the end of a subprogram. After the SRET instruction is executed, the PLC returns to the statement that calls the subprogram and continues program execution.

In AutoShop, the SRET instruction does not need to be programmed at the end of a subprogram.

Example 1:

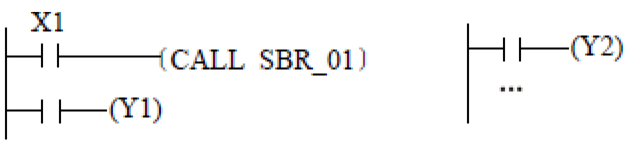


When X1 = ON, the CALL instruction is executed. The system jumps to the P2 designated subprogram and starts scanning. When scanning proceeds to the SRET instruction, the system returns to the instruction (address: 14) next to the CALL instruction and continues scanning. When X1 = OFF, the CALL instruction is not executed and the P2 subprogram is not called.

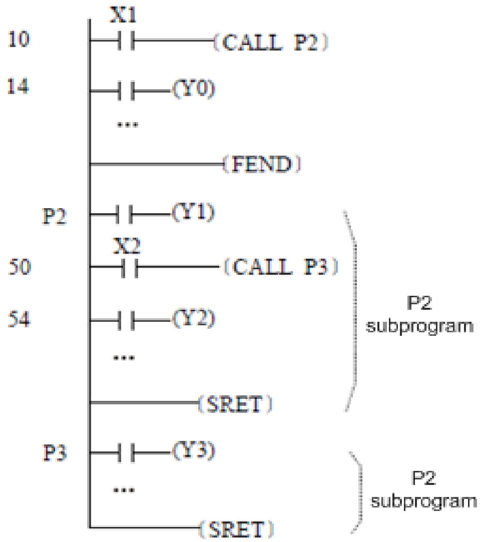
The format of the previous example instruction in AutoShop is as follows:

Right-click the subprogram SBR_01 in the program block and choose **Property**, and then name 01 as needed.

Main program Subprogram: SBR_01



Example 2:



SSRET: Conditional subprogram return

◆ Overview

The SSRET instruction executes a conditional return to a subprogram.

SSRET	Conditional subprogram return	Applicable model: H3U	
An independent instruction without operands and contact driving		16-bit instruction (1 step) SSRET: Continuous execution	

◆ Function

Omitted.

IRET: End of an interrupt program

◆ Overview

The IRET instruction ends an interrupt program.

IRET	End of an interrupt program	Applicable model: H3U	
An independent instruction without operands and contact driving		16-bit instruction (1 step) IRET, EI, and DI: Continuous execution	

◆ Function

The IRET statement is located at the end of an interrupt subprogram. After the IRET instruction is executed, the PLC returns to the statement that calls the interrupt subprogram and continues program execution. Interrupt programs are programmed in an independent window in AutoShop. The IRET instruction does not need to be programed at the end of an interrupt program.

4.2.2 Interrupt

EI: Enable interrupt; DI: Disable interrupt

◆ Overview

The EI instruction enables the interrupt function, whereas the DI instruction disables the interrupt function.

4

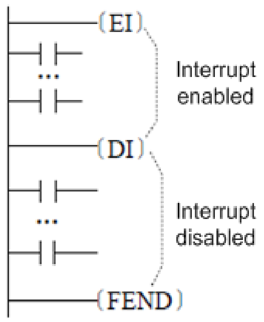
EI	**Enable interrupt**	**Applicable model:** H3U	
DI	**Disable interrupt**		
An independent instruction without operands and contact driving		16-bit instruction (1 step) IRET, EI, and DI: Continuous execution	

◆ Function

When the PLC program starts running, the interrupt function is disabled by default. Use the EI statement to enable the interrupt function, or use the DI statement to disable the interrupt function. The DI instruction is not required when the program does not have the interrupt insertion disabled range.

Interrupt types and setting:

1) External signal input interrupt: classified into the rising pulse interrupt and falling pulse interrupt triggered by input signals X0–X5. The pulse capture function can be enabled for the X signal that does not require instant response.
2) High-speed counter interrupt: used with the FNC53 (DHSCS) instruction for comparison setting. This interrupt is generated when the high-speed counter value equals the designated value.
3) Timer interrupt: an interrupt that occurs with a fixed cycle of 1–99 ms.
4) Pulse output complete interrupt: an interrupt that is executed immediately after the designated number of pulses are sent.
5) Multi-user interrupt (not available in the XP model): A maximum of 24 interrupts can be implemented by any high-speed counter.



External signal input interrupt pointers and setting (H1U-XP and H2U-XP):

Input Number	Pointer Number		DI Instruction
	Rising pulse interrupt	Falling pulse interrupt	
X000	I001	I000	M8050
X001	I101	I100	M8051
X002	I201	I200	M8052
X003	I301	I300	M8053
X004	I401	I400	M8054
X005	I501	I500	M8055

Timer interrupt pointers and setting (H1U-XP and H2U-XP):

Input Number	Interrupt Cycle **MS**	DI Instruction
I6□□	Enter values in the range 1–99 in the two blank boxes (□□) of the instruction name. For example, I605 indicates timer interrupt execution every 5 ms.	M8056
I7□□		M8057
I8□□		M8058

High-speed counter interrupt pointers and setting (H1U-XP and H2U-XP):

Input Number	DI Instruction
I010	M8059
I020	
I030	
I040	
I050	
I060	

Pulse output complete interrupt pointers and setting (H1U-XP and H2U-XP):

Port Number	Special Bit in Use	User Interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

Note: An interrupt is generated upon pulse output completion only when M8090–M8094 are enabled.

The different numbers selected by an interrupt subprogram correspond to different ports and interrupt trigger edges.

The rising pulse interrupt and falling pulse interrupt triggered by external signal input cannot be both numbered for the same X signal input. Only one trigger edge can be applied to the same X input port, and the trigger edge is set through pointer numbering.

External signal input interrupt: If M8050–M8055 are set to ON during program execution, the interrupt function is disabled for the corresponding X port.

Timer interrupt: If M8056–M8058 are set to ON during program execution, the interrupt function is disabled for the corresponding timer.

High-speed counter interrupt: If M8059 is set to ON during program execution, the interrupt function is disabled for all high-speed counters.

Interrupt programming specification and execution features:

- An interrupt that occurs within the interrupt disabled range (between the DI and EI instructions) can be memorized and executed after the EI instruction.
- An interrupt subprogram must be programmed after the FEND instruction and appended with IRET. In AutoShop, an interrupt subprogram cannot be programmed in the main program, and IRET can be omitted at the end of the subprogram.
- Pointer numbers cannot be used repeatedly.
- When multiple interrupts occur in sequence, the interrupt that occurs first takes precedence over the interrupt that occurs later. When multiple interrupts occur simultaneously, the interrupt with a higher priority takes precedence over that with a lower priority. Interrupts are sorted by priority in descending order as follows: high-speed counter interrupt, external signal input interrupt, timer interrupt, and pulse output complete interrupt.
- Other interrupts are disabled when an interrupt routine is executed.
- When the input relay and output relay are controlled during interrupt processing, the I/O refresh instruction REFF can be used to read the latest input status or immediately output the operation result to achieve high-speed control.
- The number of the input relay used as an interrupt pointer cannot be the same as the ID of any application command within the same input range, such as the high-speed counter command and pulse density (FNC56) command.
- Timers T192–T199 used by routines are recommended for subprograms and interrupt routines. The timing function of general-purpose timers is disabled when used with subroutines and interrupt programs. Attention must be paid when using the 1-ms cumulative timer.
- The input filter feature of the input relay is automatically disabled if an external signal input interrupt pointer is specified in the format of I□0□. In this case, the REFE (FNC51) instruction and the special data register D8020 (used for input filter adjustment) are not required. The input filter of the input relay not used as an external signal input interrupt pointer remains effective for 10 ms (initial value).

The H2U model is added with 24 high-speed counter interrupts during running of a high-speed counter. Any high-speed counter can be designated to generate 24 interrupt responses. This function is called high-speed counter multi-user interrupt (not available in the XP model). The setting principle is as follows:

Flag	Description
M8084	If it is set to ON, the high-speed counter multi-user interrupt function is enabled. (This function is not available in the XP model.)
D8084	Indicates a high-speed counter number in the range C235 to C255.
D8085	Indicates the number of user interrupts. A maximum of 24 user interrupts are supported and numbered from I507 to I530.
D8086	Indicates the serial numbers of multiple comparative values and is only applicable to the D element. For example, 200 is a double word from head address D200.

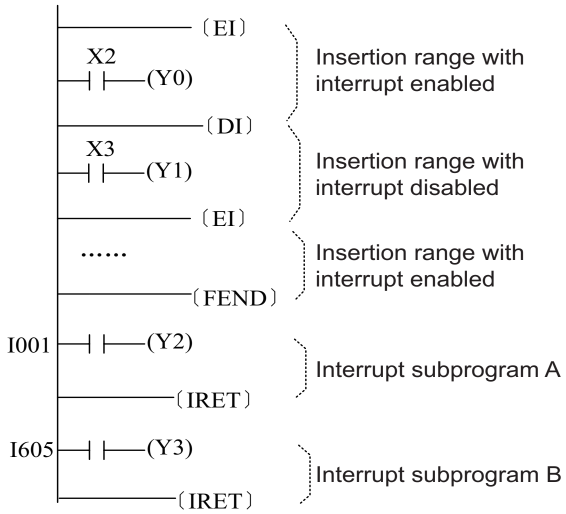
Example of comparative value placement:

D8084 = 235; D8086 = 200; D8085 = 5; M8084 = ON

C235 Data	Record Unit	Stored Unit Value	User Interrupt	**D8131** Value
100	D200 and D201	= 100	I507	0
200	D202 and D203	= 200	I508	1
300	D204 and D205	= 300	I509	2
400	D206 and D207	= 400	I510	3
500	D208 and D209	= 500	I511	4 -> 0 (M8133 = ON)

Each interrupt is generated based on the high-speed counter value and record unit value.

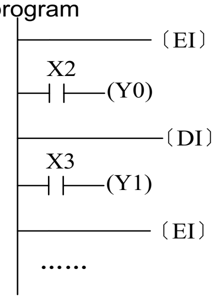
Example:



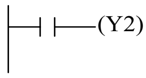
During the PLC's execution process, the interrupt insertion subprogram A or B is executed when any of the following conditions is met: the scanning proceeds to a point between EI and DI instructions; X0 is set to ON; the timer 5MS times out. When subprogram execution proceeds to IRET, the PLC returns to the main program and continues execution.

The process of compiling the left-hand program in AutoShop is as follows: Right-click the interrupt program INT_01 or INT_02 in the program block and choose Property, and then name 01 or 02 as needed.

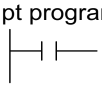
Main program



Interrupt program INT_01



Interrupt program INT_02



Right-click the interrupt program INT_01 and INT_02 in the program block, choose Property > Interrupt Event, and set them to I001 and I605.

4.2.3 Jump

CJ: Conditional jump

◆ Overview

The CJ instruction executes a program jump when conditions are met.

CJ/CJP P000–P511			Conditional jump	Applicable model: H3U	
P	Pointer P	Target pointer number for conditional jump		16-bit instruction (3 steps) CJ: Continuous execution CJP: Pulse execution	

Note: L can be used as an operand and is equivalent to P.

◆ Function

1) When the flow is active, the CJ (or CJP) instruction forces a program to jump from the instruction address to the address designated by P***. Program execution continues after the jump by skipping the program instructions within the intermediate address range.

2) When the flow is inactive, the program is executed without jump. The CJ (or CJP) instruction is not executed.

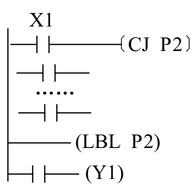
If the program has a TMR timer or counter within the intermediate address range and the timer or counter has been driven, the following actions are triggered:

Execution	CJ Jump Occurred	CJ Jump Not Occurred
T192–T199	Normally executed	Normally executed
Other timers	Timing stopped	
C235–C255	Normally executed	
Other counters	Counting stopped	

The requirements for the P*** address pointer are as follows:

- The CJ instruction must be used with the LBL instruction, and the target pointer number must belong to the current program block. Jumping across program blocks is prohibited.
- The addresses defined by P*** cannot be the same in the same program block.
- Use the CJ instruction to avoid the double coil problem when part of a program does not need to be executed or two coils are used for output.
- The CJ instruction can designate the same pointer P multiple times.

Example:
The CJ instruction is used as follows in AutoShop:



Subprograms and interrupt programs are programmed in independent windows, which eliminates the need for the FEND instruction. The instruction for jumping to the program end is CJEND in AutoShop.

LBL: Marker instruction

◆ Overview

The LBL instruction is used with the CJ instruction to mark the jump destination.

LBL P000–P511			Conditional jump	Applicable model: H3U
P	Pointer P	Target pointer number for conditional jump	16-bit instruction (3 steps) LBL: Continuous execution	

Note: L can be used as an operand and is equivalent to P.

CJEND: Conditional jump to the program end

◆ Overview

The CJEND instruction executes a jump to the program end when conditions are met. Then the current scan cycle ends.

CJEND	Conditional jump to the program end	Applicable model: H3U	
An independent instruction without operands and contact driving		16-bit instruction (3 steps) DBAND: Continuous execution	

4.2.4 Loop

FOR: Start of a loop

◆ Overview

The FOR instruction identifies the start position of a loop.

FOR S1			Start of a loop	Applicable model: H3U	
S1	Number of repeats for a loop	Number of repeats for a loop program		16-bit instruction (3 steps) FOR: Continuous execution	

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The FOR instruction identifies the start position and the number of repeats for a loop. It must be used with the NEXT instruction.

S1 is the variable that controls the number of repeats for the loop.

For details, see the description of the NEXT instruction.

NEXT: End of a loop

◆ Overview

The NEXT instruction identifies the end position of a loop.

NEXT	End of a loop	Applicable model: H3U	
An independent instruction without operands		16-bit instruction (1 step) NEXT: Continuous execution	

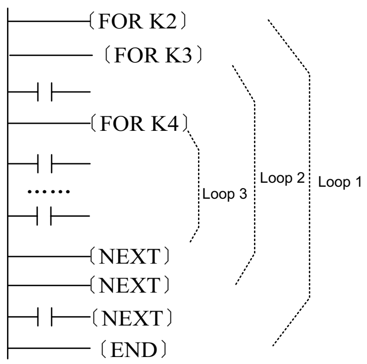
◆ Function

The NEXT instruction identifies the end position of a loop area. After a For/Next loop indicated by the FOR instruction is repeated N times, the PLC proceeds to subsequent execution.

The For/Next loop can be nested for six levels starting from the outermost level. The PLC executes parsing at the six levels in sequence. When the number of repeats for a loop is great, the PLC scan duration increases, which may result in an error when the watchdog timer times out. To avoid this error, insert the WDT instruction between the FOR and NEXT instructions.

- An error occurs in the following conditions: The NEXT instruction precedes the FOR instruction.
- The FOR instruction exists without the NEXT instruction.
- In FEND, the END instruction is followed by the NEXT instruction.
- The FOR and NEXT instructions are not equal in quantity.

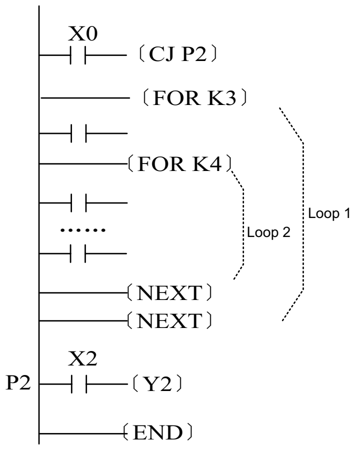
Example 1:



After loop 1 is executed twice, it will be executed in the program following the NEXT instruction. Loop 2 is executed three times each time loop 1 is executed, and loop 3 is executed four times each time loop 2 is executed. Therefore, loop 3 is executed 24 times (= 2 x 3 x 4) in total, and loop 2 is executed six times (= 2 x 3) in total.

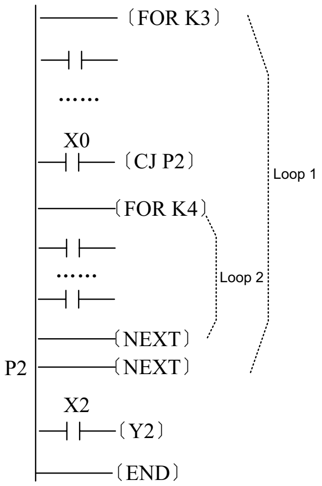
4

Example 2:



Use the CJ instruction to skip FOR–NEXT instructions. When X0 = OFF, loops 1 and 2 are executed. When X0 = ON, the CJ instruction jumps to P2. The program between loops 1 and 2 is not executed.

Example 3:



Use the CJ instruction to skip FOR–NEXT instructions. When X0 = OFF, loop 2 inside loop 1 is executed. When X0 = ON, the CJ instruction jumps to P2. FOR–NEXT instructions of loop 2 nested in loop 1 are not executed.

4.2.5 Step Sequential Control

◆ Overview

SLT	Program jump to the secondary bus	Applicable model: H3U Operand type: S Number of steps: 1 step
RET	Program return to the primary bus	

◆ Function

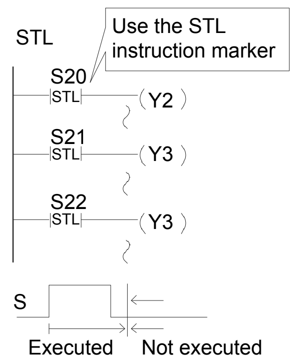
STL instructions (STL and RET)

STL divides the running process of a controlled device into several states or procedures, performs logical programming on each state, and then executes state switch based on the signal condition. STL programming simplifies logical design and makes commissioning and maintenance easier.

STL instructions can be represented by a ladder chart, where the state (S) is considered as a control procedure used for the sequential programming of input conditions and output control. This type of control separates the ongoing procedure from the preceding procedure and implements device control by executing various procedures in sequence.

STL and ladder charts differ in programming.

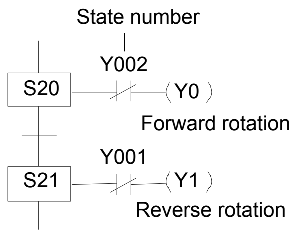
- An STL program starts with the STL instruction (which is different from S used in ladder charts) and ends with the RET instruction. The intermediate programs are guided by the S state. The operation logic of the S state is switched to the next state when conditions are met.



If the S contact of the STL instruction is connected, the circuit connected to this contact becomes active. If the S contact is disconnected, the circuit becomes inactive. The instruction is no longer executed (in the jump state) after a scan cycle.

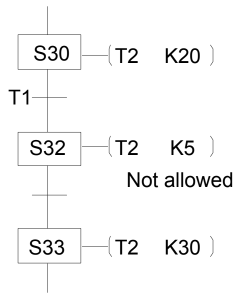
Different S states may correspond to the same output element (for example, Y3). When S21 or S22 is connected, Y3 is output. The issue of dual coil processing also exists in the same S state. Special attention is required.

The S state number cannot be used repeatedly.



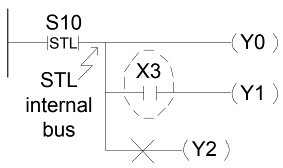
☆ **Output interlock:**

Two states are both connected for a period of time (one scan cycle) during state transition. To prevent simultaneous connection of a pair of outputs that cannot be connected at the same time, configure external interlock for the PLC and configure interlock for the corresponding program.



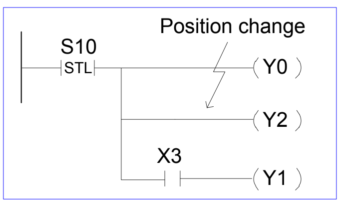
☆ **Repeated use of timers**

Like output coils, timer coils can be used to program the same element in different states, but programming in adjacent states is not allowed; otherwise, timer coils are not disconnected during process transfer and the current value cannot be reset.

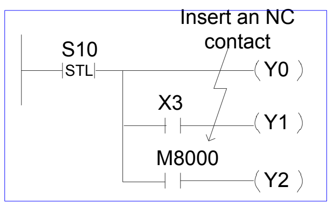


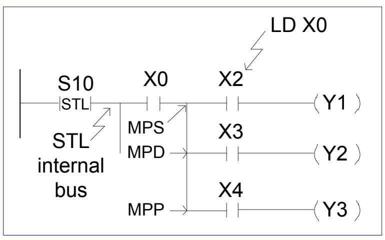
☆ **Output driving**

After the LD or LDI instruction is written to the intra-state bus, instructions that do not need contacts can no longer be used, as shown in the figure on the left. Use the method shown in the following figure to modify the circuit.



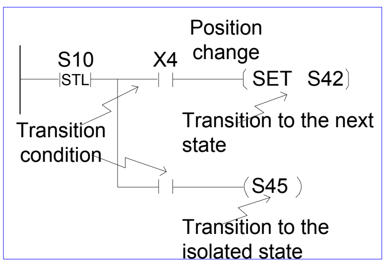
Or





☆ Locations of MPS, MRD, and MPP for stack operation

The MPS, MRD, and MPP instructions cannot be used directly in the STL internal bus in intra-state mode. A program must be compiled after the LD or LDI instruction, as shown in the figure on the left.



☆ **State transition method**

The OUT and SET instructions have the same function (automatic reset of the transition source) for the state (S) after the STL instruction. Both instructions have the self-hold function.

However, the OUT instruction executes transition to the isolated state in SFC.

The following table lists the sequential control commands with support for intra-state processing:

State \ Command		LD/LDI/LDP/LDF, AND/ANI/ ANDP/ANDF, OR/ORI/ORF, INV, OUT, SET/RST, and PLS/PLF	ANB/ORB MPS/MRD/MPP	MC/MCR
Initial and general states		Available	Available	Unavailable
Branching and merging states	Output processing	Available	Available	Unavailable
	Transfer processing	Available	Unavailable	Unavailable

- The STL instruction cannot be used within interrupt programs and subprograms.
- Though jump instructions are not prohibited within the STL instruction, these instructions are not recommended because of their complex actions.

4.3 Data Comparison

4.3.1 Contact Comparison

Contact comparison	LD=	LD contact comparison equal to	For details, see "LD#: Contact status bit operation" on Page 88
	LD>	LD contact comparison greater than	
	LD<	LD contact comparison less than	
	LD<>	LD contact comparison not equal to	
	LD>=	LD contact comparison greater than or equal to	
	LD<=	LD contact comparison less than or equal to	
	AND=	AND contact comparison equal to	For details, see "AND☼: Data comparison" on Page 86
	AND>	AND contact comparison greater than	
	AND<	AND contact comparison less than	
	AND<>	AND contact comparison not equal to	
	AND>=	AND contact comparison greater than or equal to	
	AND<=	AND contact comparison less than or equal to	
	OR=	OR contact comparison equal to	For details, see "OR☼: Data comparison" on Page 87
	OR>	OR contact comparison greater than	
	OR<	OR contact comparison less than	
	OR<>	OR contact comparison not equal to	
	OR>=	OR contact comparison greater than or equal to	
	OR<=	OR contact comparison less than or equal to	
	LD&	LD logical AND operation	Logical operation is performed bit by bit. 32-bit (9-step) operation and 16-bit (5-step) operation are supported. Two operands are included in every operation. The execution result is used as the contact status.
	LD|	LD logical OR operation	
	LD^	LD logical XOR operation	
	AND&	LD logical AND operation	
	AND|	AND logical OR operation	
	AND^	AND logical XOR operation	
	OR&	OR logical AND operation	
	OR|	OR logical OR operation	
	OR^	OR logical XOR operation	

	FLD>	Enabled when floating-point number > comparative state contact S1 > S2	These instructions are similar to contact comparison instructions. The difference is that the former compare floating-point numbers and support only 32-bit operation because 16-bit floating-point numbers do not exist. This set of instructions support only continuous execution but not pulse execution. Every instruction uses floating-point numbers as operands and occupies nine steps.
	FLD>=	Enabled when floating-point number >= comparative state contact S1 ≥ S2	
	FLD<	Enabled when floating-point number < comparative state contact S1 < S2	
	FLD<=	Enabled when floating-point number <= comparative state contact S1 ≤ S2	
	FLD=	Enabled when floating-point number = comparative state contact S1 = S2	
	FLD<>	Enabled when floating-point number <> comparative state contact S1 ≠ S2	
	FAND>	Enabled when floating-point number > comparative AND state contact S1 > S2	
	FAND>=	Enabled when floating-point number >= comparative AND state contact S1 ≥ S2	
	FAND<	Enabled when floating-point number < comparative AND state contact S1 < S2	
	FAND<=	Enabled when floating-point number <= comparative AND state contact S1 ≤ S2	
	FAND=	Enabled when floating-point number = comparative AND state contact S1 = S2	
	FAND<>	Enabled when floating-point number <> comparative AND state contact S1 ≠ S2	
	FOR>	Enabled when floating-point number > comparative OR state contact S1 > S2	
	FOR>=	Enabled when floating-point number >= comparative OR state contact S1 ≥ S2	
	FOR<	Enabled when floating-point number < comparative OR state contact S1 < S2	
	FOR<=	Enabled when floating-point number <= comparative OR state contact S1 ≤ S2	
	FOR=	Enabled when floating-point number = comparative OR state contact S1 = S2	
	FOR<>	Enabled when floating-point number <> comparative OR state contact S1 ≠ S2	

Contact comparison	LDZ>	Enabled when absolute value > comparative state contact |S1 – S2| > |S3|	These instructions use three operands: S1, S2, and S3. The result of subtracting S2 from S1 is compared with the absolute value in S3. The comparison result determines whether to set a contact to ON or OFF. These instructions contain 16 or 32 bits and do not support pulse operation.
	LDZ>=	Enabled when absolute value >= comparative state contact |S1 – S2| ≥ |S3|	
	LDZ<	Enabled when absolute value < comparative state contact |S1 – S2| < |S3|	
	LDZ<=	Enabled when absolute value <= comparative state contact |S1 – S2| ≤ |S3|	
	LDZ=	Enabled when absolute value = comparative state contact |S1 – S2| = |S3|	
	LDZ<>	Enabled when absolute value <> comparative state contact |S1 – S2| ≠ |S3|	
	ANDZ>	Enabled when absolute value > comparative AND state contact |S1 – S2| > |S3|	
	ANDZ>=	Enabled when absolute value >= comparative AND state contact |S1 – S2| ≥ |S3|	
	ANDZ<	Enabled when absolute value < comparative AND state contact |S1 – S2| < |S3|	
	ANDZ<=	Enabled when absolute value <= comparative AND state contact |S1 – S2| ≤ |S3|	
	ANDZ=	Enabled when absolute value = comparative AND state contact |S1 – S2| = |S3|	
	ANDZ<>	Enabled when absolute value <> comparative AND state contact |S1 – S2| ≠ |S3|	
	ORZ>	Enabled when absolute value > comparative OR state contact |S1 – S2| > |S3|	
	ORZ>=	Enabled when absolute value >= comparative OR state contact |S1 – S2| ≥ |S3|	
	ORZ<	Enabled when absolute value < comparative OR state contact |S1 – S2| < |S3|	
	ORZ<=	Enabled when absolute value <= comparative OR state contact |S1 – S2| ≤ |S3|	
	ORZ=	Enabled when absolute value = comparative OR state contact |S1 – S2| = |S3|	
	ORZ<>	Enabled when absolute value <> comparative OR state contact |S1 – S2| ≠ |S3|	

LD☼: Contact comparison

◆ Overview

The LD☼ instruction compares two operands and outputs the comparison result as a logical state. The variables in comparison are processed as signed numbers.

LD☼ S1 S2			Contact data comparison	Applicable model: H3U	
S1	Comparand 1	Data source to be compared or data variable unit 1		16-bit instruction (5 steps) LD=: Continuous execution	32-bit instruction (9 steps) LDD=: Continuous execution
S2	Comparand 2	Data source to be compared or data variable unit 2			

Note: The ☼ comparison operator can be =, >, <, <>, >=, or <=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

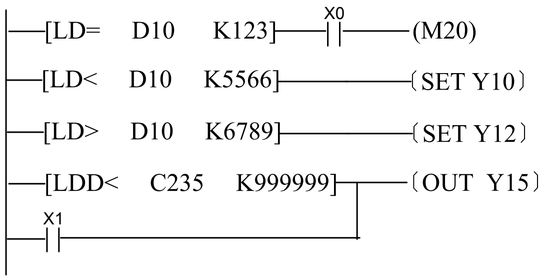
Note: The elements in gray background are supported.

◆ Function

The following table lists the methods of LD contact comparison.

16-bit Instruction	FNC NO	32-bit Instruction	Enabled State Condition	Disabled State Condition
LD=	224	LDD=	S1 = S2	S1 ≠ S2
LD>	225	LDD>	S1 > S2	S1 <= S2
LD<	226	LDD<	S1 < S2	S1 >= S2
LD<>	228	LDD<>	S1 <> S2	S1 = S2
LD<=	229	LDD<=	S1 <= S2	S1 > S2
LD>=	230	LDD>=	S1 >= S2	S1 < S2

Example:



M20 = ON when X0 = ON and D10 = K123.

Y10 = ON and remains in this state when X1 = ON and D10 < K5566.

Y12 = ON and remains in this state when D0 > K6 and D10 > K6789.

Y15 = ON and remains in this state when X2 = ON and C235 < K999999 or X3 = ON.

Use the 32-bit instruction LDD ☼ to compare 32-bit counters (C200–C255). If a different instruction is used, an error will occur.

4

AND☼ : Data comparison

◆ Overview

The AND☼ instruction compares two operands and outputs the comparison result as a logical state. The variables in comparison are processed as signed numbers.

AND☼ S1 S2			Contact data comparison	Applicable model: H3U
S1	Comparand 1	Data source to be compared or data variable unit 1	16-bit instruction (5 steps) AND=: Continuous execution	32-bit instruction (9 steps) ANDD=: Continuous execution
S2	Comparand 2	Data source to be compared or data variable unit 2		

Note: The AND ☼ instruction is preceded by other logical operations. It compares two operands and outputs the comparison result as a logical state, which is used for a program flow operation. The variables in comparison are processed as signed numbers. The ☼ comparison operator can be =, >, <, >=, <=, or <>.

◆ Operands

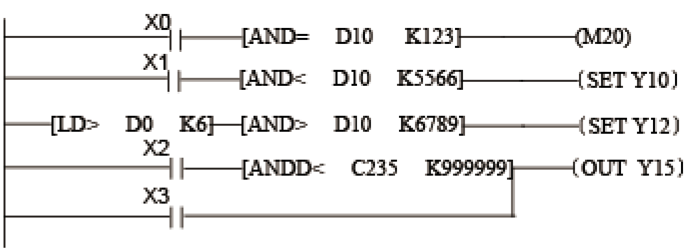
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

The following table lists the methods of AND contact comparison.

16-bit Instruction	FNC NO	32-bit Instruction	Enabled State Condition	Disabled State Condition
AND=	232	ANDD=	S1 = S2	S1 ≠ S2
AND>	233	ANDD>	S1 > S2	S1 <= S2
AND<	234	ANDD<	S1 < S2	S1 >= S2
AND<>	236	ANDD<>	S1 <> S2	S1 = S2
AND<=	237	ANDD<=	S1 <= S2	S1 > S2
AND>=	238	ANDD>=	S1 >= S2	S1 < S2

Example:



M20 = ON when X0 = ON and D10 = K123.
Y10 = ON and remains in this state when X1 = ON and D10 < K5566.
Y12 = ON and remains in this state when D0 > K6 and D10 > K6789.
Y15 = ON and remains in this state when X2 = ON and C235 < K999999 or X3 = ON.

Use the 32-bit instruction ANDD ☼ to compare 32-bit counters (C200–C255). If a different instruction is used, an error will occur.

OR ☼ : Data comparison

◆ Overview

The OR☼ instruction compares two operands and outputs the comparison result as a logical state. The variables in comparison are processed as signed numbers.

OR☼ S1 S2			Parallel contact data comparison	Applicable model: H3U
S1	Comparand 1	Data source to be compared or data variable unit 1	16-bit instruction (5 steps) OR=: Continuous execution	32-bit instruction (9 steps) ORD=: Continuous execution
S2	Comparand 2	Data source to be compared or data variable unit 2		

The ☼ comparison operator can be =, >, <, >=, <=, or <>.

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

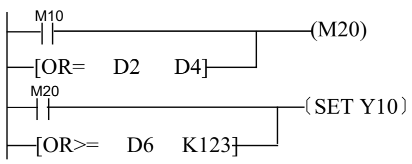
Note: The elements in gray background are supported.

◆ Function

The following table lists the methods of OR contact comparison.

16-bit Instruction	FNC NO	32-bit Instruction	Enabled State Condition	Disabled State Condition
OR=	240	ORD=	S1 = S2	S1 ≠ S2
OR>	241	ORD>	S1 > S2	S1 <= S2
OR<	242	ORD<	S1 < S2	S1 >= S2
OR<>	244	ORD<>	S1 <> S2	S1 = S2
OR<=	245	ORD<=	S1 <= S2	S1 > S2
OR>=	246	ORD>=	S1 >= S2	S1 < S2

Example:



M20 = ON when M10 = ON or D2 = D4.

Y10 = ON and remains in this state when M20 = ON or D6 > K123.

Use the 32-bit instruction ORD ☼ to compare 32-bit counters (C200–C255). If a different instruction is used, an error will occur.

LD#: Contact status bit operation

◆ Overview

Whether the LD# instruction is enabled is determined based on the bit logical operation result (a node directly connected to the left-hand bus).

LD# S1 S2		Contact status bit operation	Applicable model: H3U	
S1	Data 1	Element number of source data 1	16-bit instruction (5 steps) LD#: Continuous execution	32-bit instruction (9 steps) LDD#: Continuous execution
S2	Data 2	Element number of source data 2		

Note: The # comparison operator can be &, |, or ^.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

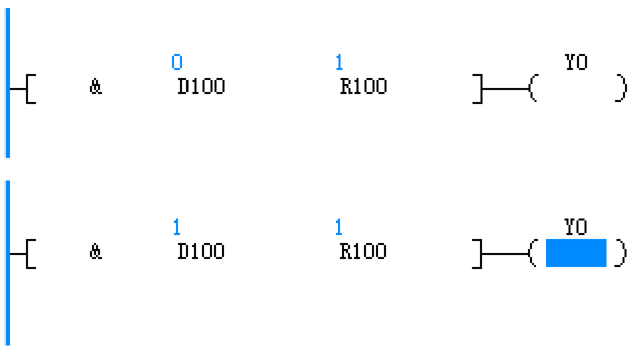
Note: The elements in gray background are supported.

◆ Function

A logical operation (AND: &; NOT: |; XOR: ^) is performed on [S1] and [S2]. The instruction is enabled if the operation result is not 0; if the operation result is 0, the instruction is disabled. The execution results are as follows:

16-bit Instruction	32-bit Instruction	Enabled State Condition	Disabled State Condition
LD&	LDD&	S1&S2 ≠ 0	S1&S2 = 0
LD|	LDD|	S1|S2 ≠ 0	S1|S2 = 0
LD^	LDD^	S1^S2 ≠ 0	S1^S2 = 0

◆ Application



AND#: AND contact status bit operation

◆ Overview

Whether the AND# instruction is enabled is determined based on the bit logical operation result (a node connected to another node in series).

AND# S1 S2			AND contact status bit operation	Applicable model: H3U	
S1	Data 1	Element number of source data 1		16-bit instruction (5 steps) AND#: Continuous execution	32-bit instruction (9 steps) ANDD#: Continuous execution
S2	Data 2	Element number of source data 2			

Note: The # comparison operator can be &, |, or ^.

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

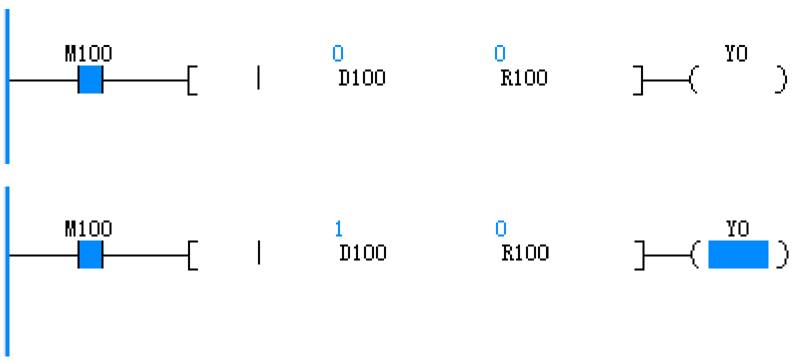
Note: The elements in gray background are supported.

◆ Function

A logical operation (AND: &; NOT: |; XOR: ^) is performed on [S1] and [S2]. The instruction is enabled if the operation result is not 0; if the operation result is 0, the instruction is disabled. The execution results are as follows:

16-bit Instruction	32-bit Instruction	Enabled State Condition	Disabled State Condition
AND&	ANDD&	S1&S2 ≠ 0	S1&S2 = 0
AND|	ANDD|	S1|S2 ≠ 0	S1|S2 = 0
AND^	ANDD^	S1^S2 ≠ 0	S1^S2 = 0

◆ Application



OR#: OR contact status bit operation

◆ Overview

Whether the OR# instruction is enabled is determined based on the bit logical operation result (a node connected to another node in parallel).

OR# S1 S2			OR contact status bit operation	Applicable model: H3U	
S1	Data 1	Element number of source data 1		16-bit instruction (5 steps) OR#: Continuous execution	32-bit instruction (9 steps) ORD#: Continuous execution
S2	Data 2	Element number of source data 2			

Note: The # comparison operator can be &, |, or ^.

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

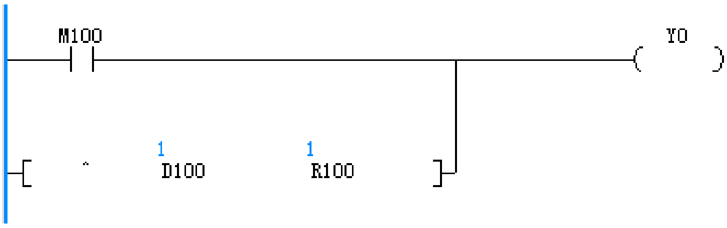
Note: The elements in gray background are supported.

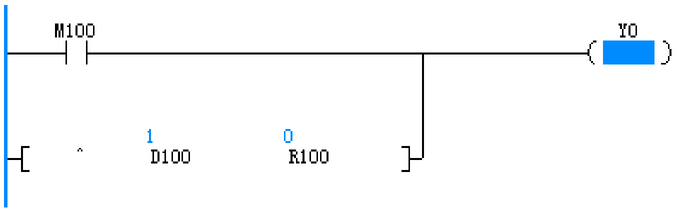
◆ Function

A logical operation (AND: &; NOT: |; XOR: ^) is performed on [S1] and [S2]. The instruction is enabled if the operation result is not 0; if the operation result is 0, the instruction is disabled. The execution results are as follows:

16-bit Instruction	32-bit Instruction	Enabled State Condition	Disabled State Condition
OR&	ORD&	S1&S2 ≠ 0	S1&S2 = 0
OR|	ORD|	S1|S2 ≠ 0	S1|S2 = 0
OR^	ORD^	S1^S2 ≠ 0	S1^S2 = 0

◆ Application





FLD#: Floating-point contact comparison

◆ Overview

The FLD# instruction compares two operands and uses the comparison result to determine whether to set a contact to ON or OFF (a node directly connected to the left-hand bus).

FLD# S1 S2			Floating-point contact comparison	Applicable model: H3U	
S1	Data 1	Element number of source data 1			32-bit instruction (9 steps) FLDD#: Continuous execution
S2	Data 2	Element number of source data 2			

Note: The # comparison operator can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

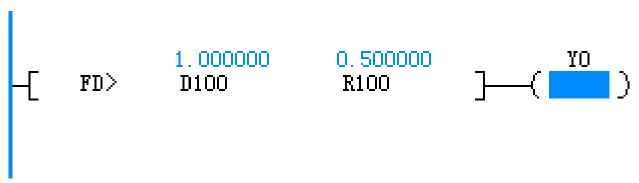
Note: The elements in gray background are supported.

◆ Function

[S1] and [S2] are compared. The instruction is enabled when conditions are met; otherwise, it is disabled.

32-bit Instruction	Enabled State Condition	Disabled State Condition
FLDD>	S1 > S2	S1 <= S2
FLDD>=	S1 >= S2	S1 < S2
FLDD<	S1 < S2	S1 >= S2
FLDD<=	S1 <= S2	S1 > S2
FLDD=	S1 = S2	S1 <> S2
FLDD<>	S1 <> S2	S1 = S2

◆ Application



FAND#: Floating-point AND contact comparison

◆ Overview

The FAND# instruction compares two operands and uses the comparison result to determine whether to set a contact to ON or OFF (a node connected to another node in series).

FAND# S1 S2			Floating-point AND contact comparison	Applicable model: H3U	
S1	Data 1	Element number of source data 1			32-bit instruction (9 steps) FANDD#: Continuous execution
S2	Data 2	Element number of source data 2			

Note: The # comparison operator can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

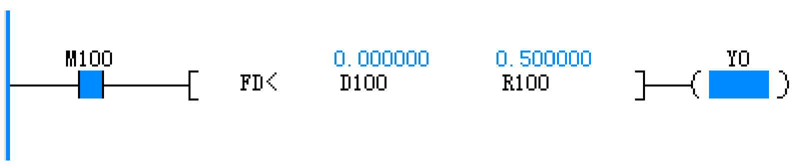
Note: The elements in gray background are supported.

◆ Function

[S1] and [S2] are compared. The instruction is enabled when conditions are met; otherwise, it is disabled.

32-bit Instruction	Enabled State Condition	Disabled State Condition
FANDD>	S1 > S2	S1 <= S2
FANDD>=	S1 >= S2	S1 < S2
FANDD<	S1 < S2	S1 >= S2
FLD<=	S1 <= S2	S1 > S2
FANDD=	S1 = S2	S1 <> S2
FANDD<>	S1 <> S2	S1 = S2

◆ Application



FOR#: Floating-point OR contact comparison

◆ Overview

The FOR# instruction compares two operands and uses the comparison result to determine whether to set a contact to ON or OFF (a node connected to another node in parallel).

FOR# S1 S2			Floating-point OR contact comparison	Applicable model: H3U	
S1	Data 1	Element number of source data 1			32-bit instruction (9 steps) FORD#: Continuous execution
S2	Data 2	Element number of source data 2			

Note: The # comparison operator can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

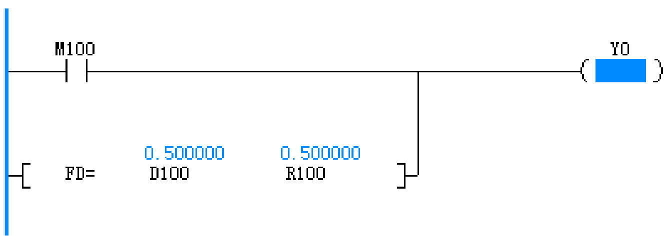
Note: The elements in gray background are supported.

◆ Function

[S1] and [S2] are compared. The instruction is enabled when conditions are met; otherwise, it is disabled.

32-bit Instruction	Enabled State Condition	Disabled State Condition
FORD>	S1 > S2	S1 <= S2
FORD>=	S1 >= S2	S1 < S2
FORD<	S1 < S2	S1 >= S2
FORD<=	S1 <= S2	S1 > S2
FORD=	S1 = S2	S1 <> S2
FORD<>	S1 <> S2	S1 = S2

◆ Application



LDZ#: Absolute value comparison contact

◆ Overview

The LDZ# instruction compares the absolute value of the S1 and S2 subtraction result with the absolute value in S3 and uses the comparison result to determine whether to set a contact to ON or OFF (a node directly connected to the left-hand bus).

LDZ# S1 S2 S3		Absolute value comparison contact	Applicable model: H3U	
S1	Subtrahend	Source element of the subtrahend	16-bit instruction (5 steps) LDZ#: Continuous execution	32-bit instruction (9 steps) LDDZ#: Continuous execution
S2	Minuend	Source element of the minuend		
S3	Comparative value	Source element of the comparative value		

Note: The # comparison operator can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The absolute value of the [S1] and [S2] subtraction result is compared with the absolute value in [S3]. The instruction is enabled when conditions are met; otherwise, it is disabled.

16-bit Instruction	32-bit Instruction	Enabled State Condition	Disabled State Condition
LDZ>	LDDZ>	|S1 – S2| > |S3|	|S1 – S2| <= |S3|
LDZ>=	LDDZ>=	|S1 – S2| >= |S3|	|S1 – S2| < |S3|
LDZ<	LDDZ<	|S1 – S2| < |S3|	|S1 – S2| >= |S3|
LDZ<=	LDDZ<=	|S1 – S2| <= |S3|	|S1 – S2| > |S3|
LDZ=	LDDZ=	|S1 – S2| = |S3|	|S1 – S2| <> |S3|
LDZ<>	LDDZ<>	|S1 – S2| <> |S3|	|S1 – S2| = |S3|

◆ Application

ANDZ#: Absolute value comparison AND contact

◆ Overview

The ANDZ# instruction compares the absolute value of the S1 and S2 subtraction result with the absolute value in S3 and uses the comparison result to determine whether to set a contact to ON or OFF (a node connected to another node in series).

ANDZ# S1 S2 S3			Absolute value comparison AND contact	Applicable model: H3U	
S1	Subtrahend	Source element of the subtrahend		16-bit instruction (5 steps) ANDZ#: Continuous execution	32-bit instruction (9 steps) ANDDZ#: Continuous execution
S2	Minuend	Source element of the minuend			
S3	Comparative value	Source element of the comparative value			

Note: The # comparison operator can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

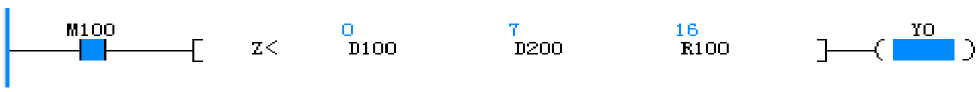
Note: The elements in gray background are supported.

◆ Function

The absolute value of the [S1] and [S2] subtraction result is compared with the absolute value in [S3]. The instruction is enabled when conditions are met; otherwise, it is disabled.

16-bit Instruction	32-bit Instruction	Enabled State Condition	Disabled State Condition
ANDZ>	ANDDZ>	|S1 – S2| > |S3|	|S1 – S2| <= |S3|
ANDZ>=	ANDDZ>=	|S1 – S2| >= |S3|	|S1 – S2| < |S3|
ANDZ<	ANDDZ<	|S1 – S2| < |S3|	|S1 – S2| >= |S3|
ANDZ<=	ANDDZ<=	|S1 – S2| <= |S3|	|S1 – S2| > |S3|
ANDZ=	ANDDZ=	|S1 – S2| = |S3|	|S1 – S2| <> |S3|
ANDZ<>	ANDDZ<>	|S1 – S2| <> |S3|	|S1 – S2| = |S3|

◆ Application



ORZ#: Absolute value comparison OR contact

◆ Overview

The ORZ# instruction compares the absolute value of the S1 and S2 subtraction result with the absolute value in S3 and uses the comparison result to determine whether to set a contact to ON or OFF (a node connected to another node in parallel).

ORZ# S1 S2 S3			Absolute value comparison OR contact	Applicable model: H3U	
S1	Subtrahend	Source element of the subtrahend		16-bit instruction (5 steps) ORZ#: Continuous execution	32-bit instruction (9 steps) ORDZ#: Continuous execution
S2	Minuend	Source element of the minuend			
S3	Comparative value	Source element of the comparative value			

4

Note: The # comparison operator can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

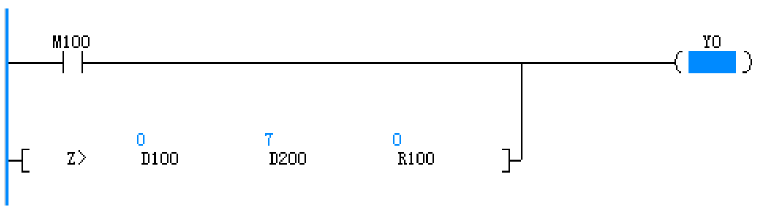
Note: The elements in gray background are supported.

◆ Function

The absolute value of the [S1] and [S2] subtraction result is compared with the absolute value in [S3]. The instruction is enabled when conditions are met; otherwise, it is disabled.

16-bit Instruction	32-bit Instruction	Enabled State Condition	Disabled State Condition
ORZ>	ORDZ>	|S1 – S2| > |S3|	|S1 – S2| <= |S3|
ORZ>=	ORDZ>=	|S1 – S2| >= |S3|	|S1 – S2| < |S3|
ORZ<	ORDZ<	|S1 – S2| < |S3|	|S1 – S2| >= |S3|
ORZ<=	ORDZ<=	|S1 – S2| <= |S3|	|S1 – S2| > |S3|
ORZ=	ORDZ=	|S1 – S2| = |S3|	|S1 – S2| <> |S3|
ORZ<>	ORDZ<>	|S1 – S2| <> |S3|	|S1 – S2| = |S3|

◆ Application



4.3.2 Comparison Output

Comparison output	CMP	Data comparison
	ECMP	Binary floating point comparison
	ZCP	Range comparison
	EZCP	Binary floating-point range comparison

CMP: Data comparison

◆ Overview

When driving conditions are met, the CMP instruction compares the values in S1 and S2 and then sets the end-address bit element D (D+1 or D+2) to ON based on the comparison result (S1 > S2, S1 = S2, or S1 < S2).

4

CMP S1 S2 D		Comparison of two values	Applicable model: H3U	
S1	Comparative value 1	Data of comparative value 1, or address of the word element that stores the data	16-bit instruction (7 steps) CMP: Continuous execution CMPP: Pulse execution	32-bit instruction (13 steps) DCMP: Continuous execution DCMPP: Pulse execution
S2	Comparative value 2	Data of comparative value 2, or address of the word element that stores the data		
D	Comparison result	Head address of three consecutive elements that store the comparison result (ON or OFF)		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

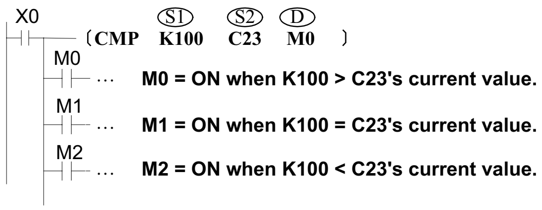
Note: The elements in gray background are supported.

◆ Function

The values of two operands are compared. The comparison result is output to the designated bit variable. Operands are processed as signed numbers in algebraic comparison.

D is a bit variable that occupies three consecutive addresses.

Example:



M0, M1, or M2 = ON when X0 = ON.
The CMP instruction is not executed when X0 switches from ON to OFF, and M0 to M2 remain in the state prior to X0 = OFF. Use the RST or ZRST instruction to clear the comparison result of M0 to M2.
Connect M0 to M2 in serial or parallel mode to acquire the ≥, ≤, and ≠ results.

ECMP: Binary floating point comparison

4

◆ Overview

The ECMP instruction compares the values of two floating-point variables and outputs the comparison result to three variables from head address D.

ECMP S1 S2 D		Binary floating point comparison	Applicable model: H3U	
S1	Comparand 1	Binary floating-point number 1 to be compared		32-bit instruction (13 steps) DECMP: Continuous execution
S2	Comparand 2	Binary floating-point number 2 to be compared		
D	Comparison result	Comparison result storage unit, which occupies three (bit) variables		

◆ Operands

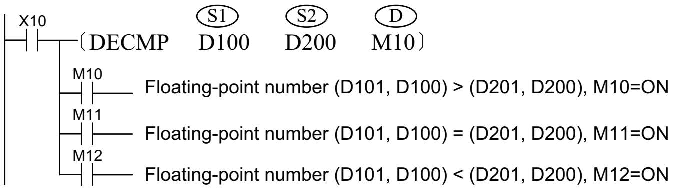
Operand	Bit Element System·User							Word Element System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The values of two floating-point variables are compared. The comparison result is output to three variables from head address Ⓓ.

Example:



When X10 = ON, M10, M11, or M12 switches to ON.
The DECP instruction is not executed when X10 switches from ON to OFF, and M10 to M12 remain in the state prior to X10 = OFF.
Use the RST or ZRST instruction to clear the comparison result of M10 to M12.
Connect M10 to M12 in serial or parallel mode to acquire the ≥, ≤, and ≠ results.

If the constant K or H is used as S1 or S2, the value is converted to a floating-point number before the comparison operation.

4

ZCP: Range comparison

◆ Overview

When driving conditions are met, the ZCP instruction sets the end-address bit element D (D+1 or D+2) to ON based on the range of the source address S (S < S1, S1 ≤ S ≤ S2, or S > S2).

ZCP S1 S2 S D			Range comparison	Applicable model: H3U
S1	Lower limit for range comparison	Data, or address of the word element that stores the data	16-bit instruction (9 steps) ZCP: Continuous execution ZCPP: Pulse execution	32-bit instruction (17 steps) DZCP: Continuous execution DZCPP: Pulse execution
S2	Upper limit for range comparison	Data, or address of the word element that stores the data		
S	Comparative variable	Data, or address of the word element that stores the data		
D	Comparison result	Head address of three consecutive elements that store the comparison result (ON or OFF)		

◆ Operands

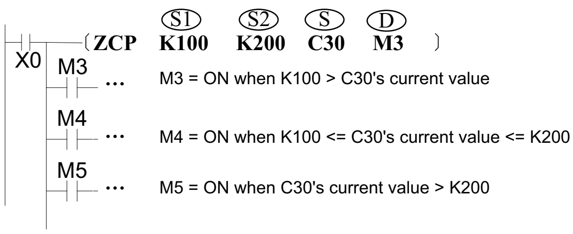
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The ZCP instruction requires contact driving and has four operands. When the control flow is active, use this instruction to algebraically compare operands as signed numbers. The comparison result is indicated by the position of S within the range defined by S1 and S2 and stored in three consecutive bit variables from head address D.

Example:



M3, M4, or M5 switches to ON when X0 = ON.
The ZCP instruction is not executed when X0 switches from ON to OFF, and M3 to M5 remain in the state prior to X0 = OFF. Use the RST or ZRST instruction to clear the comparison result of M3 to M5.

EZCP: Binary floating-point range comparison

◆ Overview

The EZCP instruction compares a binary floating-point variable range with a floating-point variable. The comparison result is output to three variables from head address D.

EZCP S1 S2 S D			Binary floating-point range comparison	Applicable model: H3U	
S1	Lower limit for range comparison	Lower limit of a binary floating-point variable range		16-bit instruction (17 steps) EZCP: Continuous execution	32-bit instruction (17 steps) DEZCPP: Jump execution
S2	Upper limit for range comparison	Upper limit of a binary floating-point variable range			
S	Comparand	Binary floating-point variable to be compared			
D	Comparison result	Comparison result storage unit, which occupies three (bit) variables			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

A binary floating-point variable range is compared with a floating-point variable. The comparison result is output to three variables from head address D.

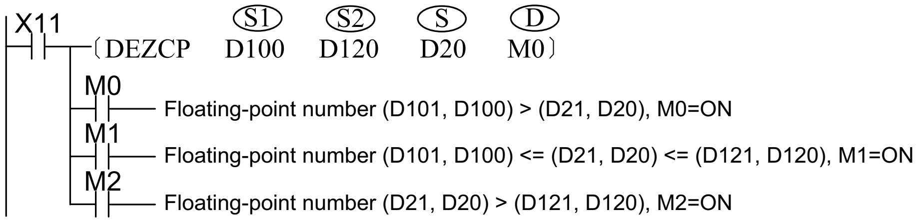
S1 is the lower limit of a binary floating-point variable range.

S2 is the upper limit of a binary floating-point variable range.

S is the binary floating-point variable to be compared.

D is the comparison result storage unit, which occupies three (bit) variables.

Example:



M0, M1, or M2 switches to ON when X11 = ON.
The DEZCP instruction is not executed when X11 switches from ON to OFF, and M0, M1, and M2 remain in the state prior to X11 = OFF.

4.4 Data Operation

Four arithmetic operations	ADD	Binary number addition
	SUB	Binary number subtraction
	MUL	Binary number multiplication
	DIV	Binary number division
	EADD	Binary floating point addition
	ESUB	Binary floating point subtraction
	EMUL	Binary floating point multiplication
	EDIV	Binary floating point division
	INC	Binary number incremented by 1
	DEC	Binary number decremented by 1
Logical operations	WAND	Binary number logical AND
	WOR	Binary number logical OR
	WXOR	Binary number logical XOR
	NEG	Binary number negation
	ENEG	Binary floating-point sign negation
Trigonometric functions	SIN	Floating point SIN operation
	COS	Floating point COS operation
	TAN	Floating point TAN operation
	ASIN	Binary floating point ARCSIN operation
	ACOS	Binary floating point ARCCOS operation
	ATAN	Binary floating point ARCTAN operation
	RAD	Binary floating point degree-to-radian conversion
	DEG	Binary floating point radian-to-degree conversion
	SINH	Binary floating point SINH operation
	COSH	Binary floating point COSH operation
	TANH	Binary floating point TANH operation

Table operations	WSUM	Value summation
	MEAN	Mean value calculation
	LIMIT	Upper/Lower limit control
	BZAND	Dead zone control
	ZONE	Zone control
	SCL	Coordinate determination (coordinates of different points)
	SCL2	Coordinate determination 2 (X and Y coordinates)
Exponent operations	EXP	Binary floating-point exponent operation
	LOGE	Binary floating-point natural logarithm operation
	LOG	Binary floating-point logarithm operation with a base of 10
	ESQR	Binary floating-point square root operation
	SQR	Binary number square root operation
	POW	Floating-point weight instruction

4.4.1 Four Arithmetic Operations

Four arithmetic operations	ADD	Binary number addition
	SUB	Binary number subtraction
	MUL	Binary number multiplication
	DIV	Binary number division
	EADD	Binary floating point addition
	ESUB	Binary floating point subtraction
	EMUL	Binary floating point multiplication
	EDIV	Binary floating point division
	INC	Binary number incremented by 1
	DEC	Binary number decremented by 1

ADD: Binary number addition

◆ Overview

The ADD instruction adds two binary numbers together.

ADD S1 S2 D			Binary number addition	Applicable model: H3U
S1	Augend	Data, or address of the word element that stores the data	16-bit instruction (7 steps) ADD: Continuous execution ADDP: Pulse execution	32-bit instruction (13 steps) DADD: Continuous execution DADDP: Pulse execution
S2	Addend	Data, or address of the word element that stores the data		
D	Sum	Address of the word element that stores the data		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The ADD instruction requires contact driving and has three operands. Use this instruction to algebraically add the values in S1 and S2 together in BIN mode. The result is stored in D. The variables in the algebraic operation are processed as signed numbers. The highest bit is the sign bit. The value 0 indicates a positive number, whereas the value 1 indicates a negative number.

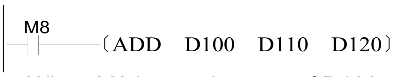
The zero flag M8020 is set if the result of the calculation is 0.

The carry flag M8022 is set if the result of the calculation is greater than 32,767 (for a 16-bit operation) or 2,147,483,647 (for a 32-bit operation).

The borrow flag M8021 is set if the result of the calculation is less than –32,768 (for a 16-bit operation) or –2,147,483,648 (for a 32-bit operation).

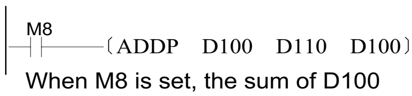
In 32-bit operation, the variable addresses in the ADD instruction contain the lower 16 bits, and the adjacent high-numbered address unit contains the higher 16 bits. Avoid repeated or overlapping addresses during programming.

Example 1:



When M8 is set, the sum of D100 (augend) and D110 (addend) is stored in D120. D120 = 8 + (–12) = k-4 if D100 = K8 and D110 = K-12.

Example 2:



When M8 is set, the sum of D100 (augend) and D110 (addend) is stored in D100.

SUB: Binary number subtraction

◆ Overview

The SUB instruction subtracts one binary number from another.

SUB S1 S2 D		**Binary number subtraction**	**Applicable model: H3U**	
S1	Subtrahend	Data, or address of the word element that stores the data	16-bit instruction (7 steps) SUB: Continuous execution SUBP: Pulse execution	32-bit instruction (13 steps) DSUB: Continuous execution DSUBP: Pulse execution
S2	Minuend	Data, or address of the word element that stores the data		
D	Difference	Address of the word element that stores the data		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The SUB instruction requires contact driving and has three operands. Use this instruction to algebraically subtract the value in S2 from that in S1 in BIN mode. The result is stored in D. The variables in the algebraic operation are processed as signed numbers. The highest bit is the sign bit. The value 0 indicates a positive number, whereas the value 1 indicates a negative number.

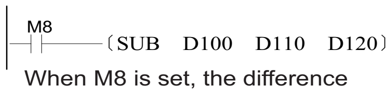
The zero flag M8020 is set if the result of the calculation is 0.

The carry flag M8022 is set if the result of the calculation is greater than 32,767 (for a 16-bit operation) or –2,147,483,647 (for a 32-bit operation).

The borrow flag M8021 is set if the result of the calculation is less than –32,768 (for a 16-bit operation) or –2,147,483,648 (for a 32-bit operation).

In 32-bit operation, the variable addresses in the SUB instruction contain the lower 16 bits, and the adjacent high-numbered address unit contains the higher 16 bits. Avoid repeated or overlapping addresses during programming.

Example:



When M8 is set, the difference between D100 (subtrahend) and D110 (minuend) is stored in D120. D120 = 10 – 8 = K2 if D100 = K10 and D110 = K8.

4

MUL: Binary number multiplication

◆ Overview

The MUL instruction multiplies two binary numbers together.

MUL S1 S2 D			Binary number multiplication	Applicable model: H3U	
S1	Multiplicand	Data, or address of the word element that stores the data		16-bit instruction (7 steps) MUL: Continuous execution MULP: Pulse execution	32-bit instruction (13 steps) DMUL: Continuous execution DMULP: Pulse execution
S2	Multiplier	Data, or address of the word element that stores the data			
D	Product	Address of the word element that stores the data. If a 16-bit instruction is executed, the product contains 32 bits; if a 32-bit instruction is executed, the product contains 64 bits.			

4

◆ Operands

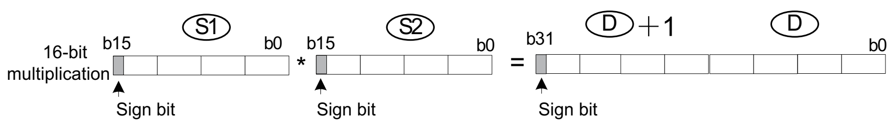
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

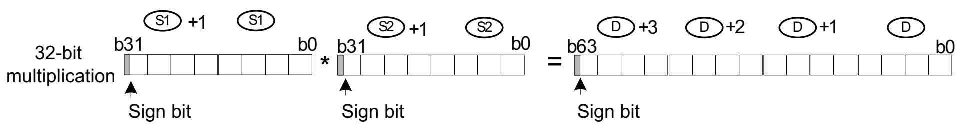
Note: The elements in gray background are supported.

◆ Function

The MUL instruction requires contact driving and has three operands. Use this instruction to algebraically multiply the values in S1 and S2 together in BIN mode. The result is stored in D. The variables in the algebraic operation are processed as signed numbers. The highest bit is the sign bit. The value 0 indicates a positive number, whereas the value 1 indicates a negative number. The product of a 32-bit multiplication occupies four registers.

In 32-bit operation, the variable addresses in the MUL instruction contain the lower 16 bits, and the adjacent high-numbered address unit contains the higher 16 bits. Avoid repeated or overlapping addresses during programming. The result of the calculation contains only 32 bits. If this limit is exceeded, use the floating point operation instruction EMUL.





Example:

Ladder chart

```
 M8
-| |-----(MUL   D100   D110   D120)
```

Instruction list

```
LD    M8
MUL   D100   D110   D120
```

When M8 is set, the product of D100 (multiplicand) and D110 (multiplier) is stored in D120.
D120 = 5 x 9 = K45 if D100 = K5 and D110 = K9.
D120, d121 = 1234 x 5678 = K7006652 if D100 = K1234 and D110 = K5678. The product contains more than 16 bits and occupies the adjacent higher bits D121 and D120 of D.

DIV: Binary number division

◆ Overview

The DIV instruction divides one binary number by another.

4

DIV S1 S2 D		Binary number division	Applicable model: H3U	
S1	Dividend	Data, or address of the word element that stores the data	16-bit instruction (7 steps) DIV: Continuous execution DIVP: Pulse execution	32-bit instruction (13 steps) DDIV: Continuous execution DDIVP: Pulse execution
S2	Divider	Data, or address of the word element that stores the data		
D	Quotient and remainder	Address of the word element that stores the data. The quotient is stored in the address D and the remainder stored in D+1.		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported. The V and Z elements in the preceding table are available only in 16-bit operation.

◆ Function

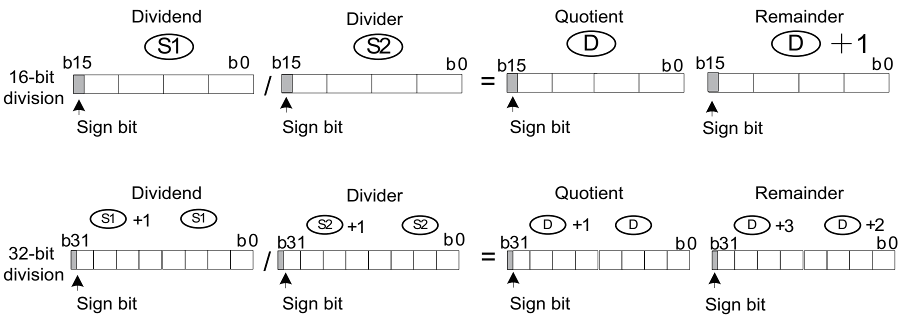
The DIV instruction requires contact driving and has three operands. Use this instruction to algebraically divide the value in S1 by that in S2 in BIN mode. The result is stored in D. The variables in the algebraic operation are processed as signed numbers. The highest bit is the sign bit. The value 0 indicates a positive number, whereas the value 1 indicates a negative number.

In 32-bit operation, the S1 and S2 variable addresses in the DIV instruction contain the lower 16 bits, and the adjacent high-numbered address unit contains the higher 16 bits. Avoid repeated or overlapping addresses during programming. The quotient is stored in the D and D+1 units, and the remainder is stored in the D+2 and D+3 address units.

A calculation error will occur if the divider S2 is 0.

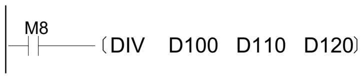
No remainder is produced if a bit element (KnX, KnY, KnM, or KnS) is designated as D.

If the dividend is a negative number, the remainder is also a negative number.



Example:

Ladder chart



Instruction list

```
LD   M8
DIV  D100  D110  D120
```

When M8 is set, D100 (dividend) is divided by D110 (divider). The quotient is stored in D120.
If D100 = K5 and D110 = K2, the remainder is stored in D121 (= K1).

4

EADD: Binary floating point addition

◆ Overview

The EADD instruction adds two binary floating-point numbers together.

EADD S1 S2 D			Binary floating point addition	Applicable model: H3U
S1	Augend	Augend of a binary floating point addition		32-bit instruction (13 steps) DEADD: Continuous execution DEADDP: Pulse execution
S2	Addend	Addend of a binary floating point addition		
D1	Sum	Unit that stores the sum of S1 and S2		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Two binary floating-point numbers are added together.

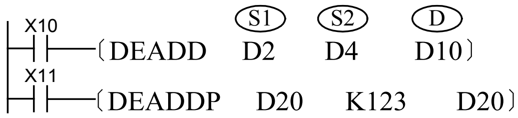
- S1 and S2 are respectively the augend and addend in a binary floating point addition.
- D is the unit that stores the sum of S1 and S2.
- If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the addition operation.

The zero flag M8020 is set if the result of the calculation is 0.

The carry flag M8022 is set if the absolute value of the calculation result is greater than the maximum floating-point value.

The borrow flag M8021 is set if the absolute value of the calculation result is less than the minimum floating-point value.

Example:



When X10 = ON, the sum of two binary floating-point numbers in (D3, D2) and (D5, D4) is stored in (D11, D10).

When X11 switches from OFF to ON, the floating-point number in (D21, D20) is incremented by 123. The constant K123 is changed to a binary floating-point number before the addition operation.

4

If the unit that stores the sum is the same as the augend or addend storage unit, use the DEADDP instruction of the pulse execution type. If the continuous execution type is used, calculation is performed upon every program scan.

ESUB: Binary floating point subtraction

◆ Overview

The ESUB instruction subtracts one binary floating-point number from another.

ESUB S1 S2 D		Binary floating point subtraction	Applicable model: H3U	
S1	Subtrahend	Subtrahend of a binary floating point subtraction		32-bit instruction (13 steps) DESUB: Continuous execution DESUBP: Pulse execution
S2	Minuend	Minuend of a binary floating point subtraction		
D	Difference	Unit that stores the difference in a binary floating point subtraction		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

One binary floating-point number is subtracted from another.

- S1 and S2 are respectively the subtrahend and minuend of a binary floating point subtraction.
- D is the unit that stores the difference in a binary floating point subtraction.
- If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the subtraction operation.

The zero flag M8020 is set if the result of the calculation is 0.

The carry flag M8022 is set if the absolute value of the calculation result is greater than the maximum floating-point value.

The borrow flag M8021 is set if the absolute value of the calculation result is less than the minimum floating-point value.

Example:

```
                          (S1)   (S2)    (D)
 X10
─┤├──[DESUB    D2     D4     D10]
 X11
─┤├──[DESUBP   D10    K123   D10]
```

When X10 = ON, the difference between two binary floating-point numbers in (D3, D2) and (D5, D4) is stored in (D11, D10).

When X11 switches from OFF to ON, the floating-point number in (D11, D10) is decremented by 123. The constant K123 is changed to a binary floating-point number before the subtraction operation.

If the unit that stores the difference is the same as the subtrahend or minuend storage unit, use the DESUBP instruction of the pulse execution type. If the continuous execution type is used, calculation is performed upon every program scan.

EMUL: Binary floating point multiplication

◆ Overview

The EMUL instruction multiplies two binary floating-point numbers together.

EMUL S1 S2 D			Binary floating point multiplication	Applicable model: H3U	
S1	Multiplicand	Multiplicand of a binary floating point multiplication			32-bit instruction (13 steps) DEMUL: Continuous execution DEMULP: Pulse execution
S2	Multiplier	Multiplier of a binary floating point multiplication			
D	Product	Unit that stores the product of a binary floating point multiplication			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

4

◆ Function

Two binary floating-point numbers are multiplied together.

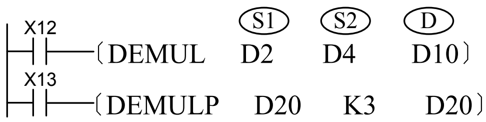
- S1 and S2 are respectively the multiplicand and multiplier of a binary floating point multiplication.
- D is the unit that stores the product of a binary floating point multiplication.
- If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the multiplication operation.

The zero flag M8020 is set if the result of the calculation is 0.

The carry flag M8022 is set if the absolute value of the calculation result is greater than the maximum floating-point value.

The borrow flag M8021 is set if the absolute value of the calculation result is less than the minimum floating-point value.

Example:



When X12 = ON, the product of multiplying the binary floating-point number in (D3, D2) by that in (D5, D4) is stored in (D11, D10).

When X13 switches from OFF to ON, the binary floating-point number in (D21, D20) is multiplied by 3 and the result is stored in (D21, D20) The constant K3 is changed to a binary floating-point number before the multiplication operation.

If the unit that stores the product is the same as the multiplicand or multiplier storage unit, use the DEMULP instruction of the pulse execution type. If the continuous execution type is used, calculation is performed upon every program scan.

EDIV: Binary floating point division

◆ Overview

The EDIV instruction divides one binary floating-point number by another.

<table>
<tr><th colspan="3">EDIV S1 S2 D</th><th>Binary floating point division</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S1</td><td>Dividend</td><td colspan="2">Dividend of a binary floating point division</td><td rowspan="3"></td><td rowspan="3">32-bit instruction (13 steps)
DEDIV: Continuous execution
DEDIVP: Pulse execution</td></tr>
<tr><td>S2</td><td>Divider</td><td colspan="2">Divider of a binary floating point division</td></tr>
<tr><td>D</td><td>Quotient</td><td colspan="2">Head address of units that store the quotient of a binary floating point division</td></tr>
</table>

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

One binary floating-point number is divided by another.

- S1 and S2 are respectively the dividend and divider of a binary floating point division.
- D is the head address for storing the quotient of a binary floating point division.
- If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the division operation.

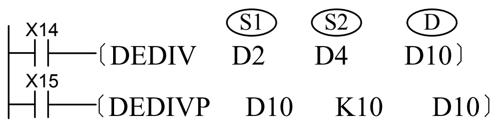
The zero flag M8020 is set if the result of the calculation is 0.

The carry flag M8022 is set if the absolute value of the calculation result is greater than the maximum floating-point value.

The borrow flag M8021 is set if the absolute value of the calculation result is less than the minimum floating-point value.

The divider cannot be 0; otherwise, a calculation error will occur, and M8067 and M8068 are set to ON to identify this error.

Example:



When X14 = ON, the quotient of dividing the binary floating-point number in (D3, D2) by that in (D5, D4) is stored in (D11, D10).

When X15 switches from OFF to ON, the binary floating-point number in (D11, D10) is divided by 10 and the result is stored in (D11, D10). The constant K10 is changed to a binary floating-point number before the division operation.

If the unit that stores the quotient is the same as the dividend or divider storage unit, use the DEDIVP instruction of the pulse execution type. If the continuous execution type is used, calculation is performed upon every program scan.

INC: Binary number incremented by 1

◆ Overview

The INC instruction increments a binary number by 1.

INC D			Binary number incremented by 1	Applicable model: H3U
D	Cumulative result	Address of the word element that stores the cumulative result	16-bit instruction (3 steps) INC: Continuous execution INCP: Pulse execution	32-bit instruction (5 steps) DINC: Continuous execution DINCP: Pulse execution

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

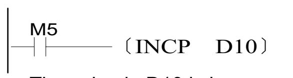
◆ Function

On every execution of the INC instruction, D has its current value incremented by a value of 1.

In 16-bit operation, –32,768 is reached after 32,767 is incremented by a value of 1. In 32-bit operation, –2,147,483,648 is reached after 2,147,483,647 is incremented by a value of 1.

This instruction does not refresh the zero flag, carry flag, and borrow flag.

Example:



The value in D10 is incremented by 1 each time M5 is set.

DEC: Binary number decremented by 1

◆ Overview

The DEC instruction decrements a binary number by 1.

DEC D		Binary number decremented by 1	Applicable model: H3U	
D	ive result	Address of the word element that stores the regressive result	16-bit instruction (3 steps) DEC: Continuous execution DECP: Pulse execution	32-bit instruction (5 steps) DDEC: Continuous execution DDECP: Pulse execution

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

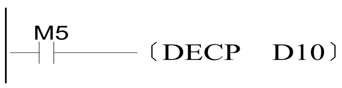
On every execution of the DEC instruction, D has its current value decremented by a value of 1.

In 16-bit operation, 32,767 is reached after –32,768 is decremented by a value of 1. In 32-bit operation, 2,147,483,647 is reached after –2,147,483,648 is decremented by a value of 1.

This instruction does not refresh the zero flag, carry flag, and borrow flag.

In 32-bit operation, the D variable address in this instruction contains the lower 16 bits, and the adjacent high-numbered address unit contains the higher 16 bits. Avoid repeated or overlapping addresses during programming.

Example:



The value in D10 is decremented by 1 each time M5 is set.

4

4.4.2 Logical Operations

Logical operations	WAND	Binary number logical AND
	WOR	Binary number logical OR
	WXOR	Binary number logical XOR
	NEG	Binary number negation
	ENEG	Binary floating-point sign negation

WAND: Binary number logical AND

◆ Overview

When driving conditions are met, the WAND instruction performs a logical AND on S1 and S2 bit by bit. The result is stored in D.

WAND S1 S2 D		Binary number logical AND	Applicable model: H3U	
S1	Data 1	Data in an AND operation, or address of the word element that stores the data	16-bit instruction (7 steps) WAND: Continuous execution WANDP: Pulse execution	32-bit instruction (13 steps) DAND: Continuous execution DANDP: Pulse execution
S2	Data 2	Data in an AND operation, or address of the word element that stores the data		
D	Operation result	Address of the word element that stores the operation result		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

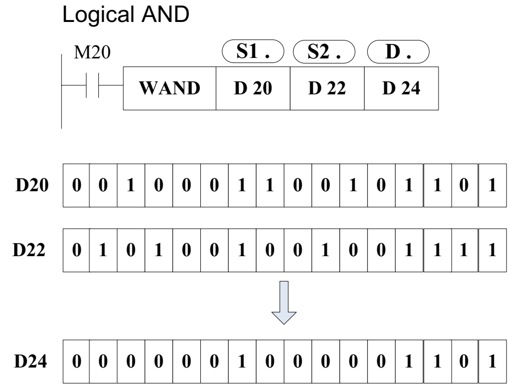
◆ Function

The bit patterns of the BIN values in S1 and S2 are analyzed. The result of the logical AND analysis is stored in the variable D.

The result of a logical AND operation is 0 if the value of either S1 or S2 is 0.

1•1 = 1 1•0 = 0 0•1 = 0 0•0 = 0

◆ Application



WOR: Binary number logical OR

4

◆ Overview

When driving conditions are met, the WOR instruction performs a logical OR on S1 and S2 bit by bit. The result is stored in D.

WOR S1 S2 D			Binary number logical OR	Applicable model: H3U	
S1	Data 1	Data in an OR operation, or address of the word element that stores the data		16-bit instruction (7 steps) WOR: Continuous execution WORP: Pulse execution	32-bit instruction (13 steps) DOR: Continuous execution DORP: Pulse execution
S2	Data 2	Data in an OR operation, or address of the word element that stores the data			
D	Operation result	Address of the word element that stores the operation result			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

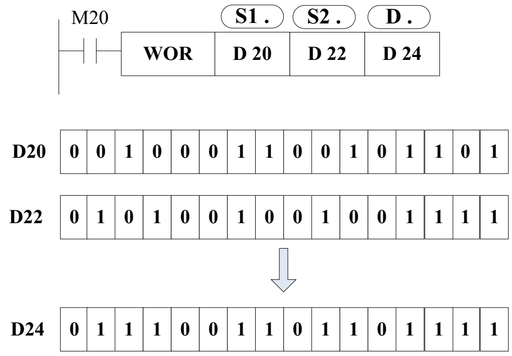
The bit patterns of the BIN values in S1 and S2 are analyzed. The result of the logical OR analysis is stored in the variable D.

The result of a logical OR operation is 0 if the value of either S1 or S2 is 0.

1 + 1 = 1   1 + 0 = 1   0 + 1 = 1   0 + 0 = 0

◆ Application

Logical OR



4

WXOR: Binary number logical XOR

◆ Overview

When driving conditions are met, the WXOR instruction performs a logical XOR on S1 and S2 bit by bit. The result is stored in D.

WXOR S1 S2 D		Binary number logical XOR	Applicable model: H3U	
S1	Data 1	Data in an XOR operation, or address of the word element that stores the data	16-bit instruction (7 steps) WXOR: Continuous execution WXORP: Pulse execution	32-bit instruction (13 steps) DXOR: Continuous execution DXORP: Pulse execution
S2	Data 2	Data in an XOR operation, or address of the word element that stores the data		
D	Operation result	Address of the word element that stores the operation result		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

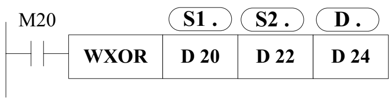
The bit patterns of the BIN values in S1 and S2 are analyzed. The result of the logical XOR analysis is stored in the variable D.

The result is 0 if S1 and S2 are the same; if they are different, the result is 1.

$1 \oplus 1 = 0 \quad 1 \oplus 0 = 1 \quad 0 \oplus 1 = 1 \quad 0 \oplus 0 = 0$

◆ Application

Logical XOR



D20	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0	1
D22	0	1	0	1	0	0	1	0	0	1	0	0	1	1	1	1
D24	0	1	1	1	0	0	0	1	0	1	1	0	0	0	1	0

NEG: Binary number negation

4

◆ Overview

When driving conditions are met, the NEG instruction inverts the bit pattern of D, adds 1 to the bit pattern, and then writes the result to D.

NEG D			Binary number negation	Applicable model: H3U	
D	Operation result	Address of the word element that stores the operation result		16-bit instruction (3 steps) NEC: Continuous execution NECP: Pulse execution	32-bit instruction (5 steps) DNEC: Continuous execution DNECP: Pulse execution

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

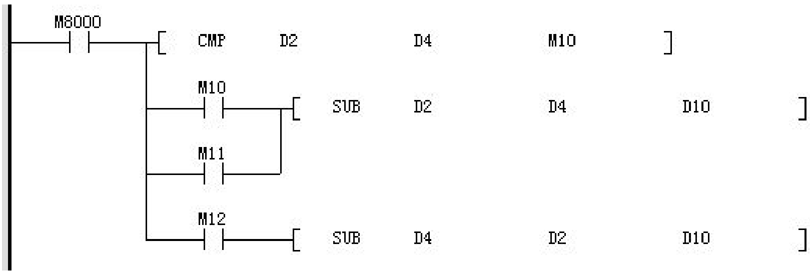
The NEG instruction requires contact driving and has one operand. It inverts the bit pattern of D, adds 1 to the bit pattern, and then writes the result to D.

The pulse execution type is generally used.

This gets the absolute value of a negative BIN number.

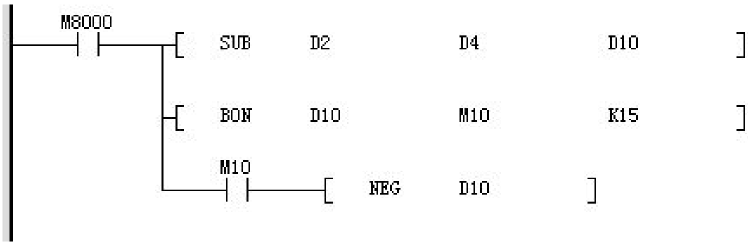
◆ Example:

The following example gets the absolute value of the difference in a subtraction:



M10 = ON if D2 > D4; M11 = ON if D2 = D4; M12 = ON if D2 < D4. This ensures that the value in D10 is positive.

The preceding program is represented as follows:



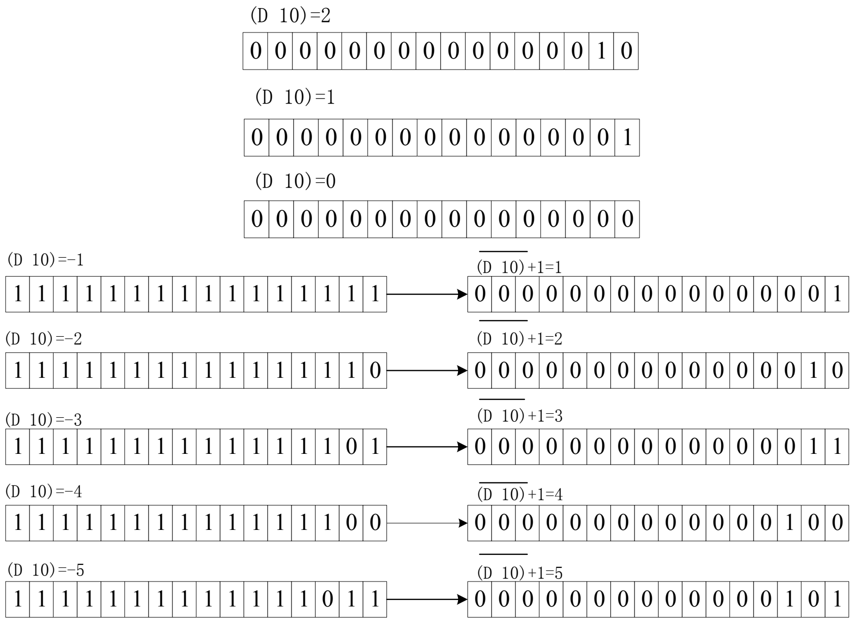
When bit 15 of D10 is 1 (indicating that the value in D10 is negative), then M10 = ON. Use the NEG instruction to get the absolute value of D10.

In the preceding examples, D10 = K4 if D2 = K4 and D4 = K8 or D2 = K8 and D4 = K4.

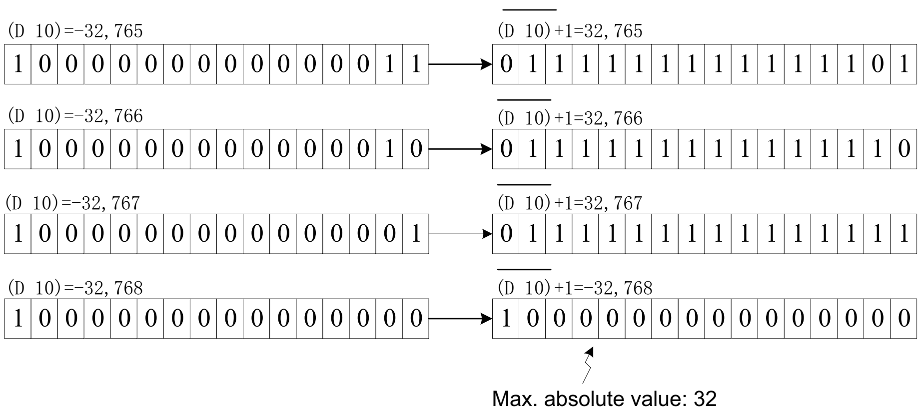
Negative number representation and absolute value:

1) Positive and negative numbers are represented by the leftmost bit content of the register. A positive number is represented by 0, whereas a negative number is represented by 1.

2) When the highest bit is 1, use the NEG instruction to convert it to an absolute value.



4



ENEG: Binary floating-point sign negation

◆ Overview

The ENEG instruction inverts the sign of a binary floating-point number (real number).

ENEG S/D			Binary floating-point sign negation	Applicable model: H3U	
S/D	Operand	Start number of elements that store the binary floating-point number subjected to a sign change			32-bit instruction (5 steps) DENEG: Continuous execution DENEGP: Pulse execution

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S/D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

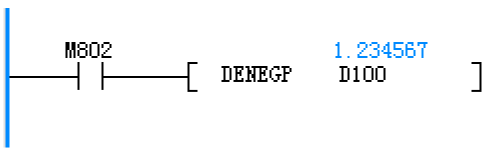
◆ Function

The sign of the binary floating-point number in [D+1, D] is inverted. The result is stored in [D+1, D]. The pulse execution type is generally used.

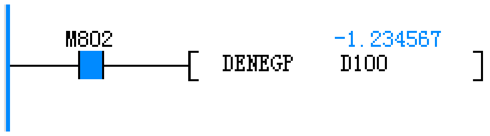
◆ Application

The following example inverts the data in D100 and D101. The result is stored in D100 and D101.

Before execution of the instruction



After execution of the instruction



4.4.3 Trigonometric Functions

Trigonometric functions	SIN	Floating point SIN operation
	COS	Floating point COS operation
	TAN	Floating point TAN operation
	ASIN	Binary floating point ARCSIN operation
	ACOS	Binary floating point ARCCOS operation
	ATAN	Binary floating point ARCTAN operation
	RAD	Binary floating point degree-to-radian conversion
	DEG	Binary floating point radian-to-degree conversion
	SINH	Binary floating point SINH operation
	COSH	Binary floating point COSH operation
	TANH	Binary floating point TANH operation

SIN: Floating point SIN operation

◆ Overview

The SIN instruction calculates the sine of the designated angle (measured in radians). The variable is in binary floating-point number format.

SIN S D			Floating point SIN operation	Applicable model: H3U
S	Data source	Angular variable whose sine is to be calculated, in the unit of rad and in binary floating-point number format; value range: 0 <= α <= 2π		32-bit instruction (9 steps) DSIN: Continuous execution DSINP: Pulse execution
D	Operation result	Sine storage unit, in binary floating-point number format		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

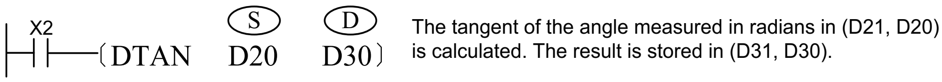
◆ Function

The sine of the designated angle (measured in radians) is calculated. The variable is in binary floating-point number format.

S is the angular variable whose sine is to be calculated, in the unit of rad and in binary floating-point number format. The value range is 0 ≤ α ≤ 2π.

D is the sine storage unit, in binary floating-point number format.

Example 1:

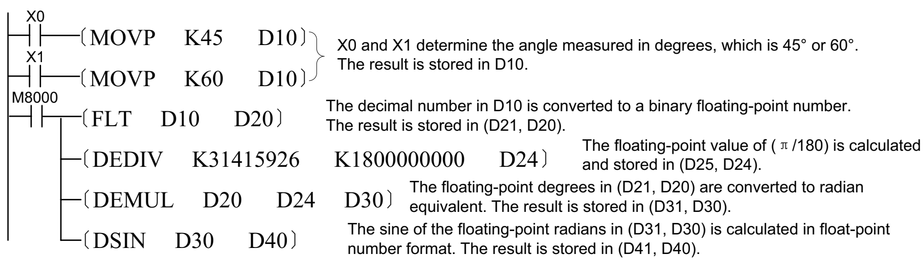


The source data and SIN operation result are in binary floating-point number format.

According to the equation Angle in radians = Angle in degrees x π/180°, an angle of 360° is converted to radians as follows: 360° x π/180° = 2π.

Example 2:

Program for calculating the sine of an angle measured in degrees:



COS: Binary floating point COS operation

◆ Overview

The COS instruction calculates the cosine of the designated angle (measured in radians). The variable is in binary floating-point number format.

COS S D		Binary floating point COS operation	Applicable model: H3U	
S	Data source	Angular variable whose cosine is to be calculated, in the unit of rad and in binary floating-point number format; value range: 0 ≤ α ≤ 2π		32-bit instruction (9 steps) DCOS: Continuous execution DCOSP: Pulse execution
D	Operation result	Cosine storage unit, in binary floating-point number format		

4

◆ **Operands**

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ **Function**

The cosine of the designated angle (measured in radians) is calculated. The variable is in binary floating-point number format.

4

Example:

The tangent of the angle measured in radians in (D21, D20) is calculated. The result is stored in (D31, D30).

The source data and COS operation result are in binary floating-point number format.

According to the equation Angle in radians = Angle in degrees x π/180°, an angle of 360° is converted to radians as follows: 360° x π/180° = 2π.

For details about the programming statement for calculating the cosine in degrees, see the example described in "SIN: Floating point SIN operation" on Page 120.

TAN: Floating point TAN operation

◆ **Overview**

The TAN instruction calculates the tangent of the designated angle (measured in radians). The variable is in binary floating-point number format.

TAN S D			**Floating point TAN operation**	**Applicable model: H3U**
S	Data source	Angular variable whose tangent is to be calculated, in the unit of rad and in binary floating-point number format; value range: 0 ≤ α < 2π		32-bit instruction (9 steps) DTAN: Continuous execution DTANP: Pulse execution
D	Operation result	Tangent storage unit, in binary floating-point number format		

◆ **Operands**

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The tangent of the designated angle (measured in radians) is calculated. The variable is in binary floating-point number format.

Example:

```
   X2           (S)    (D)
|--| |------(DTAN  D20   D30)
```

The tangent of the angle measured in radians in (D21, D20) is calculated. The result is stored in (D31, D30).

The source data and TAN operation result are in binary floating-point number format.

According to the equation Angle in radians = Angle in degrees x π/180°, an angle of 360° is converted to radians as follows: 360° x π/180° = 2π.

For details about the programming statement for calculating the tangent in degrees, see the example described in "SIN: Floating point SIN operation" on Page 120.

ASIN: Binary floating point ARCSIN operation

◆ Overview

The ASIN instruction calculates the sine in radians.

ASIN S D		Binary floating point ARCSIN operation	Applicable model: H3U	
S	Data source	Binary floating-point variable whose arcsine is to be calculated		32-bit instruction (9 steps) DASIN: Continuous execution DASINP: Pulse execution
D	Operation result	Operation result storage unit (in the range –π/2 to +π/2)		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

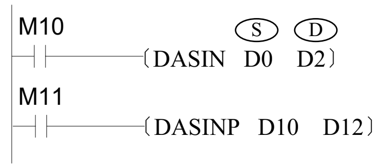
Note: The elements in gray background are supported.

◆ Function

The sine in radians is calculated.

Note: An operation error will occur if the value in S exceeds the range –1.0 to +1.0. The error code is K6706 and stored in D8067. The error flag M8067 is set to ON to identify this error.

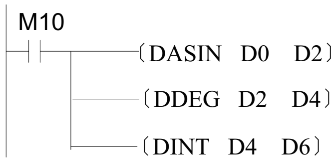
Example 1:



When M10 = ON, the sine-1 of the binary floating-point number in (D1, D0) is calculated. The result is stored in (D3, D2).

$$SIN^{-1}(D1、D0) \Longrightarrow (D3、D2)$$

Example 2:



Assume that the value in (D1, D0) is 0.707106781. When M10 switches from OFF to ON, the value in (D3, D2) is 0.78539815, that in (D5, D4) is 45, and that in (D7, D6) is 45.

ACOS: Binary floating point ARCCOS operation

◆ Overview

The ACOS instruction calculates the cosine in radians.

ACOS S D			Binary floating point ARCCOS operation	Applicable model: H3U	
S	Data source	Binary floating-point variable whose arccosine is to be calculated			32-bit instruction (9 steps) DACOS: Continuous execution DACOSP: Pulse execution
D	Operation result	Operation result storage unit (0 to π)			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

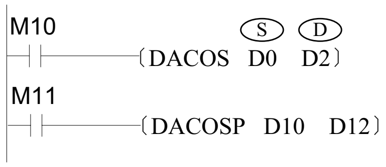
Note: The elements in gray background are supported.

◆ Function

The cosine in radians is calculated.

Note: An operation error will occur if the value in S exceeds the range –1.0 to +1.0. The error code is K6706 and stored in D8067. The error flag M8067 is set to ON to identify this error.

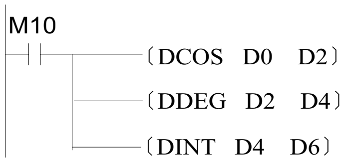
Example 1:



When M10 = ON, the cosine-1 of the binary floating-point number in (D1, D0) is calculated. The result is stored in (D3, D2).

$$\cos^{-1}(D1,\ D0) \Rightarrow (D3,\ D2)$$

Example 2:



Assume that the value in (D1, D0) is 0.866025404. When M10 switches from OFF to ON, the value in (D3, D2) is 0.52359877, that in (D5, D4) is 30, and that in (D7, D6) is 30.

4

ATAN: Binary floating point ARCTAN operation

◆ Overview

The ATAN instruction calculates the tangent in radians.

<table>
<tr><td colspan="3">ATAN S D</td><td>Binary floating point ARCTAN operation</td><td colspan="2">Applicable model: H3U</td></tr>
<tr><td>S</td><td>Data source</td><td colspan="2">Binary floating-point variable whose arctangent is to be calculated</td><td rowspan="2"></td><td rowspan="2">32-bit instruction (9 steps)
DATAN: Continuous execution
DATANP: Pulse execution</td></tr>
<tr><td>D</td><td>Operation result</td><td colspan="2">Operation result storage unit (in the range –π/2 to +π/2)</td></tr>
</table>

◆ Operands

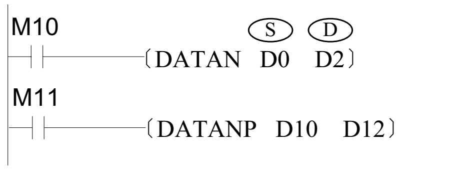
<table>
<tr><td rowspan="2">Operand</td><td colspan="7">Bit Element</td><td colspan="15">Word Element</td></tr>
<tr><td colspan="7">System·User</td><td colspan="5">System·User</td><td colspan="5">Bit Designation</td><td colspan="2">Indexed Address</td><td colspan="2">Constant</td><td>Real Number</td></tr>
<tr><td>S</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modification</td><td>K</td><td>H</td><td>E</td></tr>
<tr><td>D</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modification</td><td>K</td><td>H</td><td>E</td></tr>
</table>

Note: The elements in gray background are supported.

◆ Function

The tangent in radians is calculated.

Example 1:

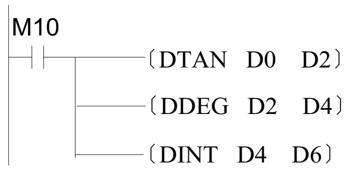


When M10 = ON, the tangent-1 of the binary floating-point number in (D1, D0) is calculated. The result is stored in (D3, D2).

$$TAN^{-1}(D1、D0) \Longrightarrow (D3、D2)$$

4

Example 2:



Assume that the value in (D1, D0) is 1.732050808.
When M10 switches from OFF to ON, the value in (D3, D2) is 1.04719753, that in (D5, D4) is 60, and that in (D7, D6) is 60.

RAD: Binary floating point degree-to-radian conversion

◆ Overview

The RAD instruction converts binary floating-point degrees to radians. The calculation formula is [Angle in radians = Angle in degrees x π/180].

RAD S D			Binary floating point degree-to-radian conversion	Applicable model: H3U	
S	Data source	Binary floating-point degrees to be converted to radians			32-bit instruction (9 steps) DRAD: Continuous execution DRADP: Pulse execution
D	Operation result	Operation result storage unit			

◆ Operands

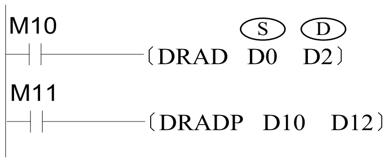
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Binary floating-point degrees are converted to radians. The calculation formula is [Angle in radians = Angle in degrees x π/180].

Example 1:



When M10 = ON, degree-to-radian conversion is performed on the binary floating-point number in (D1, D0). The result is stored in (D3, D2).

Example 2:

When M10 switches from OFF to ON, the value 90 is assigned to D0. The integer in D0 is converted to a floating-point number, which is then assigned to (D3, D2). Degree-to-radian conversion is performed on (D3, D2) and the result is assigned to (D5, D4). The final value in (D3, D2) is π/2, that is, 1.570796.

DEG: Binary floating point radian-to-degree conversion

◆ Overview

The DEG instruction converts binary floating-point radians to degrees. The calculation formula is [Angle in degrees = Angle in radians x π/180].

DEG S D			Binary floating point radian-to-degree conversion	Applicable model: H3U	
S	Data source	Binary floating-point radian variable to be converted to degrees			32-bit instruction (9 steps) DDEG: Continuous execution DDEGP: Pulse execution
D	Operation result	Operation result storage unit			

4

◆ Operands

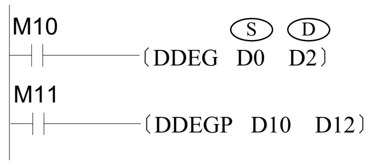
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

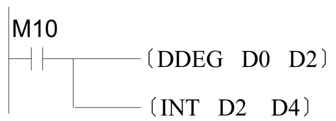
Binary floating-point radians are converted to degrees. The calculation formula is [Angle in degrees = Angle in radians x 180/π].

Example 1:



When M10 = ON, radian-to-degree conversion is performed on the binary floating-point number in (D1, D0). The result is stored in (D3, D2).

Example 2:



Assume that the value in (D1, D0) is 3.1415926. When M10 switches from OFF to ON, the value in (D3, D2) is 180. After the floating-point number is converted to an integer, the value in (D5, D4) is 180.

SINH: Binary floating point SINH operation

◆ Overview

The SINH instruction calculates the sinh of a binary floating-point number. The calculation formula is Sinh = $(e^s - e^{-s})/2$.

SINH S1 D			Binary floating point SINH operation	Applicable model: H3U	
S1	Data source	Binary floating-point variable whose sinh is to be calculated			32-bit instruction (9 steps) DSINH: Continuous execution DSINH P: Pulse execution
D	Operation result	Operation result storage unit (Error 6706 is returned if the operation result D exceeds the floating-point range.)			

◆ Operands

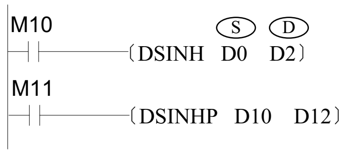
Operand	Bit Element							Word Element														
	System·User							System·User				Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The sinh of a binary floating-point number is calculated. The calculation formula is Sinh = $(e^s - e^{-s})/2$.

Example:



When M10 = ON, the sinh of the binary floating-point number in (D1, D0) is calculated. The result is stored in (D3, D2).

COSH: Binary floating point COSH operation

◆ Overview

The COSH instruction calculates the cosh of a binary floating-point number. The calculation formula is Cosh = $(e^{s} – e^{-s})/2$.

COSH S D		Binary floating point COSH operation	Applicable model: H3U	
S	Data source	Binary floating-point variable whose cosh is to be calculated		32-bit instruction (9 steps) DCOSH: Continuous execution DCOSHP: Pulse execution
D	Operation result	Operation result storage unit		

◆ Operands

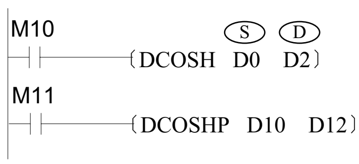
Operand	Bit Element							Word Element														
	System·User							System·User				Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The cosh of a binary floating-point number is calculated. The calculation formula is Cosh = $(e^{s} + e^{-s})/2$.

Example:



When M10 = ON, the cosh of the binary floating-point number in (D1, D0) is calculated. The result is stored in (D3, D2).

TANH: Binary floating point TANH operation

◆ Overview

The TANH instruction calculates the tanh of a binary floating-point number. The calculation formula is Tanh = $(e^{s} – e^{-s})$ $(/e^{s} + e^{-s})$.

TANH S D		Floating point TANH operation	Applicable model: H3U	
S	Data source	Binary floating-point variable whose tanh is to be calculated		32-bit instruction (9 steps) DTANH: Continuous execution DTANHP: Pulse execution
D	Operation result	Operation result storage unit		

◆ Operands

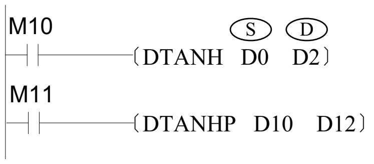
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The tanh of a binary floating-point number is calculated. The calculation formula is Tanh = $(e^{s} – e^{-s})/(e^{s} + e^{-s})$.

Example:



When M10 = ON, the tanh of the binary floating-point number in (D1, D0) is calculated. The result is stored in (D3, D2).

4.4.4 Table Operations

Table operations	WSUM	Value summation
	MEAN	Mean value calculation
	LIMIT	Upper/Lower limit control
	BZAND	Dead zone control
	ZONE	Zone control
	SCL	Coordinate determination (coordinates of different points)
	SCL2	Coordinate determination 2 (X and Y coordinates)

WSUM: Value summation

◆ Overview

The WSUM instruction calculates the sum of consecutive 16- or 32-bit data entries.

WSUM S D n			Value summation	Applicable model: H3U	
S	Source data	Start number of elements that store the data entries whose sum is to be calculated		16-bit instruction (7 steps) WSUM: Continuous execution WSUMP: Pulse execution	32-bit instruction (13 steps) DWSUM: Continuous execution DWSUMP: Pulse execution
D	Result	Start number of elements that store the sum			
n	Data count	Number of operated data entries (n > 0)			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

The sum of *n* 16-bit data entries from head address [S] is calculated. The result is stored as 32-bit data in [D+1, D].

2) 32-bit instruction

The sum of n 32-bit data entries from head address [S+1, S] is calculated. The result is stored as 64-bit data in [D+3, D+2, D+1, D].

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6705 is returned if *n* elements from head address [S] are out of range.
- Error 6705 is returned if [D] for data storage is out of range.
- Error 6706 is returned if the operand n ≤ 0.

◆ Application

The following example gets the sum of six D elements from head address D100 and stores the result in [D121, D120]. D100 to D105 each are assigned a value of 10, as shown in the following figure.

MEAN: Mean value calculation

◆ Overview

When driving conditions are met, the MEAN instruction calculates the mean value of K data entries from head address S. The result is stored in D.

MEAN S D n			Mean value calculation	Applicable model: H3U
S	Data head address	Head address of word elements that store the data entries whose mean value is to be calculated	16-bit instruction (7 steps) MEAN: Continuous execution MEANP: Pulse execution	32-bit instruction (13 steps) DMEAN: Continuous execution DMEANP: Pulse execution
D	Mean value	Address of the word element that stores the mean value		
n	Data length	Immediate value: K = 1–64		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The mean value of N variables from head address S is calculated by dividing the sum of the variables by n. The result is stored in D.

The remainder (if any) of the calculation result is discarded.

A calculation error will occur when n exceeds the range 1 to 64.

Example:

(D10 + D11 + D12 + D13)/4 = D20
Assume that D10 = K5, D11 = K5, D12 = K15, and D13 = K52.
Then D20 = K19, and the remainder 1 is discarded.

M8025: Cumulative summation flag

When M8025 = OFF, the result is as shown in the preceding figure.

When M8025 = ON, the cumulative sum of the preceding variables is stored in D+1 and D+2.

As shown in the preceding example, the sum of D10, D11, D12, and D13 is stored in (D22, D21), which occupies 32 bits.

LIMIT: Upper/Lower limit control

◆ Overview

The LIMIT instruction sets the upper and lower limits of an input value and outputs the resulting value.

LIMIT S1 S2 S3 D			Upper/Lower limit control	Applicable model: H3U	
S1	Lower limit	Minimum output limit		16-bit instruction (9 steps) LIMIT: Continuous execution LIMITP: Pulse execution	32-bit instruction (17 steps) DLIMIT: Continuous execution DLIMITP: Pulse execution
S2	Upper limit	Maximum output limit			
S3	Input value	Input value to be controlled by lower and upper limits			
D	Output value	Start number of elements that store an output value under lower/upper limit control			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

The output value in [D] is generated based on the input value in [S3] and controlled within the range defined by the lower and upper limits respectively set in [S1] and [S2].

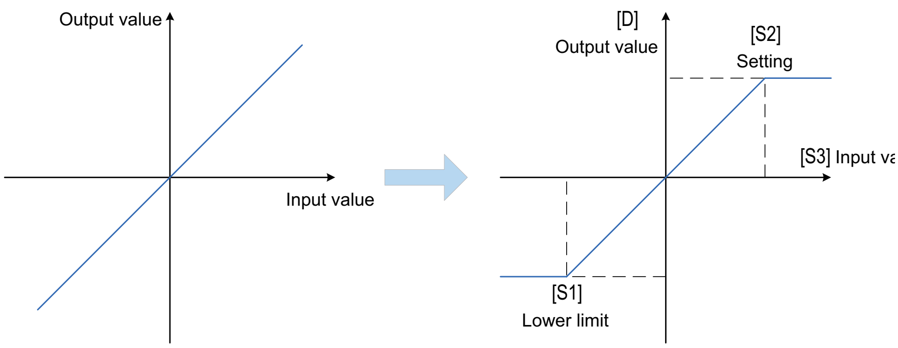
[S1] -> [D] when [S1] > [S3];

[S2] -> [D] when [S2] < [S3];

[S3] -> [D] when [S1] ≤ [S3] ≤ [S2].

If only the upper limit is applied, set the 16-bit minimum signed value –32,768 in the lower limit setting [S1].

If only the lower limit is applied, set the 16-bit maximum signed value 32,767 in the upper limit setting [S2].



2) 32-bit instruction

The output value in [D+1, D] is generated based on the input value in [S3+1, S3] and controlled within the range defined by the lower and upper limits respectively set in [S1+1, S1] and [S2+1, S2].

[S1+1, S1] -> [D+1, D] when [S1+1, S1] > [S3+1, S3];

[S2+1, S2] -> [D+1，D] when [S2+1, S2] < [S3+1, S3];

[S3+1, S3] -> [D+1, D] when [S1+1, S1] ≤ [S3+1, S3] ≤ [S2+1, S2].

If only the upper limit is applied, set the 32-bit minimum signed value –2,147,483,648 in the lower limit setting [S1+1, S1].

If only the lower limit is applied, set the 32-bit maximum signed value 2,147,483,647 in the upper limit setting [S2+1, S2].

4

An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

For the 16- and 32-bit instructions, error 6706 is returned when the lower limit is greater than the upper limit.

◆ Application

BZAND: Dead zone control

◆ Overview

The BZAND instruction controls an output value based on whether the input value is within the range (defined by upper and lower limits) of the designated dead zone.

BZAND S1 S2 S3 D		Dead zone control	Applicable model: H3U	
S1	Lower limit	Lower limit of a dead zone (with no output zone)	16-bit instruction (9 steps) BZAND: Continuous execution BZANDP: Pulse execution	32-bit instruction (17 steps) DBZAND: Continuous execution DBZANDP: Pulse execution
S2	Upper limit	Upper limit of a dead zone (with no output zone)		
S3	Input value	Input value subjected to dead zone control		
D	Output value	Number of the element that stores an output value under dead zone control		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

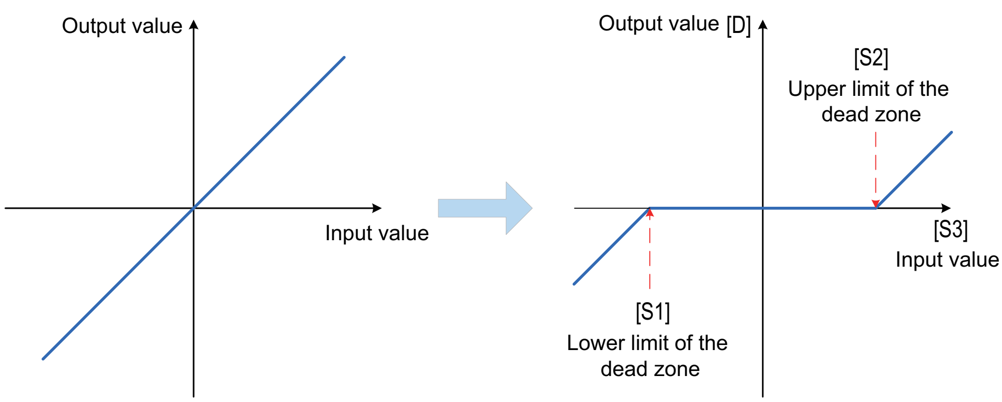
A dead zone range is set in [S1] and [S2] and applied to the input value in [S3]. The resulting value is output to the element [D].

The output value is controlled as follows:

[S3] – [S1] -> [D] when [S1] > [S3];

[S3] – [S2] -> [D] when [S2] < [S3];

0 -> [D] when [S1] ≤ [S3] ≤ [S2].



2) **32-bit instruction**

A dead zone range is set in [S1+1, S1] and [S2+1, S2] and applied to the input value in [S3+1, S3]. The resulting value is output to the element [D+1, D].

[S3+1, S3] – [S1+1, S1] -> [D+1, D] when [S1+1, S1] > [S3+1, S3];

[S3+1, S3] – [S2+1, S2] -> [D+1, D] when [S2+1, S2] < [S3+1, S3];

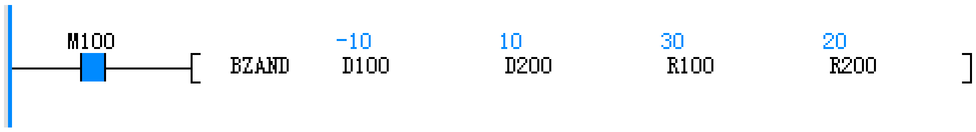
0 -> [D+1, D] when [S1+1, S1]≤ [S3+1, S3] ≤ [S2+1, S2].

Data overflow conforms to cyclical processing during instruction execution. That is, the minimum value is reached when the maximum value is incremented by 1; the maximum value is reached when the minimum value is decremented by 1.

An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

For the 16- and 32-bit instructions, error 6706 is returned when the lower limit is greater than the upper limit.

◆ Application



ZONE: Zone control

◆ Overview

The ZONE instruction controls an output value by using the designated deviation value based on whether the input value is positive or negative.

ZONE S1 S2 S3 D		Zone control	Applicable model: H3U	
S1	Negative deviation value	Negative deviation value (which may be a positive or negative number or 0) added to an input value	16-bit instruction (9 steps) ZONE: Continuous execution ZONEP: Pulse execution	32-bit instruction (17 steps) DZONE: Continuous execution DZONEP: Pulse execution
S2	Positive deviation value	Positive deviation value (which may be a positive or negative number or 0) added to an input value		
S3	Input value	Input value to be subjected to zone control		
D	Output value	Start number of elements that store an output value under zone control		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

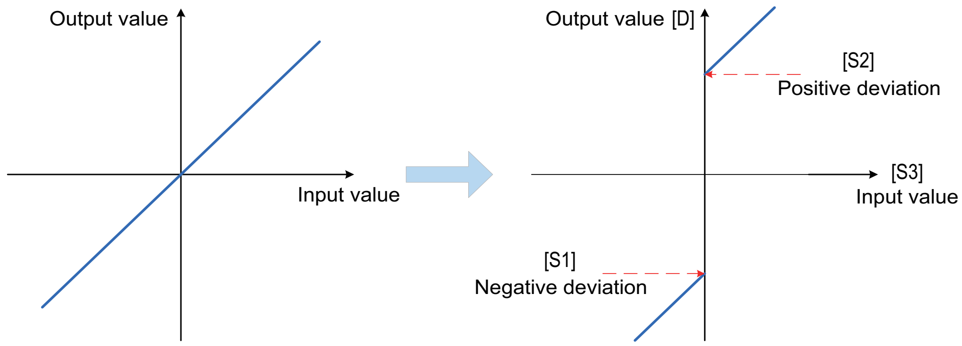
The value in [S2] or [S1] is added to the input value in [S3] based on whether it carries a plus or minus sign. The result is stored in the element [D].

[S3] + [S1] -> [D] when [S3] < 0;

[S3] + [S2] -> [D] when [S3] > 0;

0 -> [D] when [S3] = 0.

The instruction is executed as follows:



2) 32-bit instruction

The value in [S2+1, S2] or [S1+1, S1] is added to the input value in [S3+1, S3] based on whether it carries a plus or minus sign. The result is stored in the element [D+1, D].

◆ Application

4

SCL: Coordinate determination (coordinates of different points)

◆ Overview

The SCL instruction determines the coordinates of an input value based on the designated data table and outputs the resulting value.

SCL S1 S2 D		Coordinate determination (coordinates of different points)	Applicable model: H3U	
S1	Input value	Input value subjected to coordinate determination, or number of the element that stores the input value	16-bit instruction (7 steps)	32-bit instruction (13 steps)
S2	Table data	Start number of elements that store the conversion table used for coordinate determination	SCL: Continuous execution	DSCL: Continuous execution
D	Output value	Number of the element that stores the output value under coordinate control	SCLP: Pulse execution	DSCLP: Pulse execution

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

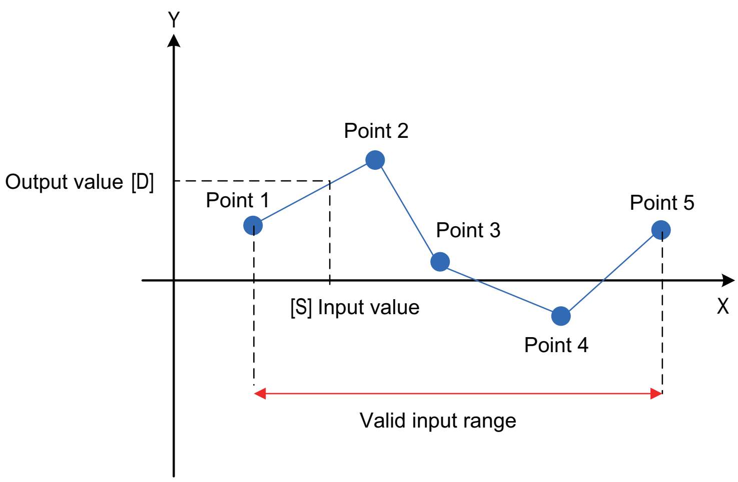
Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

The output value in [D] that corresponds to the input value in [S1] is identified based on the graph determined by the table in [S2]. If the output value is not an integer, the first digit of the fractional part is rounded.

The instruction is executed as follows:



The [S2] data organizational form in the 16-bit instruction is as follows:

Setting		Element Assignment for the Data Table
Assume that the number of coordinate points is 5.		[S2]
Point 1	X coordinate	[S2+1]
	Y coordinate	[S2+2]
Point 2	X coordinate	[S2+3]
	Y coordinate	[S2+4]
Point 3	X coordinate	[S2+5]
	Y coordinate	[S2+6]
Point 4	X coordinate	[S2+7]
	Y coordinate	[S2+8]
Point 5	X coordinate	[S2+9]
	Y coordinate	[S2+10]

2) 32-bit Instruction

The output value in [D+1, D] that corresponds to the input value in [S1+1, S1] is identified based on the graph determined by the table in [S2+1, S2]. If the output value is not an integer, the first digit of the fractional part is rounded.

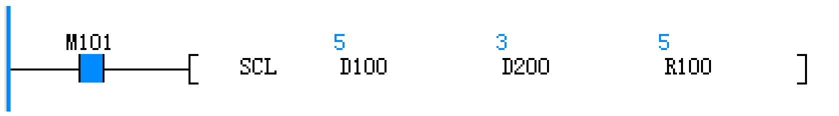
The [S2] data organizational form in the 32-bit instruction is as follows:

Setting		Element Assignment for the Data Table
Assume that the number of coordinate points is 5.		[S2+1, S]
Point 1	X coordinate	[S2+3, S2+2]
	Y coordinate	[S2+5, S2+4]
Point 2	X coordinate	[S2+7, S2+6]
	Y coordinate	[S2+9, S2+8]
Point 3	X coordinate	[S2+11, S2+10]
	Y coordinate	[S2+13, S2+12]
Point 4	X coordinate	[S2+15, S2+14]
	Y coordinate	[S2+17, S2+16]
Point 5	X coordinate	[S2+19, S2+18]
	Y coordinate	[S2+21, S2+20]

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6706 is returned when the x coordinates of table data are not sorted in ascending order.
- However, the SCL instruction is still executed properly for coordinate output within the x-coordinate ascending sorting range.
- Error 6706 is returned when the value in [S1] exceeds the range of the table data.

◆ Application



	Element Name	data type	display format	current value
1	D100	16-bit int	Dec	5
2	D200	16-bit int	Dec	3
3	D201	16-bit int	Dec	0
4	D202	16-bit int	Dec	0
5	D203	16-bit int	Dec	10
6	D204	16-bit int	Dec	10
7	D205	16-bit int	Dec	20
8	D206	16-bit int	Dec	0
9		16-bit int	Dec	
10	R100	16-bit int	Dec	5

4

SCL2: Coordinate determination 2 (X and Y coordinates)

◆ Overview

The SCL2 instruction determines the coordinates of an input value based on the designated data table and outputs the resulting value.

SCL2 S1 S2 D			Coordinate determination 2 (X and Y coordinates)	Applicable model: H3U
S1	Input value	Input value subjected to coordinate determination, or number of the element that stores the input value	16-bit instruction (7 steps) SCL2: Continuous execution SCL2P: Pulse execution	32-bit instruction (13 steps) DSCL2: Continuous execution DSCL2P: Pulse execution
S2	Table data	Start number of elements that store the conversion table used for coordinate determination		
D	Output value	Number of the element that stores the output value under coordinate control		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

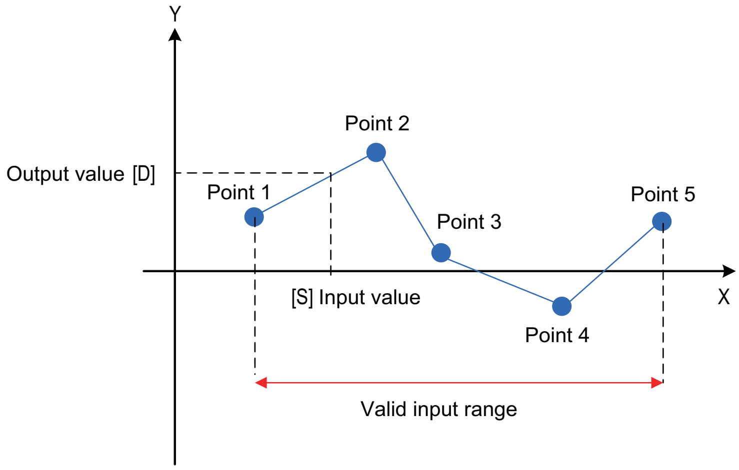
Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

The output value in [D] that corresponds to the input value in [S1] is identified based on the graph determined by the table in [S2]. If the output value is not an integer, the first digit of the fractional part is rounded.

The instruction is executed as follows:



The [S2] data organizational form in the 16-bit instruction is as follows:

Setting		Element Assignment for the Data Table
Assume that the number of coordinate points is 5.		[S2]
X coordinate	Point 1	[S2+1]
	Point 2	[S2+2]
	Point 3	[S2+3]
	Point 4	[S2+4]
	Point 5	[S2+5]
Y coordinate	Point 1	[S2+6]
	Point 2	[S2+7]
	Point 3	[S2+8]
	Point 4	[S2+9]
	Point 5	[S2+10]

2) 32-bit instruction

The output value in [D+1, D] that corresponds to the input value in [S1+1, S1] is identified based on the graph determined by the table in [S2+1, S2]. If the output value is not an integer, the first digit of the fractional part is rounded.

The [S2+1, S2] data organizational form in the 32-bit instruction is as follows:

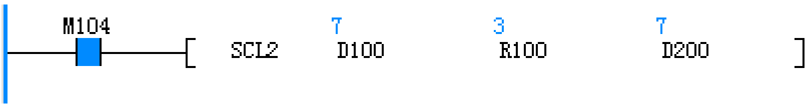
Setting		Element Assignment for the Data Table
Assume that the number of coordinate points is 5.		[S2, S2+1]
X coordinate	Point 1	[S2+3, S2+2]
	Point 2	[S2+5, S2+4]
	Point 3	[S2+7, S2+6]
	Point 4	[S2+9, S2+8]
	Point 5	[S2+11, S2+10]
Y coordinate	Point 1	[S2+13, S2+12]
	Point 2	[S2+15, S2+14]
	Point 3	[S2+17, S2+16]
	Point 4	[S2+19, S2+18]
	Point 5	[S2+21, S2+20]

4

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6706 is returned when the x coordinates of table data are not sorted in ascending order.
- However, the SCL2 instruction is still executed properly for coordinate output within the x-coordinate ascending sorting range.
- Error 6706 is returned when the value in [S1] exceeds the range of the table data.

◆ Application



	Element Name	data type	display format	current value
1	D100	16-bit int	Dec	7
2		16-bit int	Dec	
3	R100	16-bit int	Dec	3
4	R101	16-bit int	Dec	0
5	R102	16-bit int	Dec	10
6	R103	16-bit int	Dec	20
7	R104	16-bit int	Dec	0
8	R105	16-bit int	Dec	10
9	R106	16-bit int	Dec	0
10		16-bit int	Dec	
11	D200	16-bit int	Dec	7

4.4.5 Exponent Operations

Exponent operations	EXP	Binary floating-point exponent operation
	LOGE	Binary floating-point natural logarithm operation
	LOG	Binary floating-point logarithm operation with a base of 10
	ESQR	Binary floating-point square root operation
	SQR	Binary number square root operation
	POW	Floating-point weight instruction

EXP: Binary floating-point exponent operation

◆ Overview

The EXP instruction performs exponentiation of the mathematical constant *e* (approximately equal to 2.71828) with the exponent being a binary floating-point number.

EXP S D			Binary floating-point exponent operation	Applicable model: H3U
S	Data source	Binary floating-point variable used as the exponent		32-bit instruction (9 steps) DEXP: Continuous execution DEXPP: Pulse execution
D	Operation result	Unit that stores the result of exponentiation		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

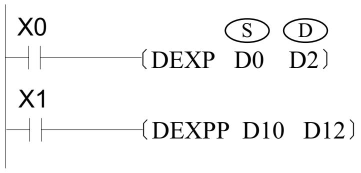
Exponentiation is performed on the mathematical constant e (approximately equal to 2.71828). The exponent is a binary floating-point number.

S is the binary floating-point variable used as the exponent.

D is the unit that stores the result of exponentiation.

Note: An error occurs when the result exceeds the range $[2^{-126}, 2^{128})$. The error code is K6706 and stored in D8067. The error flag M8067 is set to ON to identify this error.

Example:



When X0 = ON, exponentiation is performed on the mathematical constant e. The exponent is the binary floating-point number in (D1, D0). The result is stored in (D3, D2). e(D1, D0) -> (D3, D2). Because loge2128 = 88.7, when the value in (D1, D0) is greater than 88.7, then D8067 = K6706 and M8067 = ON.

LOGE: Binary floating-point natural logarithm operation

◆ Overview

The LOGE instruction calculates the natural logarithm of a binary floating-point number with the mathematical constant e (approximately equal to 2.71828) as the base.

LOGE S D			Binary floating-point natural logarithm operation	Applicable model: H3U	
S	Data source	Binary floating-point variable whose natural logarithm is to be calculated			32-bit instruction (9 steps) DLOGE: Continuous execution DLOGEP: Pulse execution
D	Operation result	Unit that stores the calculated natural logarithm			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

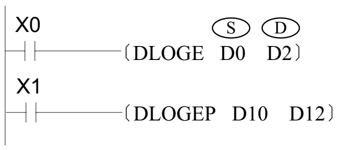
◆ Function

The natural logarithm of a binary floating-point number is calculated. The base is the mathematical constant e (approximately equal to 2.71828).

Note: The value in S must be positive. If it is 0 or negative, an operation error will occur. The error code is K6706 and stored in D8067. The error flag M8067 is set to ON to identify this error.

4

Example:



When X0 = ON, the natural logarithm of the binary floating-point number in (D1, D0) is calculated. The base is the mathematical constant e.

$$\log_e(D1、D0) \Rightarrow (D3、D2)$$

The formula for converting the natural logarithm to common logarithm is as follows (0.4342945 used for common logarithm division):

$$10^{x} = e^{\frac{x}{0.4342945}}$$

LOG: Binary floating-point logarithm operation with a base of 10

◆ Overview

The LOG instruction calculates the common logarithm of a binary floating-point number with a base of 10.

LOG S1 D		Binary floating-point logarithm operation with a base of 10	Applicable model: H3U	
S1	Data source	Binary floating-point variable whose common logarithm is to be calculated		32-bit instruction (9 steps) DLOG: Continuous execution DLOGP: Pulse execution
D1	Operation result	Unit that stores the calculated common logarithm		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

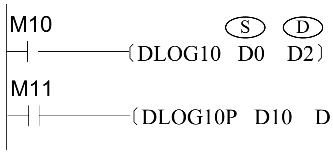
Note: The elements in gray background are supported.

◆ Function

The common logarithm of a binary floating-point number is calculated. The base is 10.

Note: The value in S must be positive. If it is 0 or negative, an operation error will occur. The error code is K6706 and stored in D8067. The error flag M8067 is set to ON to identify this error.

Example:



When M10 = ON, the common logarithm of the binary floating-point number in (D1, D0) is calculated with a base of 10.

$$\log_{10}(D1、D0) \Longrightarrow (D3、D2)$$

ESQR: Binary floating-point square root operation

◆ Overview

The ESQR instruction calculates the square root of a binary floating-point number.

<table>
<tr><th colspan="3">ESQR S D</th><th>Binary floating-point square root operation</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S</td><td>Data source</td><td colspan="2">Binary floating-point variable whose square root is to be calculated</td><td rowspan="2"></td><td rowspan="2">32-bit instruction (9 steps)
DESQR: Continuous execution
DESQRP: Pulse execution</td></tr>
<tr><td>D</td><td>Operation result</td><td colspan="2">Unit that stores the calculated square root</td></tr>
</table>

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

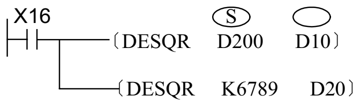
The square root of a binary floating-point number is calculated.

If the constant K or H is used as the operand S, the value is converted to a floating-point number before square root calculation.

The zero flag M8020 is set if the result of the calculation is 0.

The value in S must be positive. If it is negative, a calculation error will occur, and M8067 and M8068 are set to ON to identify this error.

Example:



The square root of the binary floating-point number in (D201, D200) is calculated $\sqrt{(D201,D200)}$. The result is stored in (D11, D10).
The square root of a floating-point number converted from K6789 is calculated. The result is stored in (D21, D20).
The constant K6789 is converted to a binary floating-point number before operation.

SQR: Binary number square root operation

◆ Overview

The SQR instruction calculates the square root of a binary number.

SQR S D			Binary number square root operation	Applicable model: H3U	
S	Data source	Data whose square root is to be calculated, or address of the word element that stores the data		16-bit instruction (5 steps) SQR: Continuous execution SQRP: Pulse execution	32-bit instruction (9 steps) DSQR: Continuous execution DSQRP: Pulse execution
D	Operation result	Address of the word element that stores the calculated square root			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

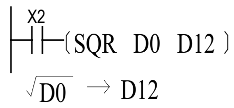
The square root of the BIN value in S is calculated. The result is stored in D.

The value in S must be positive. If it is negative, an operation error will occur, and the error flag M8067 is set to ON to identify this error and the instruction is not executed.

The operation result D must be an integer. The borrow flag M8021 is set to ON when the fractional part (if any) of the operation result is discarded.

The zero flag M8020 is set to ON when the operation result is 0.

Example:



If D0 = K100, then D12 = K10 when X2 = ON.
If D0 = K110, then D12 = K10 when X2 = ON and the fractional part is discarded.

POW: Floating-point weight instruction

◆ Overview

The POW instruction performs a mathematical operation where the binary floating-point number in [S1+1, S1] is raised to the exponent in [S2+1, S2]. The result is stored in [D+1, D].

POW S1 S2 D		Floating-point weight instruction	Applicable model: H3U	
S1	Base	Head address of elements that store the base, which must be a non-zero number		32-bit instruction (13 steps) DPOW: Continuous execution DPOWP: Pulse execution
S2	Exponent	Head address of elements that store the exponent		
D	Result	Head address of elements that store the operation result		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

As the POW instruction uses only the floating-point number format, the values in [S1] and [S2] must be converted to floating-point numbers.

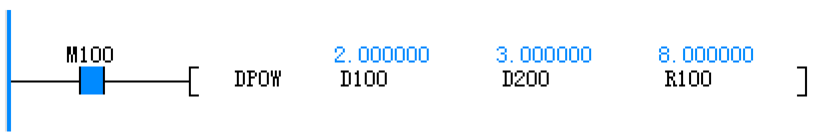
For example, assume that [S1] = 2 and [S2] = 3, then [D] = 2^3 = 8.

1. The carry flag M8022 is set to ON if the absolute value of the operation result is greater than the maximum floating-point value.

2. The borrow flag M8021 is set to ON if the absolute value of the operation result is less than the minimum floating-point value.

3. The zero flag M8020 is set to ON if the operation result is 0.

◆ Application



4.5 Data Processing

Data conversion	INT	Conversion from binary floating-point number format to BIN integer format
	BCD	Conversion from binary to BCD format
	BIN	Conversion from BCD to binary format
	FLT	Conversion from binary integer format to binary floating-point number format
	EBCD	Conversion from binary to decimal floating-point number format
	EBIN	Conversion from decimal to binary floating-point number format
	DABIN	Conversion from decimal ASCII format to BIN format
	BINDA	Conversion from BIN format to decimal ASCII format
	WTOB	Data separation by byte
	BTOW	Data combination by byte
	UNI	4-bit combination of 16-bit data
	DIS	4-bit separation of 16-bit data
	ASCI	Conversion from HEX to ASCII format
	HEX	Conversion from ASCII to HEX format
Data transfer	MOV	Value transfer
	EMOV	Binary floating point transfer
	SMOV	Shifted transfer
	BMOV	Batch data transfer
	FMOV	One-to-multiple data transfer
	CML	Inverted data transfer
	ZPUSH	Index register batch storage
	ZPOP	Index register batch recovery
Table operation	ZRST	Full data reset
	SORT	Data sorting
	SORT2	Data sorting 2
	SER	Data search
	FDEL	Deletion of data from a table
	FINS	Insertion of data to a table
	POP	Last-in data read
	RAMP	Ramp instruction
Data rotation and shift	ROR	Rotation right
	ROL	Rotation left
	RCR	Rotation right with carry
	RCL	Rotation left with carry
	SFTR	Bit shift right
	SFTL	Bit shift left
	WSFR	Word shift right
	WSFL	Word shift left
	SFWR	FIFO data write
	SFRD	FIFO data read
	SFR	16-bit data shift right with carry by n bits
	SFL	16-bit data shift left with carry by n bits

4

Other data processing	SWAP	Higher and lower byte swap
	BON	ON bit check
	SUM	Total number of ON bits
	RND	Random number generation
	XCH	Data exchange
	ANS	Annunciator setting
	ANR	Annunciator reset

4.5.1 Data Conversion

Data conversion	INT	Conversion from binary floating-point number format to BIN integer format
	BCD	Conversion from binary to BCD format
	BIN	Conversion from BCD to binary format
	FLT	Conversion from binary integer format to binary floating-point number format
	EBCD	Conversion from binary to decimal floating-point number format
	EBIN	Conversion from decimal to binary floating-point number format
	DABIN	Conversion from decimal ASCII format to BIN format
	BINDA	Conversion from BIN format to decimal ASCII format
	WTOB	Data separation by byte
	BTOW	Data combination by byte
	UNI	4-bit combination of 16-bit data
	DIS	4-bit separation of 16-bit data
	ASCI	Conversion from HEX to ASCII format
	HEX	Conversion from ASCII to HEX format

4

INT: Conversion from binary floating-point number format to BIN integer format

◆ Overview

The INT instruction performs rounding of binary floating-point numbers to remove the fractional part. The result is stored in D.

INT S D		Conversion from binary floating-point number format to BIN integer format	Applicable model: H3U	
S	Data source	Binary floating-point variable to be rounded	16-bit instruction (5 steps) INT: Continuous execution INTP: Pulse execution	32-bit instruction (9 steps) DINT: Continuous execution DINTP: Pulse execution
D	Operation result	Unit that stores the resulting BIN integer		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Binary floating-point numbers are rounded to remove the fractional part. The result is stored in D.

M8020 is set when S = 0.

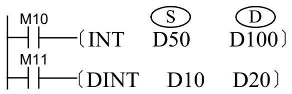
The borrow flag M8021 is set when |S| ≤ 1.

The carry flag M8022 is set if the operation result exceeds either of the following ranges (which results in an overflow):

16-bit instruction: –32,768 to +32,767

32-bit instruction: –2,147,483,648 to +2,147,483,647

Example:



The floating-point number in (D51, D50) is rounded.
The result is stored in (D100).
The floating-point number in (D11, D10) is rounded.
The result is stored in (D21, D20).
The results of the INT and DINT instructions are stored in different locations.

BCD: Conversion from binary to BCD format

◆ Overview

The BCD instruction converts binary numbers to binary coded decimal (BCD) equivalents.

<table>
<tr><th colspan="3">BCD S D</th><th>Conversion from binary to BCD format</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S</td><td>Data source</td><td colspan="2">Binary data, or address of the word element that stores the binary data</td><td rowspan="2">16-bit instruction (5 steps)
BCD: Continuous execution
BCDP: Pulse execution</td><td rowspan="2">32-bit instruction (9 steps)
DBCD: Continuous execution
DBCDP: Pulse execution</td></tr>
<tr><td>D</td><td>Conversion result</td><td colspan="2">Address of the word element that stores the BCD equivalent of a binary number</td></tr>
</table>

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

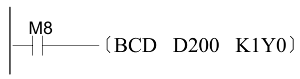
Note: The elements in gray background are supported.

◆ Function

The BCD instruction requires contact driving and has two operands. It converts the BIN value in S to a BCD number. The result is stored in D. The BCD instruction is generally used for data format processing before data is displayed.

When the 16-bit instruction is used, the value range is 0 to 9999; when the 32-bit instruction is used, it is 0 to 99,999,999. An error will occur when the conversion result exceeds 9999 or 99,999,999. M8067 and M8068 are set to ON to identify this error, and the error code is stored in D8067.

Example:



The BIN value in D200 is converted to a BCD equivalent.
The digit in the ones place of the result is stored in K1Y0 (four bit components Y0 to Y3).
If D200 = H000E (hexadecimal) = K14 (decimal), then Y0–Y3 = 0100 (BIN).
If D200 = H0028 (hexadecimal) = K40 (decimal), then Y0–Y3 = 0000 (BIN).

BIN: Conversion from BCD to binary format

◆ Overview

The BIN instruction converts BCD numbers to binary equivalents.

<table>
<tr><th colspan="3">BIN S D</th><th>Conversion from BCD to binary format</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S</td><td>Data source</td><td colspan="2">BCD data, or address of the word element that stores the data</td><td rowspan="2">16-bit instruction (5 steps)
BIN: Continuous execution
BINP: Pulse execution</td><td rowspan="2">32-bit instruction (9 steps)
DBIN: Continuous execution
DBINP: Pulse execution</td></tr>
<tr><td>D</td><td>Conversion result</td><td colspan="2">Address of the word element that stores the binary equivalent of a BCD number</td></tr>
</table>

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

4

◆ Function

The BIN instruction requires contact driving and has two operands. It converts the BCD value in S to a BIN number. The result is stored in D. The instruction is generally used to convert the data (for example, encoder disk setting) read by external ports to BIN data that can be directly used in operation.

The BCD value in S must be in the range 0 to 9999 in 16-bit operation or 0 to 99,999,999 in 32-bit operation.

When the data in S is not in BCD format (Hex indicates any digit beyond the range 0 to 9), an operation error will occur, and M8067 and M8068 are set to identify this error.

Example:

When M8 is set, the BCD value in K1Y0 is conv
to a BIN equivalent. The result is stored in D200

FLT: Conversion from binary integer format to binary floating-point number format

◆ Overview

The FLT instruction converts binary integers to binary floating-point numbers.

FLT S D			Conversion from binary integer format to binary floating-point number format	Applicable model: H3U	
S	Integer	Binary integer to be converted, or address of the word element that stores the binary integer		16-bit instruction (5 steps) FLT: Continuous execution FLTP: Pulse execution	32-bit instruction (9 steps) DFLT: Continuous execution DFLTP: Pulse execution
D	Floating-point number	Address of the word element that stores the floating-point number after conversion			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

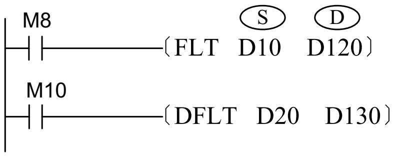
◆ Function

The integer in S is converted to a floating-point number. The result is stored in D and D+1.

The constants K and H are automatically converted in all floating point operation instructions and therefore cannot be used in the FLT instruction.

The INT instruction is the inverse function of FLT.

Example 1:



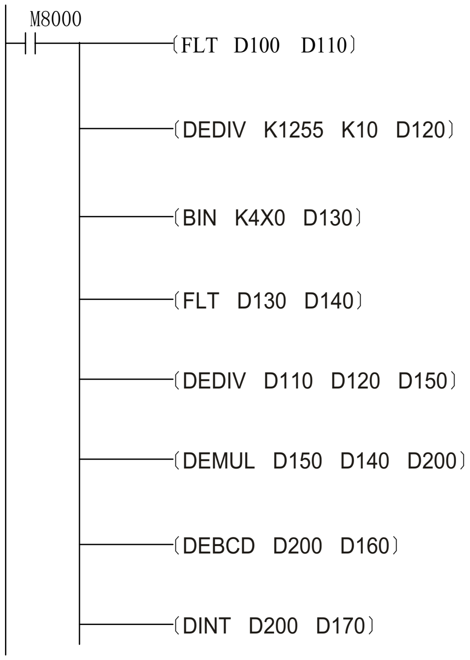
When M8 = ON, the 16-bit BIN integer in D10 is converted to a binary floating-point number. The result is stored in (D121, D120).

When M10 = ON, the 32-bit BIN integer in (D21, D20) is converted to a binary floating-point number. The result is stored in (D131, D130).

Example 2:

Use instructions to complete the following floating-point number operations.

D100/K125.5*(X17~X0)=D200



The BIN integer in D100 is converted to a binary floating-point number. The result is stored in (D111, D110).

The value in K1255 is divided by that in K10. The result (a binary floating-point number) is stored in (D121, D120).

The BCD value in X17 to X0 is converted to a 16-bit BIN integer. The result is stored in D130.

The BIN integer in D130 is converted to a binary floating-point number. The result is stored in (D141, D140).

The binary floating-point number in (D111, D110) is divided by that in (D121, D120). The result (a binary floating-point number) is stored in (D151, D150).

The binary floating-point number (D151, D150) is multiplied by that in (D141, D140). The result (a binary floating-point number) is stored in (D201, D200).

The binary floating-point number in (D201, D200) is converted to a decimal equivalent. The result is stored in (D161, D160). This function is used for decimal floating-point number monitoring.

The binary floating-point number in (D201, D200) is converted to a 32-bit BIN integer. The result is stored in (D171, D170).

EBCD: Conversion from binary to decimal floating-point number format

◆ Overview

The EBCD instruction converts binary floating-point numbers to decimal equivalents.

EBCD S D			**Conversion from binary to decimal floating-point number format**	**Applicable model: H3U**
S	Data source	Binary floating-point variable		32-bit instruction (9 steps) DEBCDP: Continuous execution
D	Operation result	Unit that stores the decimal equivalent of a binary floating-point number		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Binary floating-point numbers are converted to decimal equivalents.

Example:

The binary floating-point number in (D3, D2) is converted to a decimal equivalent. The result is stored in (D11, D10).

For the binary floating-point number in (D3, D2), the real number occupies 23 bits, the exponent occupies eight bits, and the sign occupies one bit.

For the decimal floating-point number in (D11, D10), the exponent (D3) and real number (D2) are expressed as D2 x 10^{D3} in scientific notation.

The PLC uses only binary floating-point numbers for calculation. Convert binary floating-point numbers to decimal equivalents for easy monitoring.

EBIN: Conversion from decimal to binary floating-point number format

◆ Overview

The EBIN instruction converts decimal floating-point numbers to binary equivalents.

EBIN S D			Conversion from decimal to binary floating-point number format	Applicable model: H3U
S	Data source	Decimal floating-point variable		32-bit instruction (9 steps) DEBINP: Jump execution
D	Result	Unit that stores the binary equivalent of a decimal floating-point number		

◆ Operands

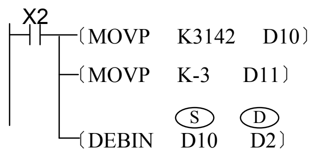
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Decimal floating-point numbers are converted to binary equivalents.

Example:



The decimal floating-point number 3.142 (stored in D11 and D10) is converted to a binary equivalent. The result is stored in (D3, D2).

4

DABIN: Conversion from decimal ASCII format to BIN format

◆ Overview

The DABIN instruction converts ASCII-encoded decimal numbers (30H to 39H) to BIN numbers.

DABIN S D			Conversion from decimal ASCII format to BIN format	Applicable model: H3U	
S	Input value	Start number of elements that store the ASCII-encoded decimal number to be converted to a BIN number		16-bit instruction (5 steps) DABIN: Continuous execution DABINP: Pulse execution	32-bit instruction (9 steps) DDABIN: Continuous execution DDABINP: Pulse execution
D	Output value	Number of the element that stores the conversion result			

◆ Operands

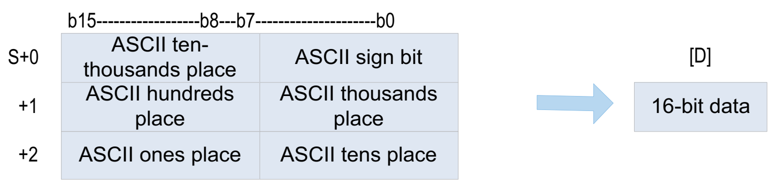
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z		Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z		Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

- The ASCII-encoded decimal number (30H to 39H) stored in [S] to [S+2] is converted to a 16-bit BIN number. The result is stored in [D].



- The value stored in [S] to [S+2] ranges from –32,768 to +32,767.
- When the number to be converted is positive, the sign (lowest byte) is set to 20H (space). When the number to be converted is negative, the sign is set to 2DH (minus sign).
- The ASCII code of every digit is in the range 30H to 39H.
- When the ASCII code of every digit is 20H (space) or 00H (NULL), it is processed as 30H.

2) 32-bit instruction

- The ASCII-encoded decimal number (30H to 39H) stored in [S] to [S+5] is converted to a 32-bit BIN number. The result is stored in [D+1, D].

	b15------------------b8	b7---------------------b0
S+0	ASCII billions place	ASCII sign bit
+1	ASCII ten-millions place	ASCII hundred-millions place
+2	ASCII hundred-thousands place	ASCII millions place
+3	ASCII thousands place	ASCII ten-thousands place
+4	ASCII tens place	ASCII hundreds place
+5	Ignored	ASCII ones place

→

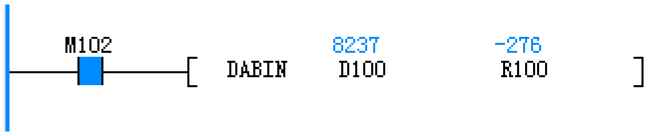
[D+1]	[D]
Higher 16 bits	Lower 16 bits

- The value stored in [S] to [S+5] ranges from –2,147,483,648 to +2,147,483,647. The higher bytes in [S+5] are ignored.
- The ASCII code of every digit is in the range 30H to 39H.
- When the ASCII code of every digit is 20H (space) or 00H (NULL), it is processed as 30H.

An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6706 is returned when the sign bit is not 20H (space) or 2DH (minus sign).
- Error 6706 is returned when the ASCII codes of data bits exceed the range 30H to 39H or are not 20H (space) or 00H (NULL).
- Error 6706 is returned when the number after conversion exceeds the 16- or 32-bit signed value range.
- Error 6705 is returned when elements in [S+2] (16-bit operation) or [S+5] (32-bit operation) are out of range.

◆ Application



	Element Name	data type	display format	current value
1	D100	16-bit int	Hex	0x202D
2	D101	16-bit int	Hex	0x3220
3	D102	16-bit int	Hex	0x3637
4		16-bit int	Dec	
5	R100	16-bit int	Dec	-276

BINDA: Conversion from BIN format to decimal ASCII format

◆ Overview

The BINDA instruction converts BIN numbers to ASCII-encoded decimal numbers (30H to 39H).

BINDA S D		Conversion from BIN format to decimal ASCII format	Applicable model: H3U	
S	Input value	Number of the element that stores the BIN number to be converted to an ASCII-encoded decimal number	16-bit instruction (5 steps) BINDA: Continuous execution BINDAP: Pulse execution	32-bit instruction (9 steps) DBINDA: Continuous execution DBINDAP: Pulse execution
D	Output value	Number of the element that stores the conversion result		

◆ Operands

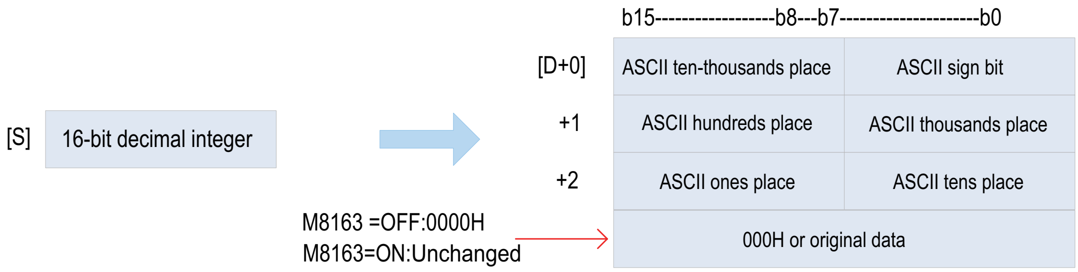
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

- Every digit of the 16-bit BIN number in [S] is ASCII encoded (30H to 39H) in decimal format. The result is stored in elements from head address [D].

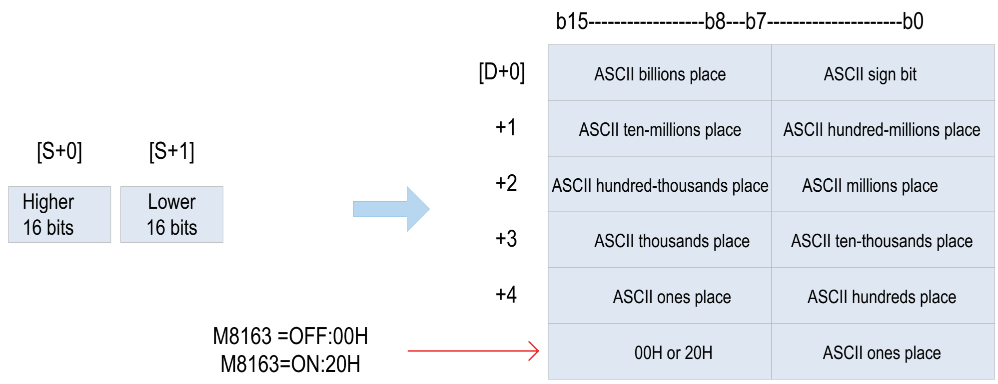


- The value in [S] ranges from –32,768 to +32,767.
- The operation result is as follows:

- When the 16-bit number is positive, the sign bit is set to 20H (space). When it is negative, the sign bit is set to 2DH (minus sign).
- When 0 exists on the left of valid digits, the sign bit is set to 20H (space).
- The value in [D+3] is determined based on whether M8163 is set to ON or OFF.

2) 32-bit instruction

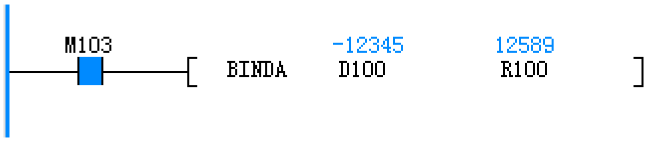
- Every digit of the 32-bit BIN number in [S+1, S] is ASCII encoded (30H to 39H) in decimal format. The result is stored in elements from head address [D].



- The value in [S+1, S] ranges from –2,147,483,648 to +2,147,483,647.
- The operation result is as follows:

- When the 16-bit number is positive, the sign bit is set to 20H (space). When it is negative, the sign bit is set to 2DH (minus sign).
- When 0 exists on the left of valid digits, the sign bit is set to 20H (space).
- The higher bytes in [D+5] are determined based on whether M8163 is set to ON or OFF.

◆ Application



Output Window

	Element Name	data type	display form	current value
1	D100	16-bit int	Dec	-12345
2	R100	16-bit int	Hex	0x312D
3	R101	16-bit int	Hex	0x3332
4	R102	16-bit int	Hex	0x3534
5	R103	16-bit int	Hex	0x0
6	R104	16-bit int	Hex	0x0
7	M8163	BOOL	Bin	OFF

WTOB: Data separation by byte

◆ Overview

The WTOB instruction separates consecutive 16-bit data entries byte by byte (every eight bits).

WTOB S D n			Data separation by byte	Applicable model: H3U
S	Source data	Start number of elements that store the data to be separated byte by byte	16-bit instruction (7 steps) WTOB: Continuous execution WTOBP: Pulse execution	
D	Result	Start number of elements that store the data already separated byte by byte		
n	Separated byte count	Number of bytes to be separated (n ≥ 0; no processing when n = 0)		

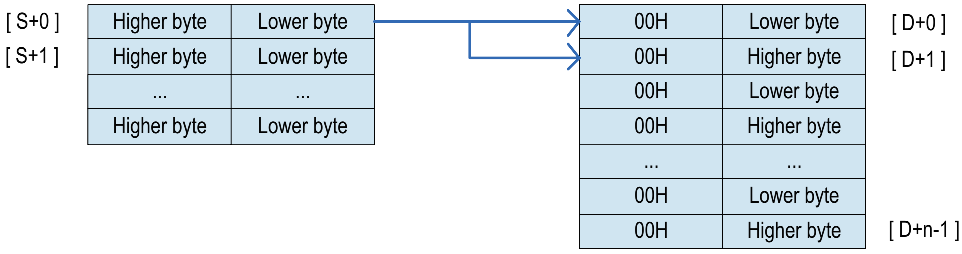
◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

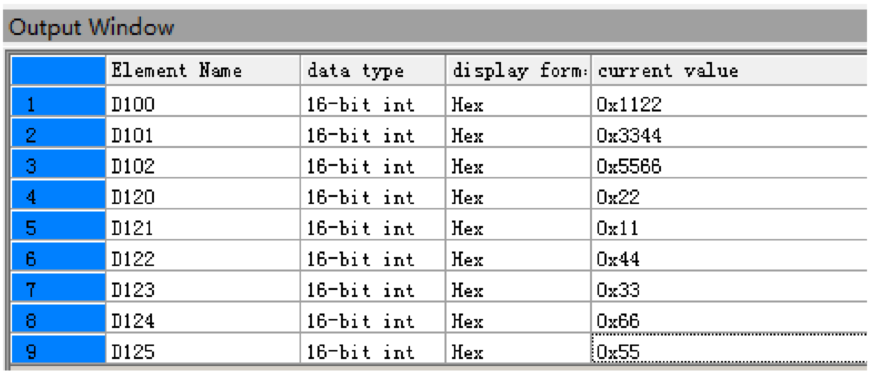
The 16-bit data in elements from head address [S] is saved to the lower eight bits in each of n elements from head address [D]. The higher eight bits store 00H. The data is stored byte by byte.



An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

Error 6705 is returned when elements from head addresses [S] and [D] are out of range.

◆ Application



	Element Name	data type	display form	current value
1	D100	16-bit int	Hex	0x1122
2	D101	16-bit int	Hex	0x3344
3	D102	16-bit int	Hex	0x5566
4	D120	16-bit int	Hex	0x22
5	D121	16-bit int	Hex	0x11
6	D122	16-bit int	Hex	0x44
7	D123	16-bit int	Hex	0x33
8	D124	16-bit int	Hex	0x66
9	D125	16-bit int	Hex	0x55

BTOW: Data combination by byte

◆ Overview

The BTOW instruction combines the lower eight bits (lower byte) of each of consecutive 16-bit data entries together.

BTOW S D n			Data combination by byte	Applicable model: H3U
S	Source data	Start number of elements that store the data to be combined byte by byte	16-bit instruction (7 steps) BTOW: Continuous execution BTOWP: Pulse execution	
D	Result	Start number of elements that store the data already combined byte by byte		
n	Combined byte count	Number of bytes to be combined (n ≥ 0; no processing when n = 0)		

◆ Operands

Operand	Bit Element							Word Element													
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The lower eight bits of each of n 16-bit data entries in elements from head address [S] are combined. The 16-bit data after combination is stored in elements from head address [D]. The higher eight bits of every source data entry from head address [S] are ignored.

An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

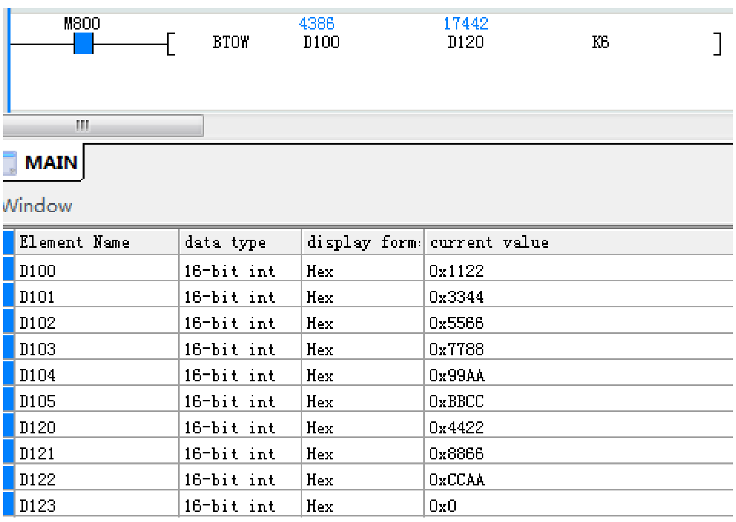
Error 6706 is returned when elements from head addresses [S] and [D] are out of range.

[D+0]	Higher byte	First byte
[D+1]	Higher byte	Second byte
	Higher byte	Third byte
	Higher byte	Fourth byte
	...	...
	Higher byte	(*n* – 1)th byte
[D+n-1]	Higher byte	*n*th byte

Second byte	First byte	[S+0]
Fourth byte	Third byte	[S+1]
...	...	
*n*th byte	(*n* – 1)th byte	

◆ Application

The lower eight bits of each of the six data entries from head address D100 are combined into 16-bit data. The result is stored in three elements from head address D120.



Element Name	data type	display form	current value
D100	16-bit int	Hex	0x1122
D101	16-bit int	Hex	0x3344
D102	16-bit int	Hex	0x5566
D103	16-bit int	Hex	0x7788
D104	16-bit int	Hex	0x99AA
D105	16-bit int	Hex	0xBBCC
D120	16-bit int	Hex	0x4422
D121	16-bit int	Hex	0x8866
D122	16-bit int	Hex	0xCCAA
D123	16-bit int	Hex	0x0

4

UNI: 4-bit combination of 16-bit data

◆ Overview

The UNI instruction combines the lower four bits of each of consecutive 16-bit data entries together.

UNI S D n			4-bit combination of 16-bit data	Applicable model: H3U
S	Source data	Start number of elements that store the data to be combined	16-bit instruction (7 steps) UNI: Continuous execution UNIP: Pulse execution	
D	Result	Number of the element that stores the data after combination		
n	Combined data count	Number of data entries to be combined (value range: 0 to 4; no processing when n = 0)		

◆ Operands

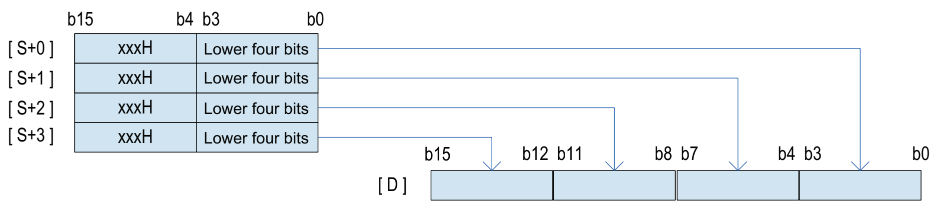
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The lower four bits of each of n 16-bit data entries from head address S are combined. The 16-bit data after combination is stored in D.

n ranges from 1 to 4. The instruction is not executed when n = 0. When n is in the range 1 to 3, the remaining higher bits are filled with 0s.



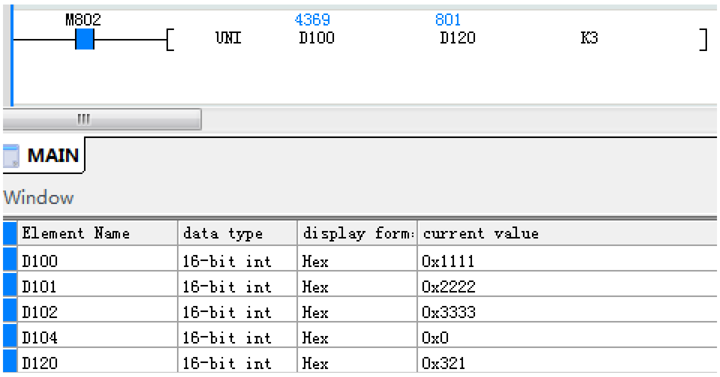
An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error K6705 is returned when the start number S is out of range.

2. Error K6706 is returned when the value of n is out of range.

◆ Application

The lower four bits of each of the three elements from head address D100 are combined into 16-bit data, and the remaining bits are filled with 0s. The result is stored in D120.



DIS: 4-bit separation of 16-bit data

◆ Overview

The DIS instruction separates 16-bit data by every four bits.

DIS S D n			4-bit separation of 16-bit data	Applicable model: H3U
S	Source data	Number of the element that stores the data to be separated	16-bit instruction (7 steps) DIS: Continuous execution DISP: Pulse execution	
D	Result	Start number of the elements that store the data after separation		
n	Separated data count	Number of separated data entries (value range: 0 to 4; no processing when n = 0)		

◆ Operands

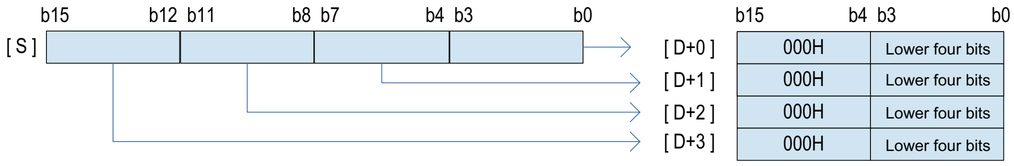
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The 16-bit data in S is separated by every four bits. The data after separation is stored in the lower four bits of each of the elements from head address D. The other 12 bits are filled with 0s.

n ranges from 1 to 4. The instruction is not executed when n = 0.



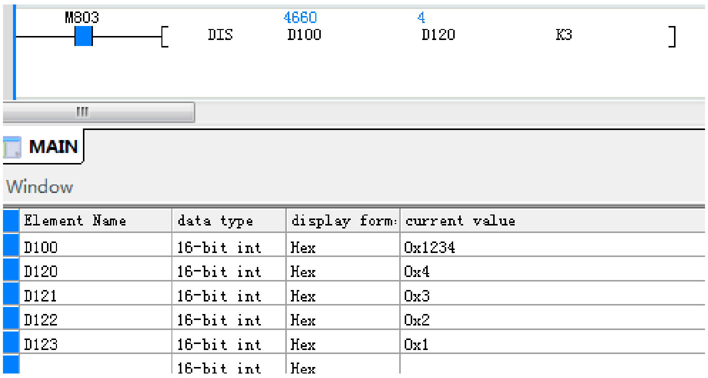
An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

Error K6705 is returned when the start number D is out of range.

Error K6706 is returned when n is out of range.

◆ Application

The 16-bit data in D100 is separated by every four bits. The result is stored in three consecutive D elements from head address D120.



ASCI: Conversion from HEX to ASCII format

◆ Overview

The ASCI instruction encodes the value in S in ASCII format. The result is stored in variables from head address D.

ASCI S D n			Conversion from HEX to ASCII format	Applicable model: H3U
S	Data source	Address of the variable or the numeric constant to be converted	16-bit instruction (7 steps) ASCI: Continuous execution ASCIP: Pulse execution	
D	Conversion result	Head address of variables that store the ASCII characters after conversion		
n	Converted character count	Number of converted characters; value range: 1 to 256		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The value in S is encoded in ASCII format. The result is stored in variables from head address D.

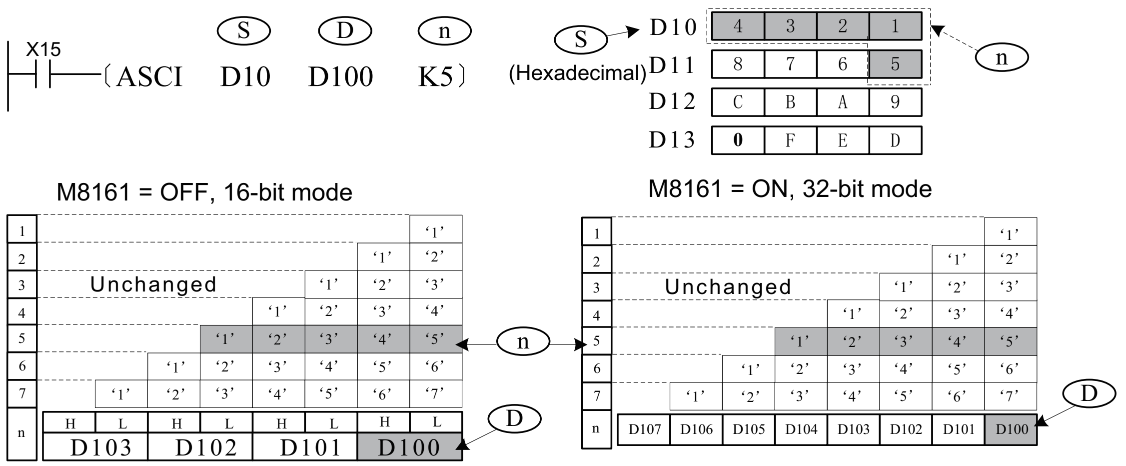
S is the address of the variable or the numeric constant to be converted.

D is the head address for storing the ASCII characters after conversion

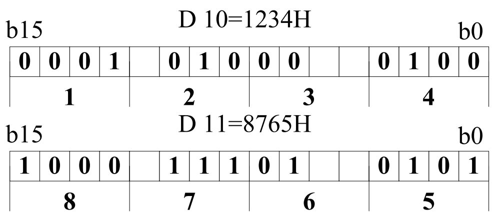
n is the number of converted characters, in the range 1 to 256.

The ASCI instruction conforms to the ASCII-HEX mapping table. For example, 0 in ASCII format corresponds to H30 in hexadecimal format, and F in ASCII format corresponds to H46 in hexadecimal format. For details about HEX-ASCII mapping, see "ASC: ASCII code conversion" on Page 322.

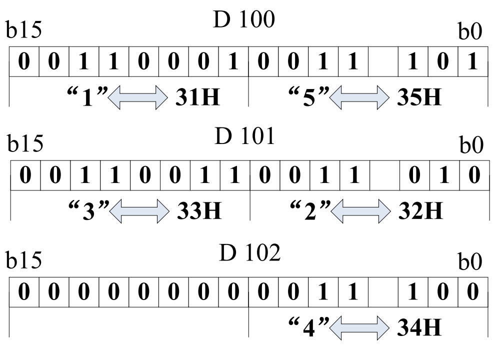
◆ Example:



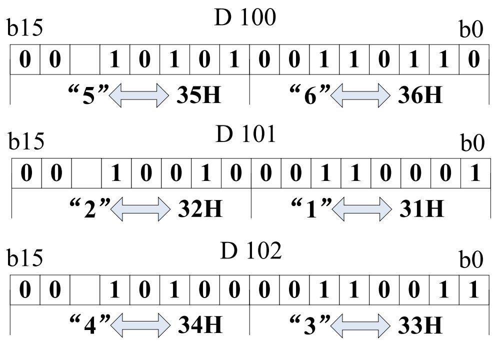
The M8161 flag determines the width mode of the target variable that stores the calculation result. When M8161 = OFF, the 16-bit mode is enabled, whereby the calculation result is stored in the higher and lower bytes of the variable separately. When M8161 = ON, the 8-bit mode is enabled, whereby the calculation result is stored only in the lower byte of the variable. Therefore, the length of the actually used variable area is increased.



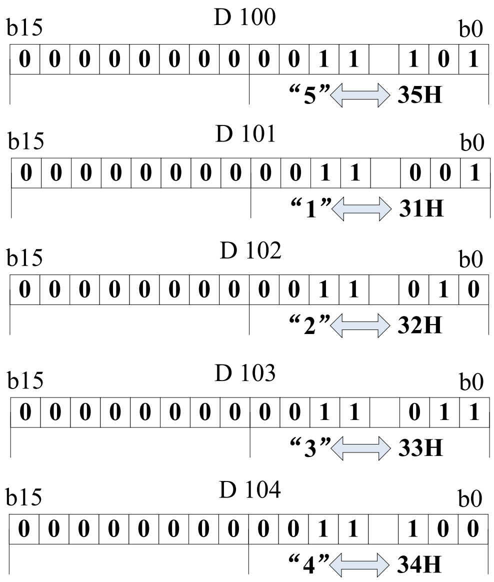
Bit composition when M8161 = OFF and n = 5
Conversion of D10 and D11



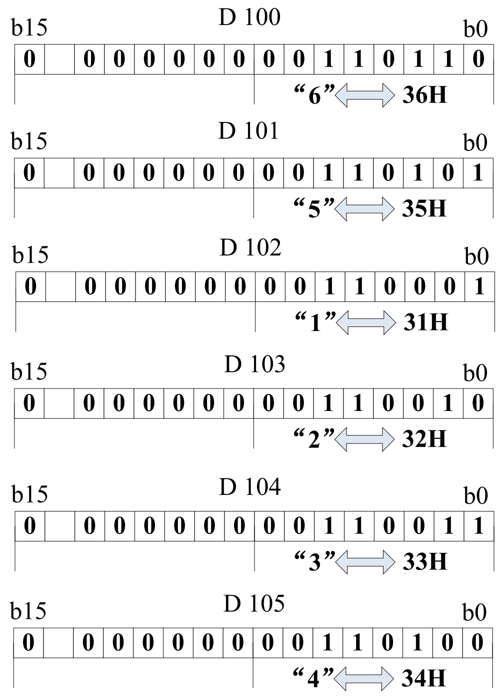
Bit composition when M8161 = OFF and n = 6
Conversion of D10 and D11



Bit composition when M8161 = ON and n = 5
Conversion of D10 and D11



Bit composition when M8161 = ON and n = 6
Conversion of D10 and D11



Note: The RS, HEX, ASCI, and CCD instructions share the M8161 flag. Pay special attention during programming.

HEX: Conversion from ASCII to HEX format

◆ Overview

The HEX instruction converts the values of variables from head address S to hexadecimal equivalents. The result is stored in variables from head address D. The number of converted characters and storage mode are configurable.

HEX S D n			Conversion from ASCII to HEX format	Applicable model: H3U	
S	Data source	Head address of variables or the numeric constants to be converted. If register variables are converted, 32-bit variable width (four ASCII characters) is used.		16-bit instruction (7 steps) HEX: Continuous execution HEXP: Pulse execution	
D	Conversion result	Head address of variables that store the hexadecimal characters after conversion. The occupied variable space is related to S2.			
n	Converted character count	Number of converted characters			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

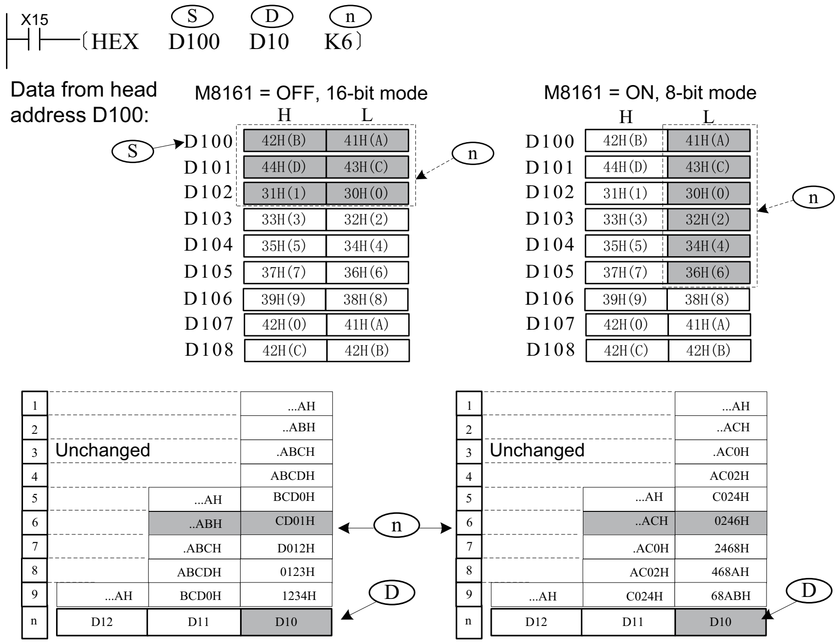
The values of variables from head address S are converted to hexadecimal equivalents. The result is stored in variables from head address D. The number of converted characters and storage mode are configurable.

S is the head address of variables or the numeric constants to be converted. If register variables are converted, 32-bit variable width (four ASCII characters) is used.

D is the head address for storing the hexadecimal characters after conversion. The occupied variable space is related to n.

n is the number of converted characters, in the range 1 to 256.

Example:



The M8161 flag determines which variable width mode to use. When M8161 = OFF, the 16-bit mode is enabled, whereby the higher and lower bytes of variables are taken for the operation. When M8161 = ON, the 8-bit mode is enabled, whereby only the lower bytes of variables are taken for the operation and the higher bytes are discarded. Therefore, the length of the actually used variable area S is increased.

Bit composition when M8161 = OFF and n = 5
Conversion of D100 to D102
(higher and lower bytes)

D 10

1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0
B				C				D				0			

D 11

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
												A			

Bit composition when M8161 = OFF and n = 6
Conversion of D100 to D102
(higher and lower bytes)

D 10

1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1
C				D				0				1			

D 11

0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1
								A				B			

Bit composition when M8161 = ON and n = 5
Conversion of D100 to D104
(lower bytes)

D 10

1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0
C				0				2				4			

D 11

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
												A			

Bit composition when M8161 = ON and n = 6
Conversion of D100 to D105
(lower bytes)

D 10

0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0
0				2				4				6			

D 11

0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0
								A				C			

Note:

The RS, HEX, ASCI, and CCD instructions share the M8161 flag. Pay special attention during programming.

The source data in the S data area must be ASCII characters; otherwise, a conversion error will occur.

If the output data format is BCD, BCD-to-BIN conversion must be performed on the hexadecimal characters after conversion to get the correct value.

4.5.2 Data Transfer

MOV: Value transfer

◆ Overview

The MOV instruction copies the data at the source address S to the destination address D.

MOV S D			Value transfer	Applicable model: H3U	
S	Data source	Data to be transferred, or address of the word element that stores the data		16-bit instruction (5 steps) MOV: Continuous execution MOVP: Pulse execution	32-bit instruction (9 steps) DMOV: Continuous execution DMOVP: Pulse execution
D	Destination to which data is copied	Address of the word element that stores the data copied to a destination			

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

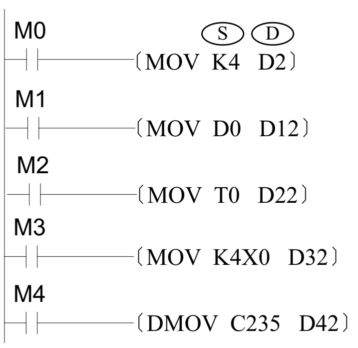
The MOV instruction requires contact driving and has two operands. It copies the value in S to D.

When the 32-bit instruction (DMOV) is executed, S and D use adjacent variable units with high-numbered addresses.

For example, the statement [DMOV D1 D5] gets the result D1 -> D5; D2 -> D6.

Example:

Ladder chart



Instruction list

```
LD    M0
MOV   K4    D2
LD    M1
MOV   D0    D12
LD    M2
MOV   T0    D22
LD    M3
MOV   K4X0  D32
LD    M4
DMOV  C235  D42
```

When M0 = ON, the content of K4 is copied to D2. When M0 switches from ON to OFF, the content (K4) of D2 remains unchanged, unless the user program modifies the value in D2 again. The value in D2 changes to 0 when the PLC switches from STOP to RUN or is powered on again. The value remains unchanged when the registers with support for retention upon power failure are powered on or switch from STOP to RUN.

EMOV: Binary floating point transfer

◆ Overview

The EMOV instruction transfers binary floating-point numbers. Contact driving is required. After the instruction is executed, the value of the binary floating-point number in S is copied to D.

EMOV S D			Binary floating point transfer	Applicable model: H3U
S	Data source	Source from which a binary floating-point number is transferred		32-bit instruction (9 steps) DEMOV: Continuous execution DEMOVP: Pulse execution
D	Transfer destination	Unit that stores the binary floating-point number transferred to a destination		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

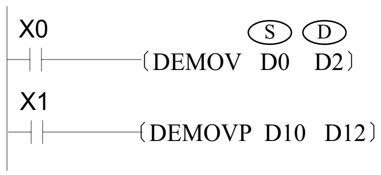
◆ Function

The EMOV instruction transfers binary floating-point numbers. Contact driving is required. After the instruction is executed, the value of the binary floating-point number in S is copied to D.

S is the source from which a binary floating-point number is transferred.

D is the unit that stores the binary floating-point number transferred to a destination.

Example:



Assume that the binary floating-point value in (D1, D0) is 12.345. When X0 = ON, the binary floating-point value in (D3, D2) changes to 12.345. When M0 switches from ON to OFF, the value 12.345 in (D3, D2) remains unchanged, unless the user program modifies the value again. The value in (D3, D2) is changed when the PLC switches from STOP to RUN or is powered on again. The value remains unchanged when the registers with support for retention upon power failure are powered on or switch from STOP to RUN.

SMOV: Shifted transfer

◆ Overview

The SMOV instruction transfers m2 bits starting from the m1th bit in S to m2 bits starting from the nth bit in D.

SMOV S m1 m2 D n		Shifted transfer	Applicable model: H3U	
S	Data source	Address of the word element that stores the bits to be transferred	16-bit instruction (11 steps) SMOV: Continuous execution SMOVP: Pulse execution	
m1	Initial bit to be transferred	Position of the initial bit in S to be transferred		
m2	Transferred bit count	Number of bits in S to be transferred		
D	Destination operand	Address of the word element that stores the bits transferred to a destination		
n	Initial bit at the destination	Position of the initial bit transferred to D		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The SMOV instruction requires contact driving and has a maximum of five operands, which are described as follows:

S is the data source from which bits will be transferred. When M8168 = OFF, the BCD mode (decimal bits) is enabled. The M operand is in the range 0000 to 9999 and cannot be a negative number. When M8168 = ON, the BIN mode is enabled. The S operand can be a negative number.

m1 is the number of the initial bit to be transferred. The value range is 1 to 4.

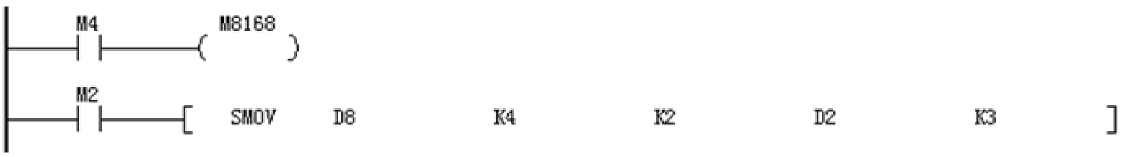
m2 is the number of bits to be transferred. The value range is 1 to m1.

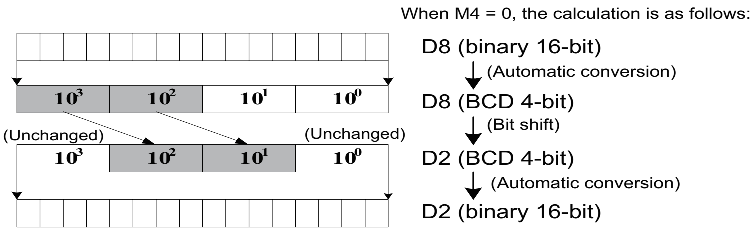
D is the destination variable to which bits are transferred.

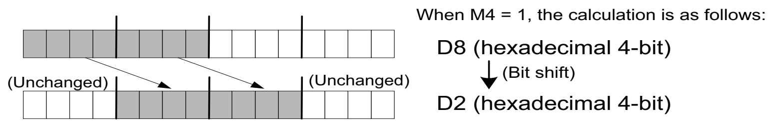
n is the initial bit of the destination variable that stores a transferred bit. The value range is m2 to 4.

The data bit transfer process is related to the status of the special flag M8168. When M8168 = OFF, the BCD mode (decimal bits) is enabled When M8168 = ON, the BIN mode is enabled, whereby every four bits (hexadecimal) are transferred at a time as a whole unit.

◆ **Example:**







Assume that D8 = K1234 and D2 = K5678. When M8168 = OFF (BCD mode enabled), the value in D2 changes to K5128 if M2 is set to ON.

When M8168 = ON (BIN mode enabled) with D8 = H04D2 = K1234 and D2 = H162E = K5678, then D2 = H104E = K4174 if M2 is set to ON.

BMOV: Batch data transfer

◆ **Overview**

When driving conditions are met, the BMOV instruction transfers the data of n registers from head address S to the n registers from head address D.

BMOV S D n		**Batch data transfer**	**Applicable model:** H3U
S	Data source head address	Head address of word elements that store the data to be transferred in batches	16-bit instruction (7 steps) BMOV: Continuous execution BMOVP: Pulse execution
D	Transfer destination head address	Head address of word elements that store the data arriving at a destination	
n	Data length	Number of word elements whose data will be transferred in batches	

◆ **Operands**

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

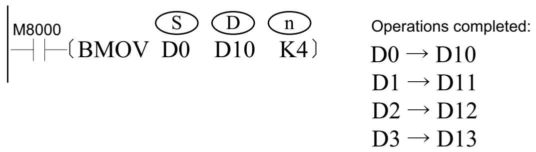
Note: The elements in gray background are supported.

◆ Function

The BMOV instruction requires contact driving and has three operands. It copies the values of n variables from head address S to n units from head address D.

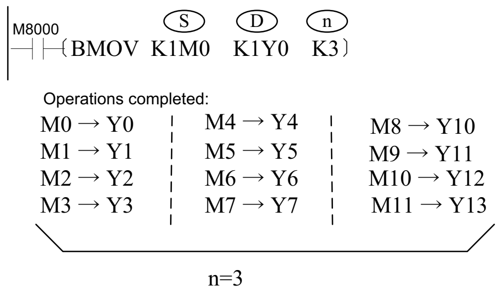
The value of n ranges from 1 to 512.

When the special variable M8024 is set to 1, the batch transfer direction is inverted. That is, the values of the n variables from head address D are copied to the n units from head address S.



When bit elements are used as the operands, S and D must have the same number of bits.

◆ Application



FMOV: One-to-multiple data transfer

◆ Overview

When driving conditions are met, the FMOV instruction transfers the data in S to n registers from head address D.

FMOV S D n			**One-to-multiple data transfer**	**Applicable model: H3U**	
S	Data source	Data to be transferred to n registers, or address of the word element that stores the data		16-bit instruction (7 steps) FMOV: Continuous execution FMOVP: Pulse execution	32-bit instruction (13 steps) DFMOV: Continuous execution DFMOVP: Pulse execution
D	Transfer destination head address	Head address of word elements that store the data arriving at a destination			
n	Target number	Number of word elements that store the data arriving at a destination			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

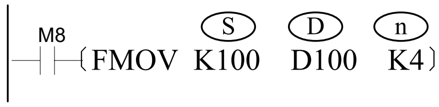
◆ Function

The FMOV instruction requires contact driving and has three operands. It copies the data in S to n units from head address D.

The value of n ranges from 1 to 512.

FMOV is a 16-bit multi-point transfer instruction, whereas DFMOV is 32-bit.

◆ Application



Operations completed when M8 = ON:

k100 → D100
k100 → D101
k100 → D102
k100 → D103

CML: Inverted data transfer

◆ Overview

The CML instruction inverts the bit pattern in S and transfers the resulting data to D.

CML S D			Inverted data transfer	Applicable model: H3U	
S	Inverted data source	Data to be inverted, or address of the word element that stores the data		16-bit instruction (5 steps) CML: Continuous execution CMLP: Pulse execution	32-bit instruction (9 steps) DCML: Continuous execution DCMLP: Pulse execution
D	Transfer destination	Address of the word element that stores the inverted data arriving at a destination			

◆ Operands

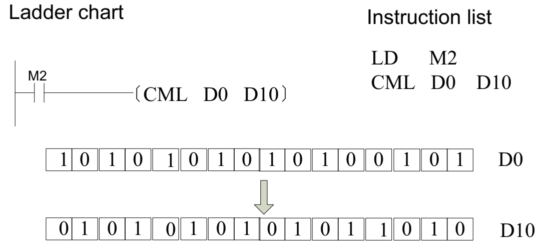
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

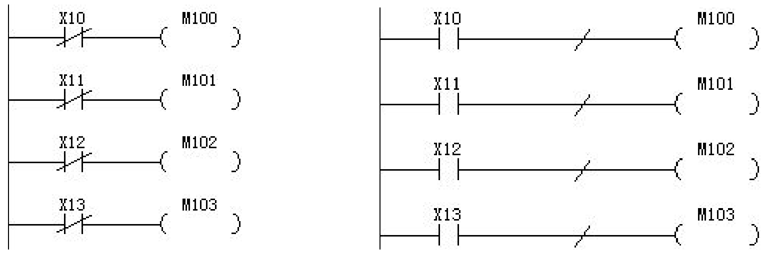
◆ Function

The CML instruction requires contact driving and has two operands. It inverts the bit pattern of the BIN value in S and copies the resulting data to D. When the number of bits in D is less than 16, the inverted value in S is transferred to D by aligning from lower bits. When the 32-bit instruction DCML is executed, S and D use the adjacent variable units with high-numbered addresses for the operation. For example, the statement [DCML D1 D5] gets the result /D1 -> D5; /D2 -> D6.

Example 1:

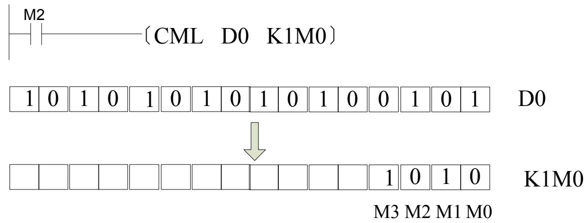


Example 2:



The preceding two programs can be executed by the CML instruction as follows:

Example 3:



ZPUSH: Index register batch storage

◆ Overview

The ZPUSH instruction copies the current values of index registers V0–V7 and Z0–Z7 to elements for temporary storage.

Use the ZPOP instruction to recover the temporarily stored values.

ZPUSH D			Index register batch storage	Applicable model: H3U	
D	Destination address	Start number of elements that temporarily store the current values of index registers V0–V7 and Z0–Z7 D: Number of batch storing times D+1 to D+16 x Number of batch storing times: Position of batch data storage		16-bit instruction (3 steps) ZPUSH: Continuous execution ZPUSHP: Pulse execution	

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

4

◆ Function

The data of index registers Z0–Z7 and V0–V7 is copied in batches to elements from head address [D]. The number of batch storing times is incremented by 1 on every execution of the instruction.

Use the ZPOP instruction to read the data copied by the ZPUSH instruction. The two instructions are used in pair.

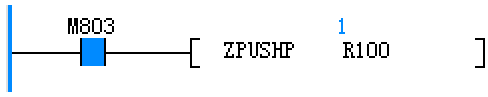
The structure of data stored in batches from head address [D] is as follows:

[D+0]	Number of batch storing times
[D+1]	Z0
[D+2]	V0
[D+3]	Z1
[D+4]	V1
[D+5]	Z2
[D+6]	V2
⋮	⋮
[D+15]	Z7
[D+16]	V7
[D+17]	Z0
[D+18]	Z1
⋮	⋮
[D+31]	Z7
[D+32]	V7
[D+33]	Z0
[D+34]	V0
⋮	⋮

An operation error occurs in the following conditions. The ZPUSH instruction is not executed. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6705 is returned when the operand of the ZPUSH instruction is out of range.
- Error 6706 is returned when the number of batch storing times is less than 0.

◆ Application



- The data of V and Z is as follows:

Window

Element Name	data type	display form:	curre
Z0	16-bit int	Dec	1
V0	16-bit int	Dec	2
Z1	16-bit int	Dec	3
V1	16-bit int	Dec	4
Z2	16-bit int	Dec	5
V2	16-bit int	Dec	6
Z3	16-bit int	Dec	7
V3	16-bit int	Dec	8
Z4	16-bit int	Dec	9
V4	16-bit int	Dec	10
Z5	16-bit int	Dec	11
V5	16-bit int	Dec	12
Z6	16-bit int	Dec	13
V6	16-bit int	Dec	14
Z7	16-bit int	Dec	15
V7	16-bit int	Dec	16

The result of an execution is as follows. R100 indicates the number of batch storing times, and R101 to R116 store the values of index registers.

Window

Element Name	data type	display form:	current value
R100	16-bit int	Dec	1
R101	16-bit int	Dec	2
R102	16-bit int	Dec	3
R103	16-bit int	Dec	4
R104	16-bit int	Dec	5
R105	16-bit int	Dec	6
R106	16-bit int	Dec	7
R107	16-bit int	Dec	8
R108	16-bit int	Dec	9
R109	16-bit int	Dec	10
R110	16-bit int	Dec	11
R111	16-bit int	Dec	12
	16-bit int	Dec	

4

ZPOP: Index register batch recovery

◆ Overview

The ZPOP instruction recovers the content of index registers V0–V7 and Z0–Z7 that is copied by the ZPUSH instruction for temporary storage.

ZPOP D			Index register batch recovery	Applicable model: H3U
D	Destination address	Head address of elements that temporarily store the content of index registers V0–V7 and Z0–Z7 copied by the ZPUSH (FNC 102) instruction in batches	16-bit instruction (3 steps) ZPOP: Continuous execution ZPOPP: Pulse execution	

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

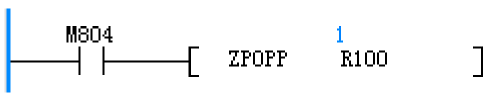
- The content that the ZPUSH instruction has copied to [D] is recovered to the corresponding index registers. The number of batch storing times is decremented by 1 on every execution of the ZPOP instruction.
- Use the ZPUSH instruction to copy the data of index registers in batches for temporary storage.

An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when the operand of the ZPOP instruction is out of range.

2. Error 6706 is returned when the number of batch storing times is 0 or negative.

◆ Application



- Before execution of the instruction

Window

Element Name	data type	display form:	current value
R100	16-bit int	Dec	1
R101	16-bit int	Dec	1
R102	16-bit int	Dec	2
R103	16-bit int	Dec	3
R104	16-bit int	Dec	4
R105	16-bit int	Dec	5
R106	16-bit int	Dec	6
R107	16-bit int	Dec	7
R108	16-bit int	Dec	8
R109	16-bit int	Dec	9
R110	16-bit int	Dec	10
R111	16-bit int	Dec	11
R112	16-bit int	Dec	12
R113	16-bit int	Dec	13
R114	16-bit int	Dec	14
R115	16-bit int	Dec	15
R116	16-bit int	Dec	16

4

- After the ZPOP instruction is executed, the data in R101 to R116 is copied to the corresponding index registers, and the value 1 in R100 changes to 0.

Window

Element Name	data type	display form:	current value
	16-bit int	Dec	
Z0	16-bit int	Dec	1
Z1	16-bit int	Dec	3
Z2	16-bit int	Dec	5
Z3	16-bit int	Dec	7
Z4	16-bit int	Dec	9
Z5	16-bit int	Dec	11
Z6	16-bit int	Dec	13
Z7	16-bit int	Dec	15
V0	16-bit int	Dec	2
V1	16-bit int	Dec	4
V2	16-bit int	Dec	6
V3	16-bit int	Dec	8
V4	16-bit int	Dec	10
V5	16-bit int	Dec	12
V6	16-bit int	Dec	14
V7	16-bit int	Dec	16

4.5.3 Table Operation

Table operation	ZRST	Full data reset
	SORT	Data sorting
	SORT2	Data sorting 2
	SER	Data search
	FDEL	Deletion of data from a table
	FINS	Insertion of data to a table
	POP	Last-in data read
	RAMP	Ramp instruction

ZRST: Full data reset

◆ Overview

The ZRST instruction resets data in batches.

ZRST D1 D2			Full data reset	Applicable model: H3U	
D1	Batch reset head address	Head address of elements whose data will be reset in batches		16-bit instruction (5 steps) ZRST: Continuous execution ZRSTP: Pulse execution	
D2	Batch reset end address	End address of elements whose data will be reset in batches			

4

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The values of all variables between D1 and D2 are cleared. D1 and D2 can be specified as word variables or Y, M, and S bit variables.

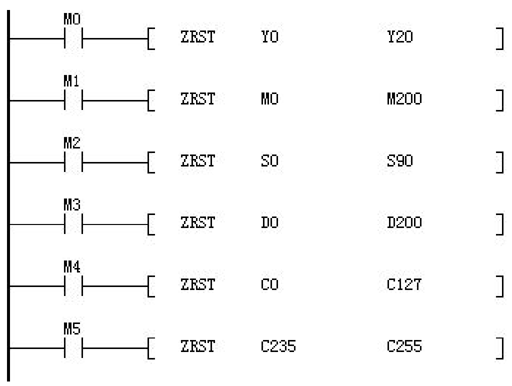
Requirements

D1 and D2 must be of the same element type.

D1 cannot be greater than D2. If they are the same, only the data in the designated element is reset.

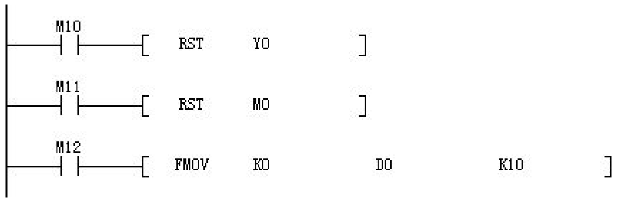
The ZRST instruction is a 16-bit instruction, but 32-bit counters can be designated for D1 and D2. Ensure that they use counters of the same bit type.

◆ Application



◆ Additional information

The bit elements Y, M, and S and the word elements T, C, and D can also use the RST instruction for data reset. The word elements T, C, and D and the bit registers KnY, KnM, and KnS can also use the FMOV instruction for multi-point data clearance. Example:



SORT: Data sorting

4

◆ Overview

When driving conditions are met, the SORT instruction sorts the data in the nth column of a table with m1 rows and m2 columns from head address S in ascending order. The sorting result is stored in a table from head address D.

SORT S m1 m2 D n		Data sorting	Applicable model: H3U	
S	Table 1 head address	Head address of word elements that store the data of table 1 (which occupies m1 x m2 points)	16-bit instruction (11 steps) SORT: Continuous execution	
m1	Table row count	Number of table rows, or address of the word element that stores the data		
m2	Table column count	Number of table columns, or address of the word element that stores the data		
D	Table 2 head address	Head address of word elements that store the data of table 2 (which occupies m1 x m2 points)		
n	Sorted column number	Number of the column whose data will be sorted; value range: 1 to m2		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The parameters in the nth column of a range of arrays which occupy an area of m1 (rows) x m2 (columns) from head address S are sorted. The result is stored in a variable area from head address D.

S is the start unit for the first variable in the first row (or record).

m1 is the number of array rows (or records).

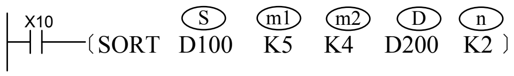
m2 is the number of array columns, or the number of columns in every record.

D is the start unit for storing the sorted data. The number of subsequent occupied variable units is the same as the number of array variables before the sorting.

n is the number of the array column whose data will be sorted.

The value of n ranges from 1 to m2.

◆ Application



When X10 = ON, a sorting operation is performed. After the SORT instruction is executed, M8029 is set to ON

To re-initiate sorting, switch X10 to OFF again.

Equivalent table and data example of the preceding instruction:

S, m2, m1

Row Number \ Column Number	1 Student Number	2 Language	3 Mathematics	4 Physics
1	D 100 1	D 105 85	D 110 78	D 115 83
2	D 101 2	D 106 82	D 111 91	D 116 81
3	D 102 3	D 107 77	D 112 89	D 117 88
4	D 103 4	D 108 90	D 113 81	D 118 75
5	D 104 5	D 109 87	D 114 95	D 119 77

Table data sorted based on (n) = K2 specified by the instruction:

D, n = K2

Row Number \ Column Number	1 Student Number	2 Language	3 Mathematics	4 Physics
1	D200 3	D205 77	D210 89	D215 88
2	D201 2	D206 82	D211 91	D216 81
3	D202 1	D207 85	D212 78	D217 83
4	D203 5	D208 87	D213 95	D218 77
5	D204 4	D209 90	D214 81	D219 75

Table data sorted based on (n) = K4 specified by the instruction:

D, n = K4

Row Number \ Column Number	1 Student Number	2 Language	3 Mathematics	4 Physics
1	D200 4	D205 90	D210 81	D215 75
2	D201 5	D206 87	D211 95	D216 77
3	D202 2	D207 82	D212 91	D217 81
4	D203 1	D208 85	D213 78	D218 83
5	D204 3	D209 77	D214 89	D219 88

SORT2: Data sorting 2

◆ Overview

The SORT2 instruction sorts the data of the designated column in ascending or descending order by row.

Data is stored consecutively by row for easy addition of row data.

SORT2 S m1 m2 D n			Data sorting 2	Applicable model: H3U	
S	Source address	Start number of elements that store a data table (which occupies m1 x m2 points)		16-bit instruction (11 steps) SORT2: Continuous execution	32-bit instruction (21 steps) DSORT2: Continuous execution
m1	Row count	Number of rows; value range: 1 to 32			
m2	Column count	Number of columns; value range: 1 to 6			
D	Destination address	Start number of elements that store the operation result (which occupies m1 x m2 points)			
n	Target column	Number of the column whose data is sorted; value range: 1 to m2			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The data of the nth column of a data table from head address [S] which occupies m1 x m2 points is sorted in ascending or descending order by row. The result is stored in a data table from head address [D] which occupies m1 x m2 points.

The ON/OFF state of M8165 determines the sorting order. When M8165 = ON, sorting is in descending order; when M8165 = OFF, sorting is in ascending order.

Data sorting starts when the instruction flow is active. Sorting is completed after the number of scans reaches m1. Then the execution complete flag M8029 is set to ON.

The following shows the sorting process of a data table which occupies 3 x 3 points.

4

- Before sorting

Row Number / Column Number	1	2	3
1	S	S+1	S+2
	1	2	8
2	S+3	S+4	S+5
	2	6	7
3	S+6	S+7	S+8
	3	4	3

- Table data after sorted in ascending order based on the second column

Row Number / Column Number	1	2	3
1	D	D+1	D+2
	1	2	8
2	D+3	D+4	D+5
	3	4	3
3	D+6	D+7	D+8
	2	6	7

◆ **Note:**

- The operands cannot be modified when the SORT2 instruction is executed.
- Switch the flow from OFF to ON before you execute the instruction for the second time.
- Keep the operands and data unchanged during execution.
- The contents of S and D can overlap completely or be staggered, but they cannot overlap partially.
- The 32-bit instruction is used in the same way as the 16-bit instruction. The operands occupy two 16-bit elements.

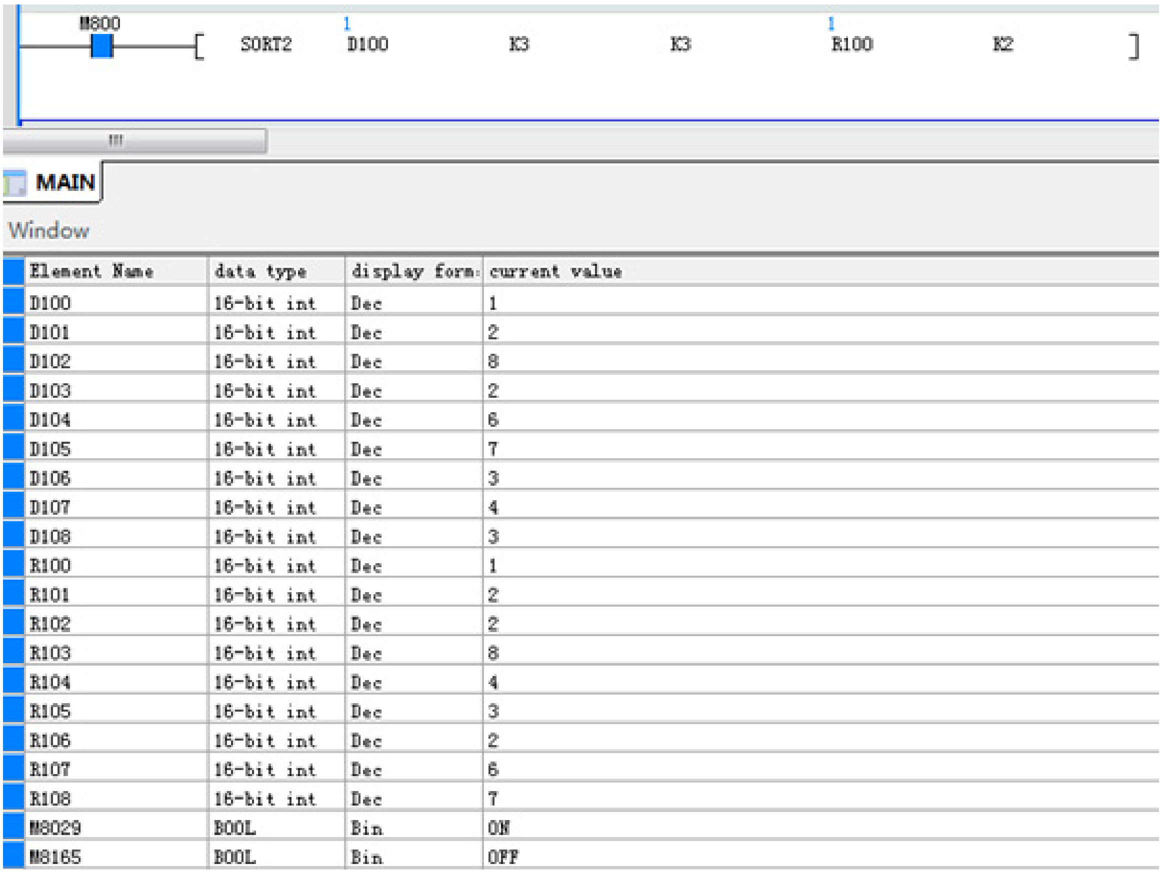
An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error K6705 is returned when the value in S or D is out of range.

2. Error K6706 is returned when the value of m1, m2, or n is out of range.

◆ Application

Data from head address D100 is sorted in ascending order based on the data of the second column. The result is stored in elements from head address R100.



Element Name	data type	display form	current value
D100	16-bit int	Dec	1
D101	16-bit int	Dec	2
D102	16-bit int	Dec	8
D103	16-bit int	Dec	2
D104	16-bit int	Dec	6
D105	16-bit int	Dec	7
D106	16-bit int	Dec	3
D107	16-bit int	Dec	4
D108	16-bit int	Dec	3
R100	16-bit int	Dec	1
R101	16-bit int	Dec	2
R102	16-bit int	Dec	2
R103	16-bit int	Dec	8
R104	16-bit int	Dec	4
R105	16-bit int	Dec	3
R106	16-bit int	Dec	2
R107	16-bit int	Dec	6
R108	16-bit int	Dec	7
M8029	BOOL	Bin	ON
M8165	BOOL	Bin	OFF

SER: Data search

◆ Overview

When driving conditions are met, the SER instruction searches k data entries from head address S1 to find the address of the data compliant with the condition set in D2. The result is stored in five consecutive registers from head address D.

SER S1 S2 D n			Data search	Applicable model: H3U	
S1	Search head address	Head address of the data to be searched (the search object is k consecutive registers)		16-bit instruction (9 steps) SER: Continuous execution SERP: Pulse execution	32-bit instruction (17 steps) DSER: Continuous execution DSERP: Pulse execution
S2	Compared data	Compared data, or address of the word element that stores the data			
D	Search result storage head address	Head address of word elements that store the search result			
n	Searched data count	Number of searched data entries			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

A defined data stack is searched to find the units with the same data as the compared data as well as the maximum and minimum values.

S1 is the head address of the searched data stack.

S2 is the data to be searched for.

D is the head address of search result storage.

n is the length of the searched data area. In 16-bit operation, the value range is 1 to 256; in 32-bit operation, the value range is 1 to 128.

In 32-bit operation, S1, S2, and D point to 32-bit variables, and n is calculated based on 32-bit variable width.

◆ Application

X20 (SER D10 D0 D80 K10)
(S1: D10, S2: D0, D: D80, n: K10)

S1	Searched Data	S2	Element Number	Parameter Condition
D 10	(D 10)=K100	Compared data (D0) =K100	0	Equal
D 11	(D 11)=K123		1	
D 12	(D 12)=K100		2	Equal
D 13	(D 13)=K98		3	
D 14	(D 14)=K111		4	
D 15	(D 15)=K66		5	Minimum
D 16	(D 16)=K100		6	Equal
D 17	(D 17)=K100		7	Equal
D 18	(D 18)=K210		8	Maximum
D 19	(D 19)=K88		9	

(n: D10 to D19)

Search result

D	Parameter	Definition
D80	4	Number of equal parameters
D81	0	ID of the first equal parameter
D82	7	ID of the last equal parameter
D83	5	ID of the minimum parameter
D84	8	ID of the maximum parameter

Usage:

Comparison is performed only when X20 = ON in the instruction flow. Signed numbers are compared algebraically, for example, –8 < +2.

When there are multiple minimum or maximum values, the element with the greatest number is displayed.

The search result is stored in five consecutive units from head address D. If no equal data exists, the values in D80 to D82 in the preceding example are all 0s.

FDEL: Deletion of data from a table

◆ Overview

The FDEL instruction deletes any data from a table.

FDEL S D n			Deletion of data from a table	Applicable model: H3U
S	Deleted data	Number of the element that stores the data to be deleted	16-bit instruction (7 steps) FDEL: Continuous execution FDELP: Pulse execution	
D	Data table information	Start number of elements that store a data table D: Number of stored data entries D+1: Start position of a data table		
n	Position of deletion	Position in a table at which data is deleted		

4

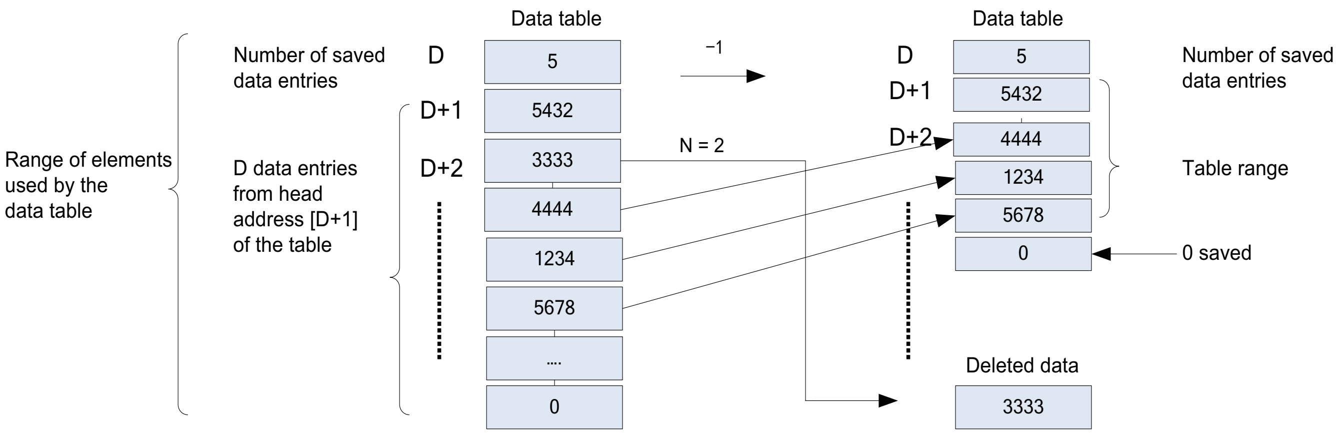
◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

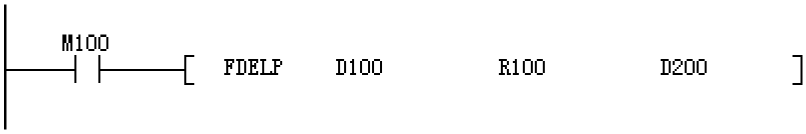
The nth data entry in a table from head address [D+1] is deleted. The deleted data is stored in [S]. The (n+1) th data entry and subsequent ones move forward in [D+1], and D (which indicates the number of stored data entries) is decremented by 1.



An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when the number of stored data entries is out of range.
2. Error 6706 is returned when n > [D].
3. Error 6706 is returned when n ≤ 0.
4. Error 6706 is returned when [D] ≤ 0.

◆ Application



- Before execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Dec	0
R100	16-bit int	Dec	5
R101	16-bit int	Dec	1111
R102	16-bit int	Dec	2222
R103	16-bit int	Dec	3333
R104	16-bit int	Dec	4444
R105	16-bit int	Dec	5555
R106	16-bit int	Dec	0
D200	16-bit int	Dec	3

- After execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Dec	3333
R100	16-bit int	Dec	4
R101	16-bit int	Dec	1111
R102	16-bit int	Dec	2222
R103	16-bit int	Dec	3333
R104	16-bit int	Dec	4444
R105	16-bit int	Dec	5555
R106	16-bit int	Dec	0
D200	16-bit int	Dec	3

FINS: Insertion of data to a table

◆ Overview

The FINS instruction inserts data at any position in a table.

FINS S D n			Insertion of data to a table	Applicable model: H3U	
S	Inserted data	Number of the element that stores the data to be inserted		7-bit instruction (5 steps) FINS: Continuous execution FINSP: Pulse execution	
D	Data table information	Start number of elements that store a data table D: Number of stored data entries D+1: Start position of a data table			
n	Position of insertion	Position in a table at which data is inserted			

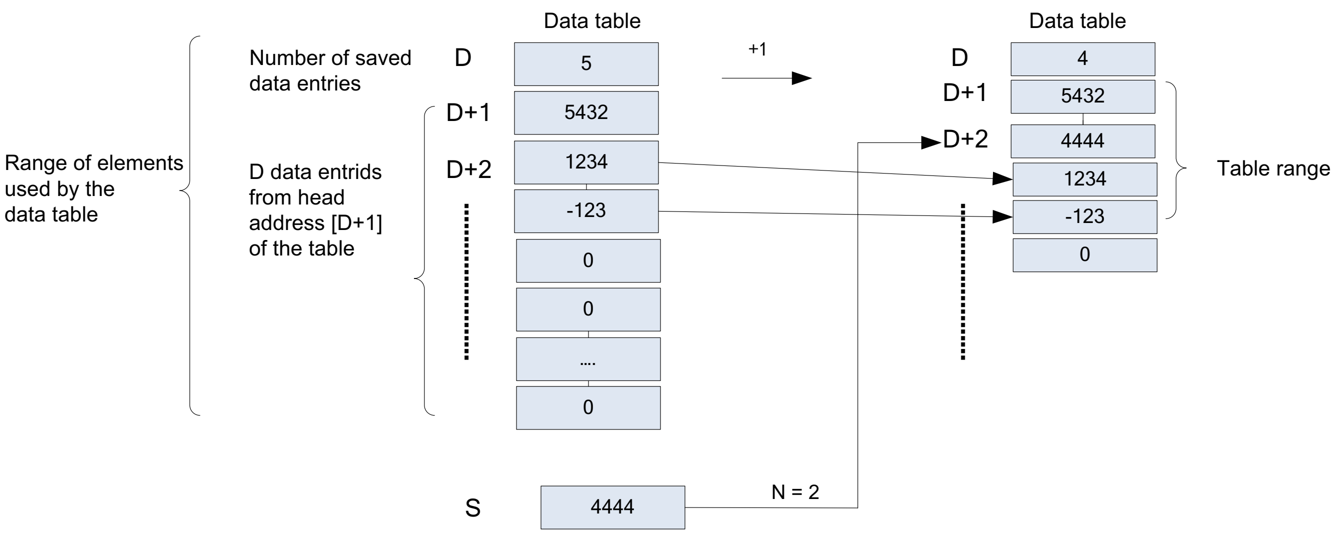
◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

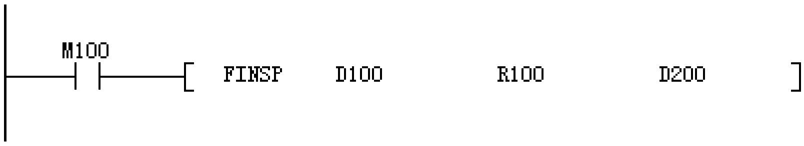
The data stored in [S] is inserted at the nth data entry position in a table from head address [D+1]. The original nth data entry and subsequent ones move backward, and [D] (which indicates the number of stored data entries) is incremented by 1.



An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when the number of stored data entries is out of range.
2. Error 6705 is returned when the table after data insertion is out of range.
3. Error 6706 is returned when n > [D].
4. Error 6706 is returned when n ≤ 0.
5. Error 6706 is returned when [D] < 0.

◆ Application



● Before execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Dec	4
D101	16-bit int	Dec	1111
D102	16-bit int	Dec	2222
D103	16-bit int	Dec	3333
D104	16-bit int	Dec	4444
D105	16-bit int	Dec	5555
D106	16-bit int	Dec	6666
D107	16-bit int	Dec	4
	16-bit int	Dec	
R100	16-bit int	Dec	0

● After execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Dec	3
D101	16-bit int	Dec	1111
D102	16-bit int	Dec	2222
D103	16-bit int	Dec	3333
D104	16-bit int	Dec	4444
D105	16-bit int	Dec	5555
D106	16-bit int	Dec	6666
D107	16-bit int	Dec	4
	16-bit int	Dec	
R100	16-bit int	Dec	4444

POP: Last-in data read

◆ Overview

The POP instruction reads the data that is last written by the SFWR instruction which is used for shifted write operation with first in last out (FILO) control.

POP S D n		Last-in data read	Applicable model: H3U	
S	Data to be read	Start number of elements that store first-in data (including pointer data) S: Pointer data (number of stored data entries) S+1: Data area	7-bit instruction (7 steps) POP: Continuous execution POPP: Pulse execution	
D	Stored result	Number of the element that stores the last-out data		
n	Data count	Number of stored data points (Because pointer data is also included, set n to a value plus 1. The value range is 2 ≤ n ≤ 512.)		

◆ Operands

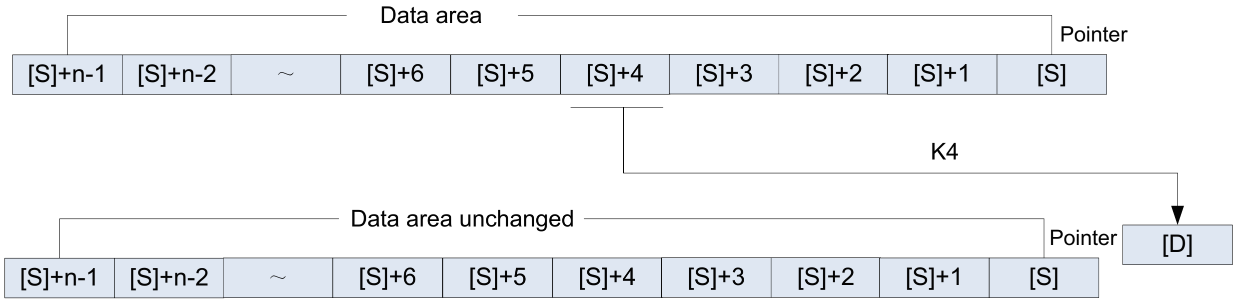
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

For the word elements in the range [S–S+n-1], the POP instruction reads the values in elements from head address S as well as the offset pointer (pointer data) in [S]. The result is stored in [D]. The pointer in [S] is decremented by 1. The value of n ranges from 2 to 512.

	Description
S	Pointer data (number of stored data entries)
[S]+1	Data area (First-in data written by the SFWR instruction)
[S]+2	
[S]+3	
–	
[S]+n-3	
[S]+n-2	
[S]+n-1	



When the pointer in [S] is 0, the zero flag M8020 is set to ON and the POP instruction is not executed.

Use a comparison instruction to check whether the pointer is in the range 1 ≤ [S] ≤ (n – 1) before executing the POP instruction.

When the pointer in [S] is 1, 0 is written to [S] and the zero flag M8020 is set to ON.

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

Error 6706 is returned when [S] > n – 1.

Error 6706 is returned when [S] < 0.

◆ Application

- Before execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Dec	4
D101	16-bit int	Dec	1111
D102	16-bit int	Dec	2222
D103	16-bit int	Dec	3333
D104	16-bit int	Dec	4444
D105	16-bit int	Dec	5555
D106	16-bit int	Dec	6666
D107	16-bit int	Dec	4
	16-bit int	Dec	
R100	16-bit int	Dec	0

- After execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Dec	3
D101	16-bit int	Dec	1111
D102	16-bit int	Dec	2222
D103	16-bit int	Dec	3333
D104	16-bit int	Dec	4444
D105	16-bit int	Dec	5555
D106	16-bit int	Dec	6666
D107	16-bit int	Dec	4
	16-bit int	Dec	
R100	16-bit int	Dec	4444

RAMP: Ramp instruction

◆ Overview

When driving conditions are met, the RAMP instruction changes the value in D linearly from S1 to S2 after a number (indicated by S3) of scan cycles are completed.

RAMP S1 S2 D n		Ramp instruction	Applicable model: H3U	
S1	Start value	Address of the word element that stores the ramp start value	16-bit instruction (9 steps) RAMP: Continuous execution	
S2	End value	Address of the word element that stores the ramp end value		
D	Current value	Address of the word element that stores the current ramp value		
n	Cycle count	Number of scan cycles required to complete a ramp change; value range: 1 to 32,767		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Linear interpolation is performed between two given data records within the designated time range. Process values are output in sequence based on the scan execution time until the end value of the range is reached.

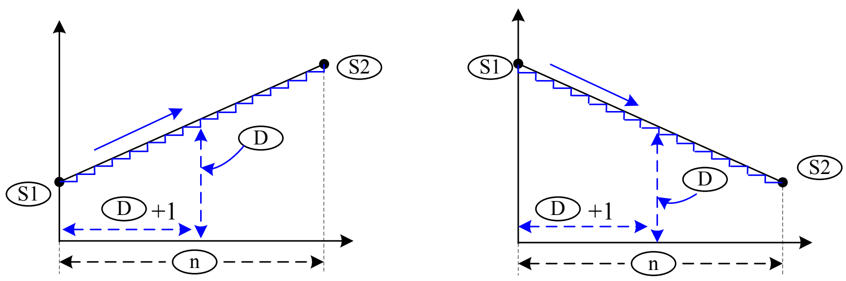
S1 is the start value unit of a ramp signal.

S2 is the end value unit of a ramp signal.

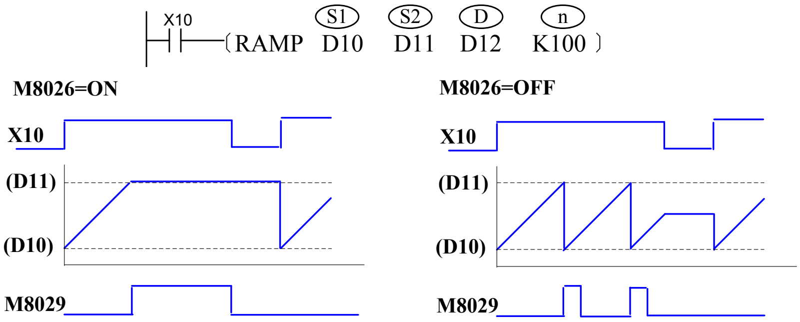
D stores the process values of a linear interpolation signal. The interpolation times counter is stored in D+1.

n is the number of scan execution times required to complete interpolation. The value range is 1 to 32,767. Because interpolation output is performed during the normal main loop, set program execution to fixed scan mode to ensure linear output. (For details, see the description of M8039 and D8039.)

Interpolation adopts integer calculation, and the fractional part is discarded. The function of the instruction is shown as follows:



The RAMP instruction has two modes, which mode to use is selected by the M8026 flag. M8029 is set to ON when interpolation is completed. The execution is shown as follows:



4.5.4 Data Rotation and Shift

Data rotation and shift	ROR	Rotation right
	ROL	Rotation left
	RCR	Rotation right with carry
	RCL	Rotation left with carry
	SFTR	Bit shift right
	SFTL	Bit shift left
	WSFR	Word shift right
	WSFL	Word shift left
	SFWR	FIFO data write
	SFRD	FIFO data read
	SFR	16-bit data shift right with carry by n bits
	SFL	16-bit data shift left with carry by n bits

ROR: Rotation right

◆ Overview

When driving conditions are met, the bit pattern in D is rotated K bit places to the right on every execution of the ROR instruction. The lower bits that are rotated out of D fill the higher bits of D.

ROR D n		Rotation right	Applicable model: H3U	
D	Operand to be rotated	Address of the word element that stores the data	16-bit instruction (5 steps) ROR: Continuous execution RORP: Pulse execution	32-bit instruction (9 steps) DROR: Continuous execution DRORP: Pulse execution
n	Number of bit places rotated on every execution	Value range: 1 ≤ n ≤16 (16-bit operation); 1 ≤ n ≤ 32 (32-bit operation)		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

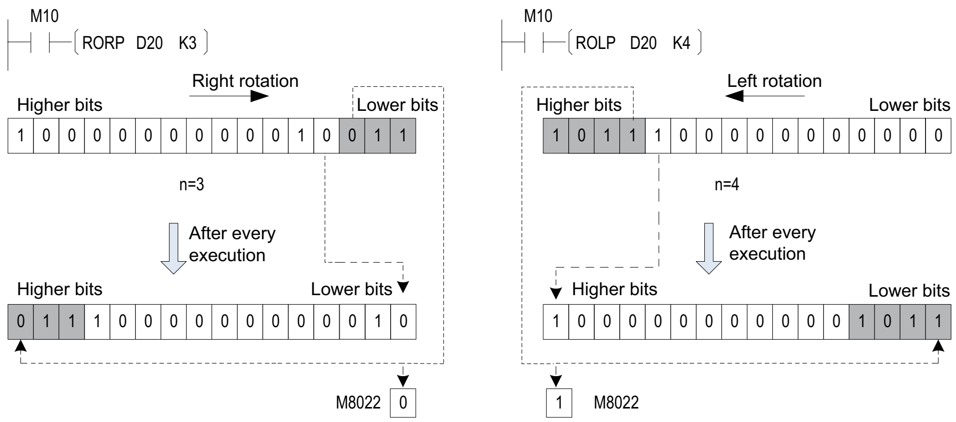
Note: The elements in gray background are supported.

◆ Function

The bit pattern in D is rotated n bit places to the right. The instruction of the pulse execution type is generally used. When the 32-bit instruction is executed, the register variable occupies two consecutive units.

When KnY, KnM, and KnS are specified in D, only K4 (16-bit operation) and K8 (32-bit operation) are valid.

Example:



ROL: Rotation left

◆ Overview

When driving conditions are met, the bit pattern in D is rotated K bit places to the left on every execution of the ROL instruction. The higher bits that are rotated out of D fill the lower bits of D.

ROL D n		Rotation left	Applicable model: H3U	
D	Operand to be rotated	Address of the word element that stores the data	16-bit instruction (5 steps) ROL: Continuous execution ROLP: Pulse execution	32-bit instruction (9 steps) DROL: Continuous execution DROLP: Pulse execution
n	Number of bit places rotated on every execution	Value range: 1 ≤ n ≤ 16 (16-bit operation); 1 ≤ n ≤ 32 (32-bit operation)		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

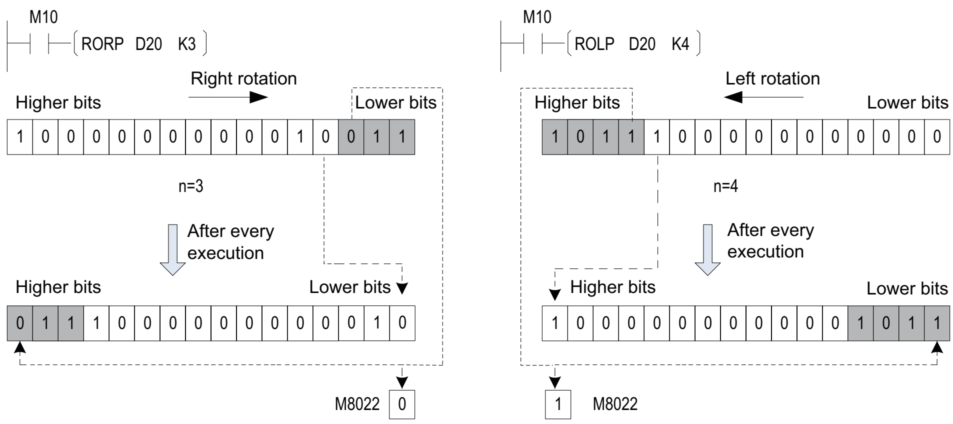
Note: The elements in gray background are supported.

◆ Function

The bit pattern in D is rotated n bit places to the left. The instruction of the pulse execution type is generally used. When the 32-bit instruction is executed, the register variable occupies two consecutive units.

When KnY, KnM, and KnS are specified in D, only K4 (16-bit operation) and K8 (32-bit operation) are valid. The status of the last bit rotated is copied to the carry flag.

Example:



RCR: Rotation right with carry

◆ Overview

When driving conditions are met, the bit pattern in D with the carry flag M8022 is rotated K bit places to the right on every execution of the RCR instruction. The lower bits with the carry flag that are rotated out of D fill the higher bits of D.

RCR D n			Rotation right with carry	Applicable model: H3U	
D	Operand to be rotated	Address of the word element that stores the data		16-bit instruction (5 steps) RCR: Continuous execution RCRP: Pulse execution	32-bit instruction (9 steps) DRCR: Continuous execution DRCRP: Pulse execution
n	Number of bit places rotated on every execution	Value range: 1 ≤ n ≤ 16 (16-bit operation);1 ≤ n ≤ 32 (32-bit operation)			

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

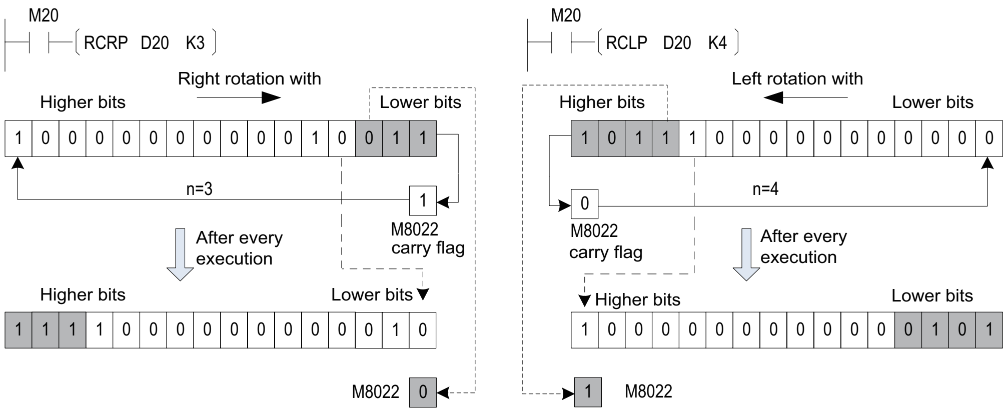
The bit pattern in D with the carry flag M8022 is rotated n bit places to the right.

The instruction of the pulse execution type is generally used.

When the 32-bit instruction is executed, the register variable occupies two consecutive units.

When KnY, KnM, and KnS are specified in D, only K4 (16-bit operation) and K8 (32-bit operation) are valid.

Example:



RCL: Rotation left with carry

◆ Overview

When driving conditions are met, the bit pattern in D with the carry flag M8022 is rotated K bit places to the left on every execution of the ROL instruction. The higher bits with the carry flag that are rotated out of D fill the lower bits of D.

RCL D n		Rotation left with carry	Applicable model: H3U	
D	Operand to be rotated	Address of the word element that stores the data	16-bit instruction (5 steps) RCL: Continuous execution RCLP: Pulse execution	32-bit instruction (9 steps) DRCL: Continuous execution DRCLP: Pulse execution
n	Number of bit places rotated on every execution	Value range: 1 ≤ n ≤ 16 (16-bit operation); 1 ≤ n ≤ 32 (32-bit operation)		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

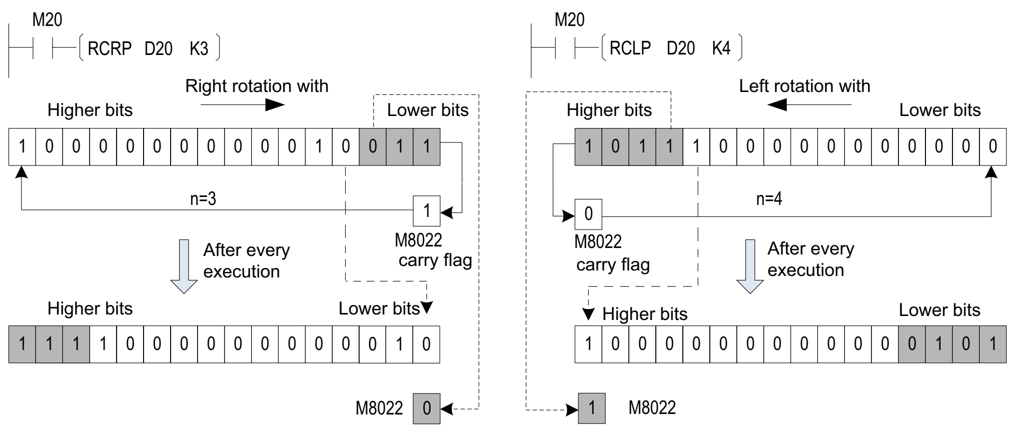
◆ Function

The bit pattern in D with the carry flag M8022 is rotated n bit places to the left. The instruction of the pulse execution type is generally used.

When the 32-bit instruction is executed, the register variable occupies two consecutive units.

When KnY, KnM, and KnS are specified in D, only K4 (16-bit operation) and K8 (32-bit operation) are valid.

Example:



SFTR: Bit shift right

◆ Overview

When driving conditions are met, the SFTR instruction shifts a combination of bit elements with a length of K1 from head address D to the right by K2 bit places, to accommodate a combination of bit elements with a length of K2 from head address S that fill the higher bits. The K2 lower bits that are moved out are discarded. The original values in the bit element combination S remain unchanged.

SFTR S D n1 n2		Bit shift right	Applicable model: H3U	
S	Bit element head address	Head address of shifted bit elements	16-bit instruction (9 steps) SFTR: Continuous execution SFTRP: Pulse execution	
D	Incoming bit head address	Head address of incoming bit elements		
n1	Incoming bit count	Number of incoming bit elements		
n2	Bit element count	Number of shifted bit elements		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

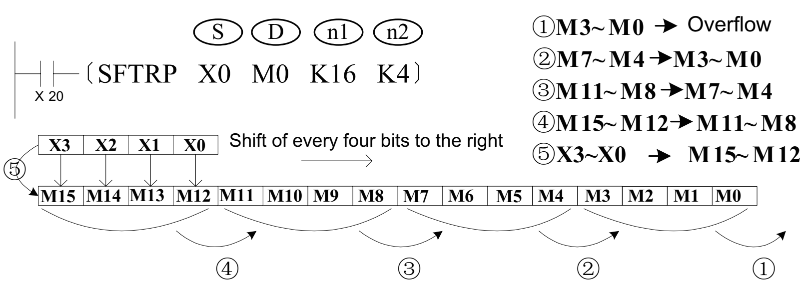
Note: The elements in gray background are supported.

◆ Function

n1 bit variables from head address D are shifted n2 bit places to the right, to accommodate n2 bit variables from head address S that fill the higher bits.

The instruction of the pulse execution type is generally used.

Example:



SFTL: Bit shift left

◆ Overview

When driving conditions are met, the SFTL instruction shifts a combination of bit elements with a length of K1 from head address D to the left by K2 bit places, to accommodate a combination of bit elements with a length of K2 from head address S that fill the lower bits. The K2 higher bits that are moved out are discarded. The original values in the bit element combination S remain unchanged.

SFTL S D n1 n2		Bit shift left	Applicable model: H3U	
S	Bit element head address	Head address of shifted bit elements	16-bit instruction (9 steps) SFTL: Continuous execution SFTLP: Pulse execution	
D	Incoming bit head address	Head address of incoming bit elements		
n1	Incoming bit count	Number of incoming bit elements		
n2	Bit element count	Number of shifted bit elements		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

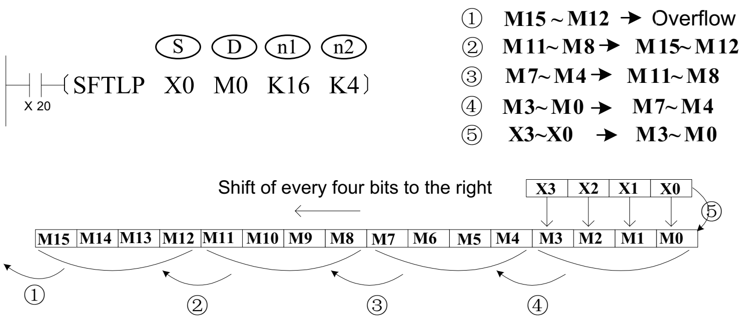
◆ Function

n1 bit variables from head address D are shifted n2 bit places to the left, to accommodate n2 bit variables from head address S that fill the lower bits.

The instruction of the pulse execution type is generally used.

The variable types applicable to the operands of SFTR and SFTL are as follows:

Example:



WSFR: Word shift right

◆ Overview

When driving conditions are met, the WSFR instruction shifts a combination of word elements with a length of K1 from head address D to the right by K2 word places, to accommodate a combination of word elements with a length of K2 from head address S that fill the higher words. The K2 lower words that are moved out are discarded. The original values in the word element combination S remain unchanged.

WSFR S D n1 n2		Word shift right	Applicable model: H3U	
S	Word element head address	Head address of shifted word elements	16-bit instruction (9 steps) WSFR: Continuous execution WSFRP: Pulse execution	
D	Incoming word head address	Head address of incoming word elements		
n1	Incoming word count	Number of incoming word elements		
n2	Word element count	Number of shifted word elements		

◆ Operands

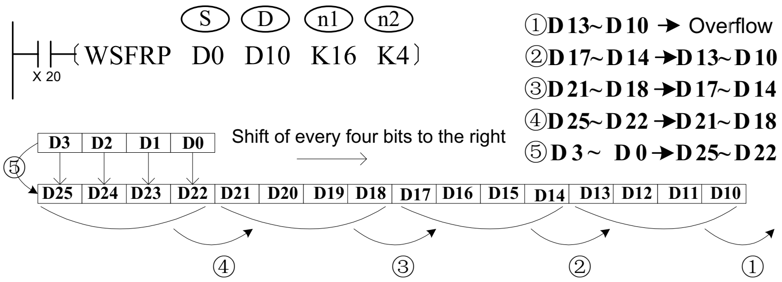
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

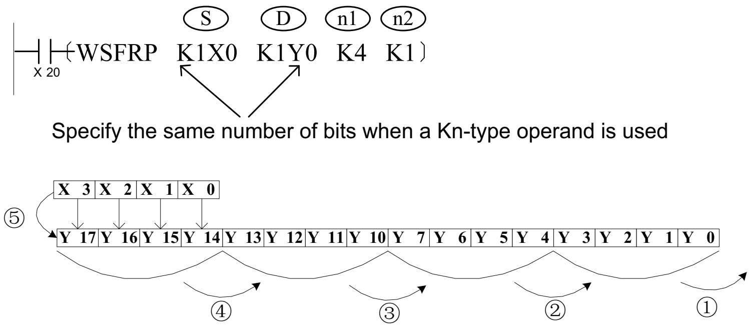
Note: The elements in gray background are supported.

◆ Function

n1 word variables from head address D are shifted n2 word places to the right, to accommodate n2 word variables from head address S that fill the higher words. The instruction of the pulse execution type is generally used.

Example:





Bits are shifted to the right in the following sequence after every scan:

1: Y3-Y0 → Carry flag

2: Y17-Y14 → Y13-Y10

3: Y13-Y10 → Y7-Y4

4: Y7-Y4 → Y3-Y0

5: X3-X00 → Y17-Y14 completed

WSFL: Word shift left

◆ Overview

When driving conditions are met, the WSFL instruction shifts a combination of word elements with a length of K1 from head address D to the left by K2 word places, to accommodate a combination of word elements with a length of K2 from head address S that fill the lower words. The K2 higher words that are moved out are discarded. The original values in the word element combination S remain unchanged.

WSFL S D n1 n2			Word shift left	Applicable model: H3U
S	Word element head address	Head address of shifted word elements	16-bit instruction (9 steps) WSFL: Continuous execution WSFLP: Pulse execution	
D	Incoming word head address	Head address of incoming word elements		
n1	Incoming word count	Number of incoming word elements		
n2	Word element count	Number of shifted word elements		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

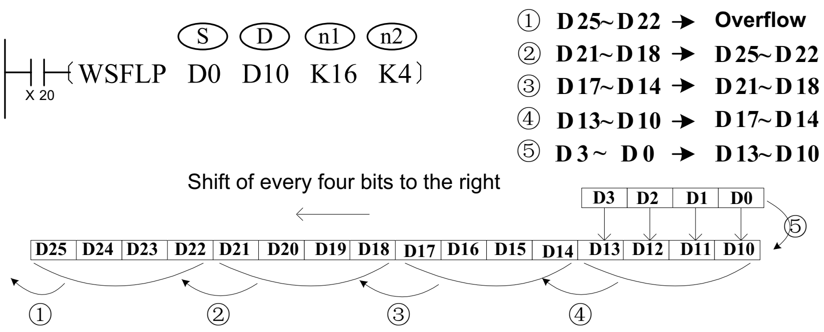
◆ Function

n1 word variables from head address D are shifted n2 word places to the left, to accommodate n2 word variables from head address S that fill the lower words.

The instruction of the pulse execution type is generally used.

4

Example:



SFWR: FIFO data write

◆ Overview

When driving conditions are met, the SFWR instruction writes the current content of S to a data register with a length of n from head address D+1. The value of the pointer D is incremented by 1 each time a data entry is written to the database.

SFWR S D n			FIFO data write	Applicable model: H3U
S	Data source	Data to be written, or address of the word element that stores the data	16-bit instruction (7 steps) SFWR: Continuous execution SFWRP: Pulse execution	
D	Data area head address	Head address of word elements that store the data in a data area		
n	Data area length	Length of a data area, including pointer data		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

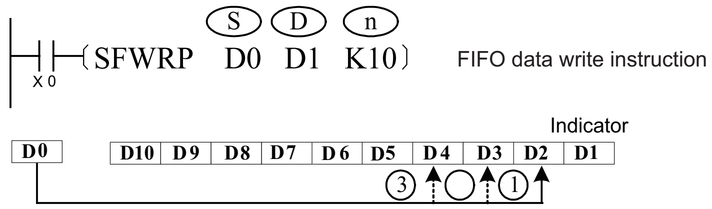
Note: The elements in gray background are supported.

◆ Function

The content of S is written to a first in first out (FIFO) queue with a length of n from head address D. The operand with the first number stores a pointer. When the instruction is executed, the pointer is incremented by 1 and then the content of the source operand (S) is written to the FIFO queue (D). The position of insertion into the queue is specified by the pointer.

The instruction of the pulse execution type is generally used.

Example:



When X0 = 1, a data entry in D0 is written to D2 and the value in D1 changes to 1. When X0 switches from OFF to ON, another data entry in D0 is written to D3 and the value in D1 changes to 2, and so on. If the value in D1 exceeds the value of n minus 1, insertion into the FIFO queue is stopped. The carry flag M8022 is set to 1 to identify this situation.

SFRD: FIFO data read

◆ Overview

When driving conditions are met, the SFRD instruction reads the data from head address S+1 in a data register with a length of n. The read data is written to the destination register D.

SFRD S D n		FIFO data read	Applicable model: H3U	
S	Data area head address	Head address of word elements that store the data in a data area	16-bit instruction (7 steps) SFRD: Continuous execution SFRDP: Pulse execution	
D	Read data	Address for storing read data		
n	Data area length	Length of a data area		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

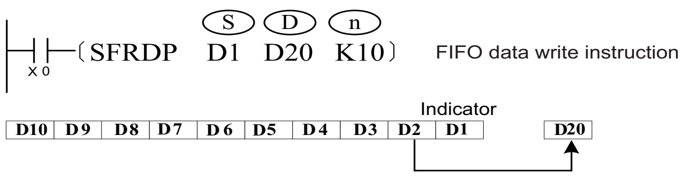
◆ Function

The first piece of data in the FIFO queue (S) is read. The data within the queue is subsequently moved one word to the right to fill the read area, and the queue pointer is decremented by 1. The read data is written to D. The operand with the first number stores a pointer. When the instruction is executed, the pointer is decremented by 1 and then the content of the source operand specified by S is written to the FIFO queue specified by D. The position of insertion into the queue is specified by the pointer. If the pointer is 0, the preceding operation is not performed and the zero flag M8020 is set to 1 to identify this situation.

4

The instruction of the pulse execution type is generally used.

Example:



When X0 switches from OFF to ON, this instruction acts in the following sequence (the content of D10 unchanged):
1. The content of D2 is read and transferred to D20.
2. D10 to D3 are shifted one register to the right.
3. The value of pointer D1 is decremented by 1.

SFR: 16-bit data shift right with carry by n bits

◆ Overview

The SFR instruction shifts the 16 bits of a word element to the right by n bit places.

SFR S n			16-bit data shift right with carry by n bits	Applicable model: H3U	
S	Word to be shifted	Number of the element that stores the data to be shifted		16-bit instruction (5 steps) SFR: Continuous execution SFRP: Pulse execution	
n	Shift times	Number of shift times; value range: 0 ≤ n ≤ 15			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V and Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V and Z	Modification	K	H	E

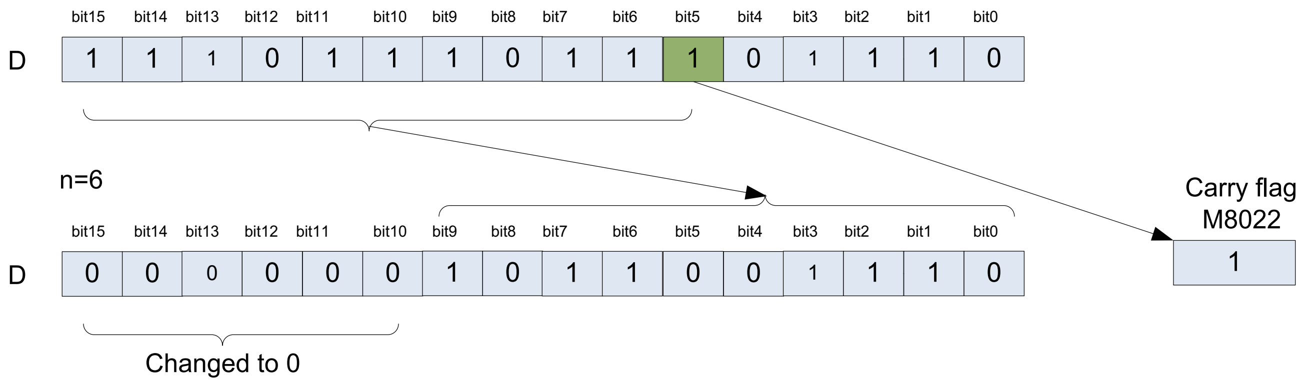
Note: The elements in gray background are supported.

◆ Function

The 16 bits of the word with the carry flag in the element [D] are shifted n bit places to the right. n is a number in the range 0 to 15.

When n ≥ 16, bits are shifted by a number of bit places calculated by n%16 (remainder). For example, when n = 20, bits are shifted four bit places (20%16 = 4) to the right.

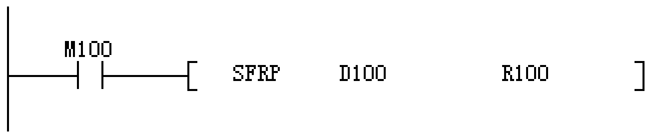
The 1/0 state of the (n – 1)th bit in [D] is written to the carry flag M8022. The n bits starting from the highest bit in [D] are filled with 0s.



An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

Error 6706 is returned when n < 0.

◆ Application



● Before execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Hex	0xAAAA
	16-bit int	Dec	
R100	16-bit int	Hex	0x8
	16-bit int	Dec	
M8022	BOOL	Bin	OFF
	16-bit int	Dec	

● After execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Hex	0xAA
	16-bit int	Dec	
R100	16-bit int	Hex	0x8
	16-bit int	Dec	
M8022	BOOL	Bin	ON
	16-bit int	Dec	

SFL: 16-bit data shift left with carry by n bits

◆ Overview

The SFL instruction shifts the 16 bits of a word element to the left by n bit places.

SFL S n			16-bit data shift left with carry by n bits	Applicable model: H3U	
S	Word to be shifted	Number of the element that stores the data to be shifted		16-bit instruction (5 steps) SFL: Continuous execution SFLP: Pulse execution	
n	Shift times	Number of shift times; value range: 0 ≤ n ≤ 15			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

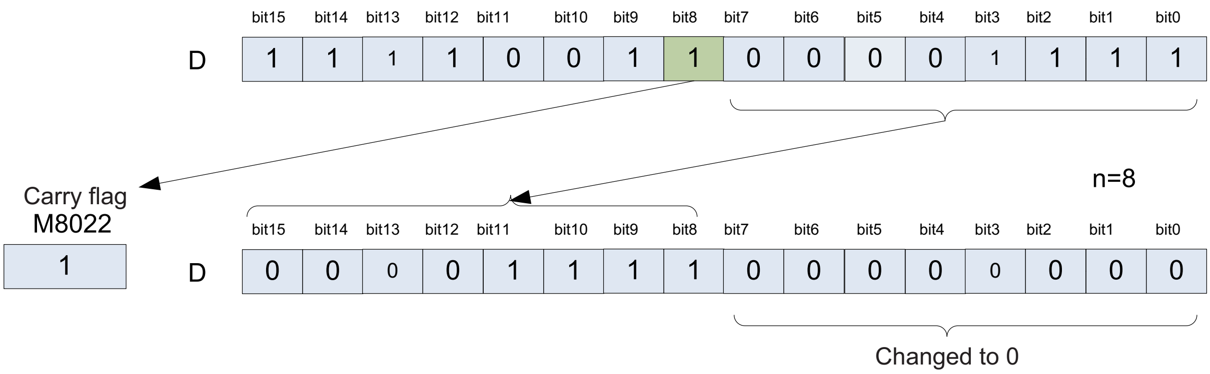
The 16 bits of the word with the carry flag in the element [D] are shifted n bit places to the left.

n is a number in the range 0 to 15.

When n ≥ 16, bits are shifted by a number of bit places calculated by n%16 (remainder). For example, when n = 20, bits are shifted four bit places (20%16 = 4) to the left.

The 1/0 state of the nth bit in [D] is written to the carry flag M8022.

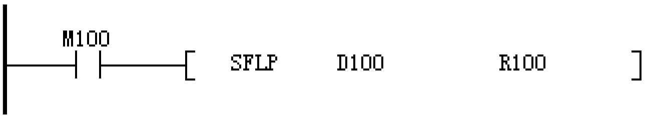
The n bits starting from the lowest bit in [D] are filled with 0s.



An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

Error 6706 is returned when n < 0.

◆ Application



● Before execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x5555
	16-bit int	Dec	
R100	16-bit int	Hex	0x8
	16-bit int	Dec	
M8022	BOOL	Bin	OFF
	16-bit int	Dec	

● After execution of the instruction

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x5500
	16-bit int	Dec	
R100	16-bit int	Hex	0x8
	16-bit int	Dec	
M8022	BOOL	Bin	ON
	16-bit int	Dec	

4

4.5.5 Other Data Processing

SWAP: Higher and lower byte swap

◆ Overview

The SWAP instruction exchanges the higher and lower bytes of the variable in S.

SWAP S			Higher and lower byte swap	Applicable model: H3U	
S	Operand	Unit that stores the data whose higher and lower bytes will be exchanged		16-bit instruction (3 steps) SWAP: Continuous execution SWAPP: Pulse execution	32-bit instruction (5 steps) DSWAP: Continuous execution DSWAPP: Pulse execution

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The higher and lower bytes of the variable in S are exchanged.

In 16-bit operation, the higher eight bits and the lower eight bits are exchanged.

In 32-bit operation, the higher eight bits and the lower eight bits of two registers are exchanged.

The instruction of the pulse execution type is generally used. If the continuous execution type is used, calculation is performed on every program scan.

Example:

In the figure on the left, the values of higher and lower eight bits in D20 are exchanged.
In the figure on the right, the values of higher and lower eight bits in D20 are exchanged, and the values of higher and lower eight bits in D21 are exchanged.

BON: ON bit check

◆ Overview

When driving conditions are met, the BON instruction checks the status of the Kth bit of the binary data in S. The result is used for D status control.

BON S D n		ON bit check	Applicable model: H3U	
S	Source data	Data, or address of the word element that stores the data	16-bit instruction (7 steps) BON: Continuous execution BONP: Pulse execution	32-bit instruction (13 steps) DBON: Continuous execution DBONP: Pulse execution
D	Controlled bit	Controlled bit element		
n	Designated bit	Designated bit in S; value range: 1 ≤ n ≤ 15 (16-bit operation); 1 ≤ n ≤ 31 (32-bit operation)		

◆ Operands

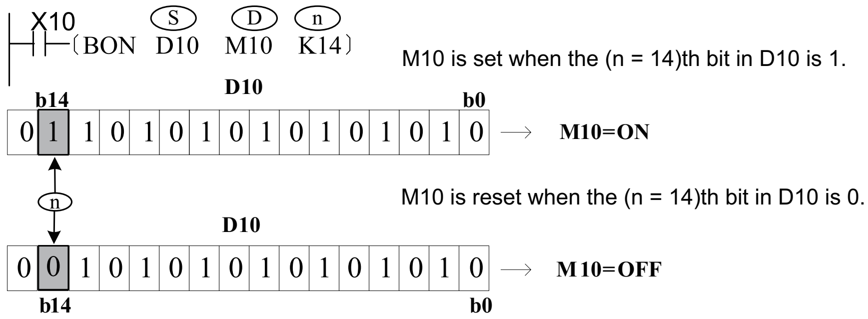
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ **Function**

The status of the nth bit in S is checked. The result is stored in D.

Example:



M10 remains in the current state when X10 switches from ON to OFF.

4

SUM: Total number of ON bits

◆ **Overview**

When driving conditions are met, the SUM instruction counts the ON bits (with a value of 1) of the binary data in S. The result is stored in D.

SUM S D			Total number of ON bits	Applicable model: H3U	
S	Data to be counted	Data to be counted, or address of the element that stores the data		16-bit instruction (5 steps) SUM: Continuous execution SUMP: Pulse execution	32-bit instruction (9 steps) DSUM: Continuous execution DSUMP: Pulse execution
D	Counting result	Address of the element that stores the data			

◆ **Operands**

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

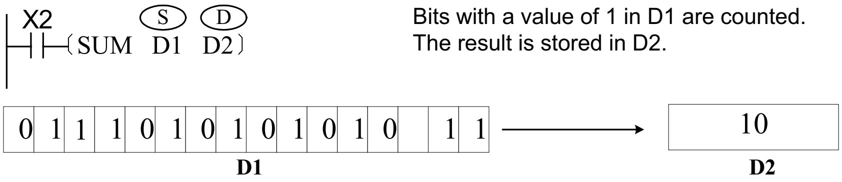
◆ **Function**

ON bits (with a value of 1) of the BIN data in S are counted. The result is stored in D.

When DSUM and DSUMP are executed, the number of bits with a value of 1 among the 32 bits in (S+1, S) is written to D. All bits in D+1 are set to 0.

If all bits in S are 0, the zero flag M8020 is set to ON.

Example:



RND: Random number generation

◆ Overview

The RND instruction generates random numbers.

RND D			Random number generation	Applicable model: H3U
D	Destination address	Address of the element that stores random numbers	16-bit instruction (3 steps) RND: Continuous execution RNDP: Pulse execution	

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

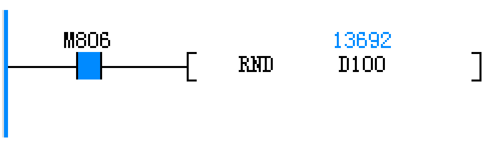
◆ Function

Pseudo random numbers are generated in the range 0 to 32,767. The generated numbers are stored as random numbers in [D].

Upon STOP-to-RUN switching, write only one non-zero value (in the range –2,147,483,648 to +2,147,483,647) as the initial value to (D8311, D8310).

◆ Application

Random numbers are generated and stored in D100.



XCH: Data exchange

◆ Overview

When driving conditions are met, the XCH instruction exchanges the data in S and D.

XCH S D			Data exchange	Applicable model: H3U
S	Data 1	Word element 1 that stores the data to be exchanged	16-bit instruction (5 steps) XCH: Continuous execution XCHP: Pulse execution	32-bit instruction (9 steps) DXCH: Continuous execution DXCHP: Pulse execution
D	Data 2	Word element 2 that stores the data to be exchanged		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The XCH instruction requires contact driving and has two operands. It exchanges the values in S and D.

Example 1:

Before execution
D110=K180
D120=K200

After execution
D110=K200
D120=K180

Example 2:

Before execution
D110=K180
D111=K150
D120=K200
D121=K100

After execution
D110=K200
D111=K100
D120=K180
D121=K150

When the special variable M8160 is set to 1 and the addresses of D and S are the same, the higher eight bits and the lower eight bits are exchanged in both 16- and 32-bit operations. The XCH instruction is equivalent to the SWAP instruction. The SWAP instruction is generally used.

ANS: Annunciator setting

◆ Overview

When driving conditions are met, the ANS instruction starts the timer in S. When the timer completes its cycle K, the selected annunciator flag (D) is set.

ANS S K D		Annunciator setting	Applicable model: H3U	
S	Timer T	Timer T for annunciator setting	16-bit instruction (7 steps) ANS: Continuous execution	
m	Timing cycle	Timing cycle of timer T		
D	Annunciator flag	Annunciator flag, in the range S900 to S999		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	**T**	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	**K**	**H**	E
D	X	Y	M	T	C	**S**	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

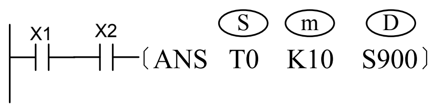
Note: The elements in gray background are supported.

◆ Function

The ANS instruction is used for annunciator driving.

The value in S ranges from T0 to T199, and the value in D ranges from S900 to S999.

Example:



If X1 and X2 are connected simultaneously for more than 1s, S900 is set. Even if X1 or X2 switches to OFF later, S900 remains active (but T0 is reset and its value changes to 0). If the connection duration is less than 1s, the timer is reset when X1 or X2 switches to OFF.

If M8049 (annunciator effectiveness) is preset to ON, the number of the lowest active (ON) annunciator in the range S900 to S999 is stored in D8049. When any annunciator in the same range is set ON, M8048 (annunciator action) is set to ON.

Note: The following table lists related elements.

Element	Name	Description
M8049	Annunciator effectiveness	When M8049 = ON, D8049 and M8049 take effect.
M8048	Annunciator action	When M8049 = ON and any annunciator in the range S900 to S999 is ON, M8048 is set to ON.
D8049	Number of the lowest active (ON) annunciator	Stores the number of the lowest active annunciator in the range S900 to S999.

ANR: Annunciator reset

◆ Overview

When driving conditions are met, the ANR instruction resets the annunciators in the range S900 to S999 in batches.

ANR	Annunciator reset	Applicable model: H3U	
No operand		16-bit instruction (1 step) ANR: Continuous execution ANRP: Pulse execution	

4.6 Matrix Instructions

Matrix operations	BK+	Matrix addition operation	-
	BK-	Matrix subtraction operation	
	MAND	Matrix AND operation	These instructions perform bit-based operation on a set of word elements. 32-bit operation is not supported. Pulse operation is supported.
	MOR	Matrix OR operation	
	MXOR	Matrix XOR operation	
	MXNR	Matrix XNR operation	
	MINV	Matrix inverse operation	
	MBC	Matrix bit status counting operation	
Matrix comparison	BKCMP=	Matrix equal-to comparison (S1 = S2)	-
	BKCMP>	Matrix greater-than comparison (S1 > S2)	
	BKCMP<	Matrix less-than comparison (S1 < S2)	
	BKCMP<>	Matrix not-equal-to comparison (S1 ≠ S2)	
	BKCMP<=	Matrix less-than-or-equal-to comparison (S1 ≤ S2)	
	BKCMP>=	Matrix greater-than-or-equal-to comparison (S1 ≥ S2)	
	MCMP	Matrix comparison operation	These instructions perform bit-based operation on a set of word elements. 32-bit operation is not supported. Pulse operation is supported.
Matrix read/ write	MBRD	Matrix bit read operation	
	MBWR	Matrix bit write operation	
Matrix rotation and shift	MBS	Matrix bit shift operation	
	MBR	Matrix bit rotation operation	

4.6.1 Matrix Operations

Matrix operations	BK+	Matrix addition operation
	BK-	Matrix subtraction operation
	MAND	Matrix AND operation
	MOR	Matrix OR operation
	MXOR	Matrix XOR operation
	MXNR	Matrix XNR operation
	MINV	Matrix inverse operation
	MBC	Matrix bit status counting operation

BK+: Matrix addition operation

◆ Overview

The BK+ instruction adds BIN numbers together in matrix format.

BK+ S1 S2 D n			Matrix addition operation	Applicable model: H3U
S1	Source address	Start number of elements that store the data subjected to an addition operation	16-bit instruction (9 steps) BK+: Continuous execution BK+P: Pulse execution	32-bit instruction (17 steps) DBK+: Continuous execution DBK+P: Pulse execution
S2	Source address	Constant subjected to an addition operation, or start number of elements that store the data subjected to an addition operation		
D	Destination address	Start number of elements that store the operation result		
n	Data count	Number of data entries in an operation		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The n data entries (16- or 32-bit) from head addresses [S1] and [S2] are added together. The result is stored in n units (16- or 32-bit) from head address [D].

[S1+0]	K1111		[S2+0]	K1111		[D+0]	K2222
[S1+1]	K1111		[S2+1]	K-2222		[D+1]	K-1111
...	...	+	...	...	=	...	...
[S1+n-2]	K1111		[S2+n-2]	K3333		[D+n-2]	K4444
[S1+n-1]	K1111		[S2+n-1]	K4444		[D+n-1]	K5555

A signed constant (16- or 32-bit) can be directly set in [S2].

[S1+0]	K1111				[D+0]	K3333
[S1+1]	K1111				[D+1]	K3333
...	...	+	K2222	=	...	...
[S1+n-2]	K1111				[D+n-2]	K3333
[S1+n-1]	K1111				[D+n-1]	K3333

4

An error is returned in the following conditions. The instruction is not executed. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when elements from head addresses [S1], [S2], and [D] are out of range.

2. Error 6705 is returned when [S1] or [S2] has overlapping elements with [D].

◆ Application

Element Name	data type	display format	current value
D100	16-bit int	Dec	1111
D101	16-bit int	Dec	1111
D102	16-bit int	Dec	1111
D103	16-bit int	Dec	1111
D104	16-bit int	Dec	1111
D105	16-bit int	Dec	0
D200	16-bit int	Dec	2222
D201	16-bit int	Dec	2222
D202	16-bit int	Dec	2222
D203	16-bit int	Dec	2222
D204	16-bit int	Dec	2222
D205	16-bit int	Dec	0
R100	16-bit int	Dec	3333
R101	16-bit int	Dec	3333
R102	16-bit int	Dec	2222
R103	16-bit int	Dec	3333
R104	16-bit int	Dec	3333
R105	16-bit int	Dec	0

BK-: Matrix subtraction operation

◆ Overview

The BK- instruction subtracts the BIN number stored at one source address from another in matrix format.

BK- S1 S2 D n			Matrix subtraction operation	Applicable model: H3U	
S1	Source address	Start number of elements that store the data subjected to a subtraction operation		16-bit instruction (9 steps) BK-: Continuous execution BK-P: Pulse execution	32-bit instruction (17 steps) DBK-: Continuous execution DBK-P: Pulse execution
S2	Source address	Constant subjected to a subtraction operation, or start number of elements that store the data subjected to a subtraction operation			
D	Destination address	Start number of elements that store the operation result			
n	Data count	Number of data entries in an operation			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The n data entries (16- or 32-bit) from head address [S2] are subtracted from the n data entries (16- or 32-bit) from head address [S1]. The result is stored in n units (16- or 32-bit) from head address [D].

[S1+0]	K1111
[S1+1]	K1111
...	...
[S1+n-2]	K1111
[S1+n-1]	K1111

—

[S2+0]	K1111
[S2+1]	K-2222
...	...
[S2+n-2]	K3333
[S2+n-1]	K4444

=

[D+0]	K0
[D+1]	K3333
...	...
[D+n-2]	K-2222
[D+n-1]	K-3333

A signed constant (16- or 32-bit) can be directly set in [S2].

[S1+0]	K1111
[S1+1]	K1111
...	...
[S1+n-2]	K1111
[S1+n-1]	K1111

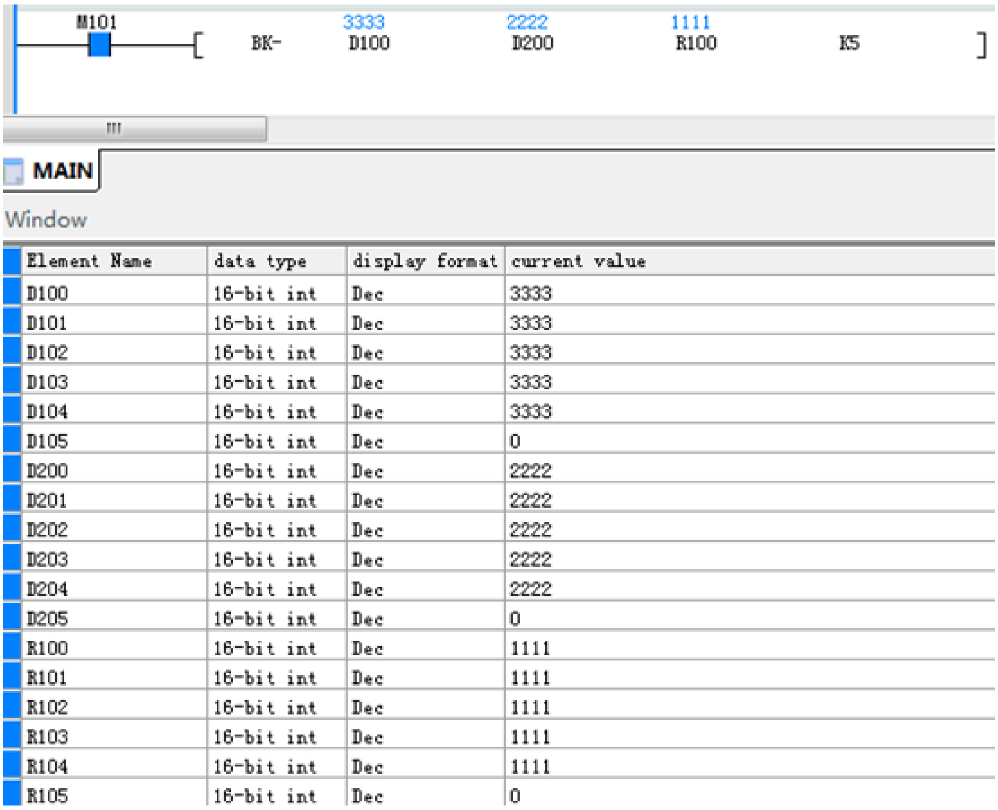
— K-2222 =

[D+0]	K3333
[D+1]	K3333
...	...
[D+n-2]	K3333
[D+n-1]	K3333

An error is returned in the following conditions. The instruction is not executed. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when elements from head addresses [S1] and [S2] are out of range.
2. Error 6705 is returned when [S1] and [S2] have overlapping elements with [D].

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Dec	3333
D101	16-bit int	Dec	3333
D102	16-bit int	Dec	3333
D103	16-bit int	Dec	3333
D104	16-bit int	Dec	3333
D105	16-bit int	Dec	0
D200	16-bit int	Dec	2222
D201	16-bit int	Dec	2222
D202	16-bit int	Dec	2222
D203	16-bit int	Dec	2222
D204	16-bit int	Dec	2222
D205	16-bit int	Dec	0
R100	16-bit int	Dec	1111
R101	16-bit int	Dec	1111
R102	16-bit int	Dec	1111
R103	16-bit int	Dec	1111
R104	16-bit int	Dec	1111
R105	16-bit int	Dec	0

MAND: Matrix AND operation

◆ Overview

The MAND instruction performs AND operation in matrix format.

<table>
<tr><th colspan="3">MAND S1 S2 D n</th><th>Matrix AND operation</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S1</td><td>Matrix 1</td><td colspan="2">Operand element 1 in an operation</td><td rowspan="4">16-bit instruction (9 steps)
MAND: Continuous execution
MANDP: Pulse execution</td><td rowspan="4"></td></tr>
<tr><td>S2</td><td>Matrix 2</td><td colspan="2">Operand element 2 in an operation</td></tr>
<tr><td>D</td><td>Operation result</td><td colspan="2">Start number of elements that store the operation result</td></tr>
<tr><td>n</td><td>Data count</td><td colspan="2">Number of data entries in an operation; value range: 1 to 256</td></tr>
</table>

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

An AND operation is performed on the bit patterns of the n bytes of data from head addresses [S1] and [S2]. The result is stored in elements from head address [D].

The result is 1 when the values of two bits are both 1; otherwise, the result is 0.

Assume that n = 4. The result of a matrix AND operation is as follows:

	bit15															bit0
[S1+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

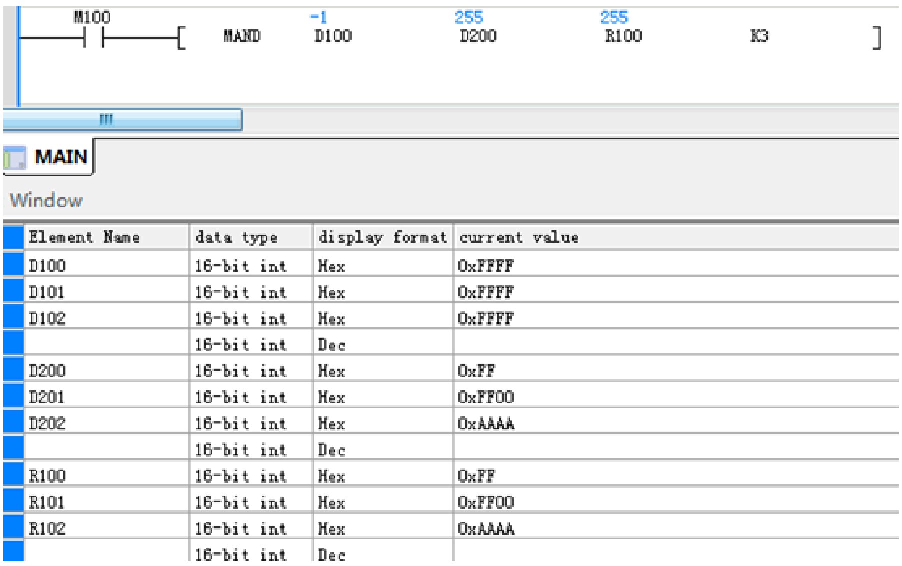
(bit XNR)

[S2+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1

↓

[D+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

◆ Application



4

MOR: Matrix OR operation

◆ Overview

The MOR instruction performs OR operation in matrix format.

MOR S1 S2 D n			Matrix OR operation	Applicable model: H3U
S1	Matrix 1	Operand element 1 in an operation	16-bit instruction (9 steps) MOR: Continuous execution MORP: Pulse execution	
S2	Matrix 2	Operand element 2 in an operation		
D	Operation result	Start number of elements that store the operation result		
n	Data count	Number of data entries in an operation; value range: 1 to 256		

4

◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

An OR operation is performed on the bit patterns of the n bytes of data from head addresses [S1] and [S2]. The result is stored in elements from head address [D].

The result is 1 when the value of any bit is 1; otherwise, the result is 0.

Assume that n = 4. The result of a matrix OR operation is as follows:

	bit15															bit0
[S1+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

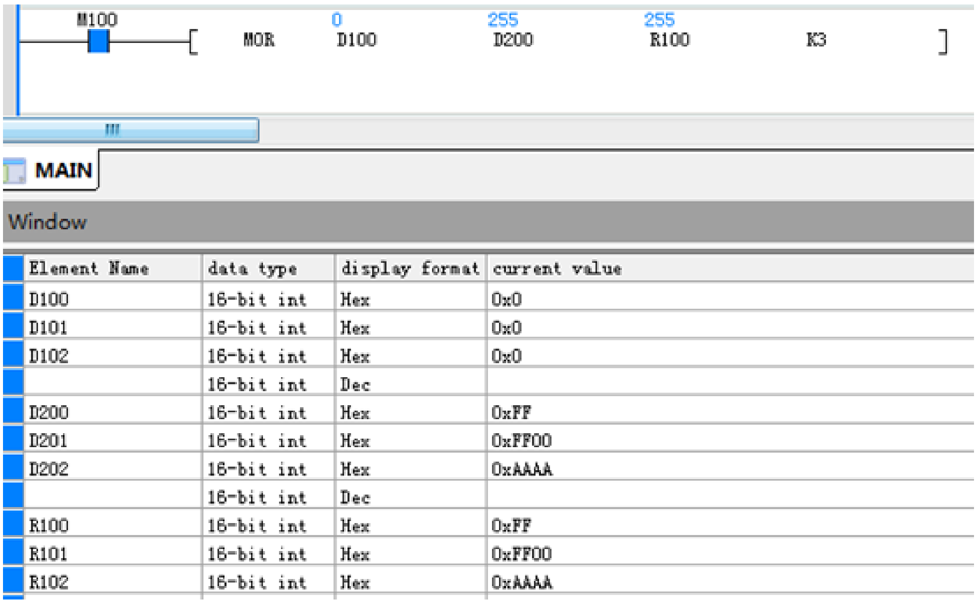
(bit OR)

[S2+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1

↓

[D+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1

◆ Application



MXOR: Matrix XOR operation

◆ Overview

The MXOR instruction performs XOR operation in matrix format.

MXOR S1 S2 D n		Matrix XOR operation	Applicable model: H3U	
S1	Matrix 1	Operand element 1 in an operation	16-bit instruction (9 steps) MXOR: Continuous execution MXORP: Pulse execution	
S2	Matrix 2	Operand element 2 in an operation		
D	Operation result	Start number of elements that store the operation result		
n	Data count	Number of data entries in an operation; value range: 1 to 256		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

An XOR operation is performed on the bit patterns of the n bytes of data from head addresses [S1] and [S2]. The result is stored in elements from head address [D].

The result is 1 when the values of two bits are different; otherwise, the result is 0.

Assume that n = 4. The result of a matrix XOR operation is as follows:

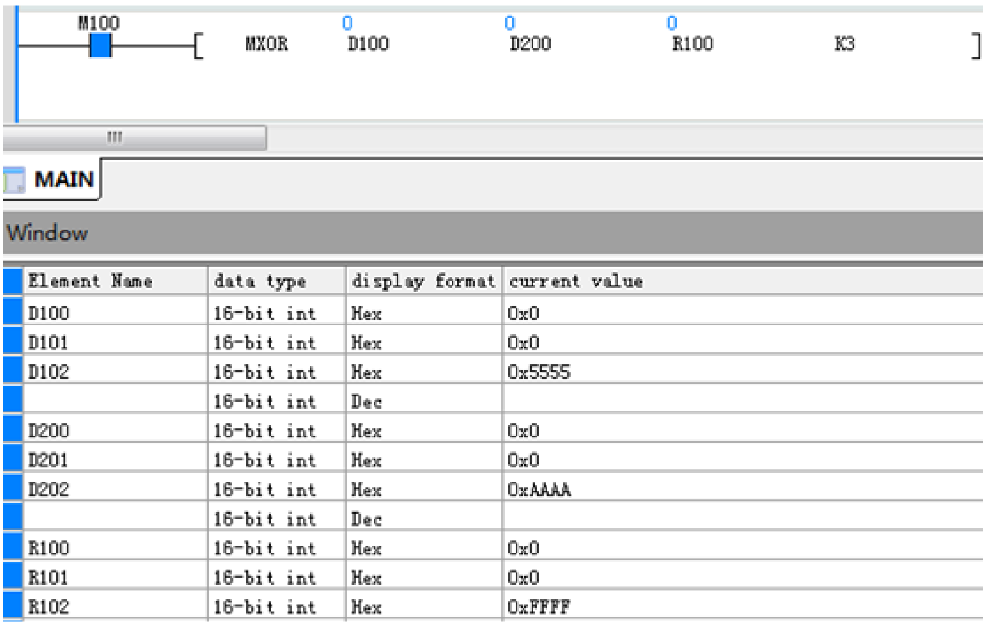
	bit15															bit0
[S1+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

(bit XOR)

[S2+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1

[D+0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[+1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[+2]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
[+3]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x0
D101	16-bit int	Hex	0x0
D102	16-bit int	Hex	0x5555
	16-bit int	Dec	
D200	16-bit int	Hex	0x0
D201	16-bit int	Hex	0x0
D202	16-bit int	Hex	0xAAAA
	16-bit int	Dec	
R100	16-bit int	Hex	0x0
R101	16-bit int	Hex	0x0
R102	16-bit int	Hex	0xFFFF

4

MXNR: Matrix XNR operation

◆ Overview

The MXNR instruction performs XNR operation in matrix format.

MXNR S1 S2 D n			Matrix XNR operation	Applicable model: H3U
S1	Matrix 1	Operand element 1 in an operation	16-bit instruction (9 steps) MXNR: Continuous execution MXNRP: Pulse execution	
S2	Matrix 2	Operand element 2 in an operation		
D	Operation result	Start number of elements that store the operation result		
n	Data count	Number of data entries in an operation; value range: 1 to 256		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

An XNR operation is performed on the bit patterns of the n bytes of data from head addresses [S1] and [S2]. The result is stored in elements from head address [D].

The result is 1 when the values of two bits are the same; otherwise, the result is 0.

Assume that n = 4. The result of a matrix XNR operation is as follows:

	bit15															bit0
[S1+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

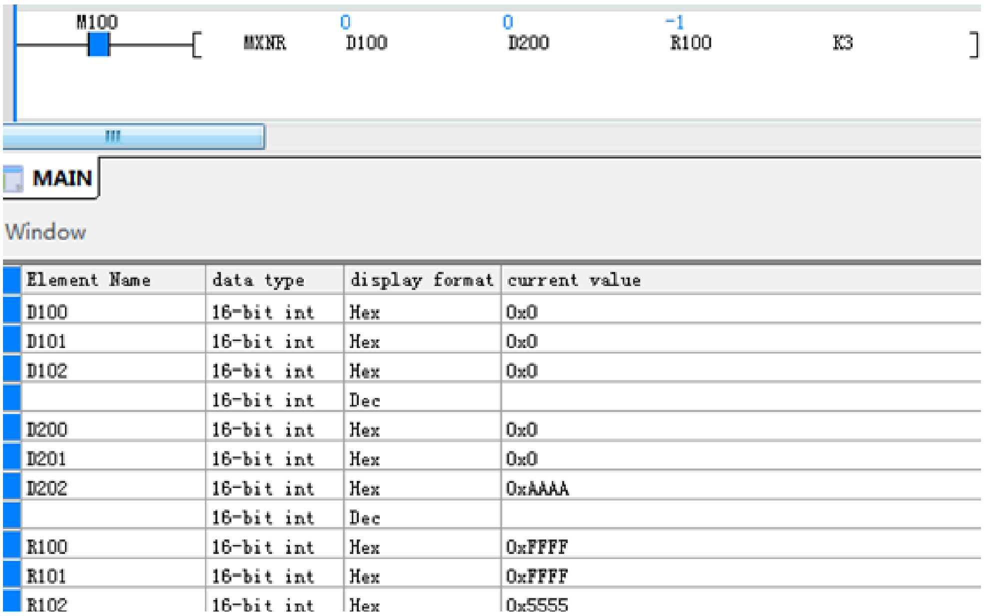
(bit XNR)

[S2+0]	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1
[+3]	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1

↓

[D+0]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0
[+1]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0
[+2]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0
[+3]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0

◆ **Application**



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x0
D101	16-bit int	Hex	0x0
D102	16-bit int	Hex	0x0
	16-bit int	Dec	
D200	16-bit int	Hex	0x0
D201	16-bit int	Hex	0x0
D202	16-bit int	Hex	0xAAAA
	16-bit int	Dec	
R100	16-bit int	Hex	0xFFFF
R101	16-bit int	Hex	0xFFFF
R102	16-bit int	Hex	0x5555

MINV: Matrix inverse operation

◆ **Overview**

The MINV instruction inverts the bit pattern of the designated matrix.

MINV S D n		**Matrix inverse operation**	**Applicable model:** H3U	
S	Matrix	Operand element in an operation	16-bit instruction (7 steps) MINV: Continuous execution MINVP: Pulse execution	
D	Operation result	Start number of elements that store the operation result		
n	Data count	Number of data entries in an operation; value range: 1 to 256		

◆ **Operands**

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

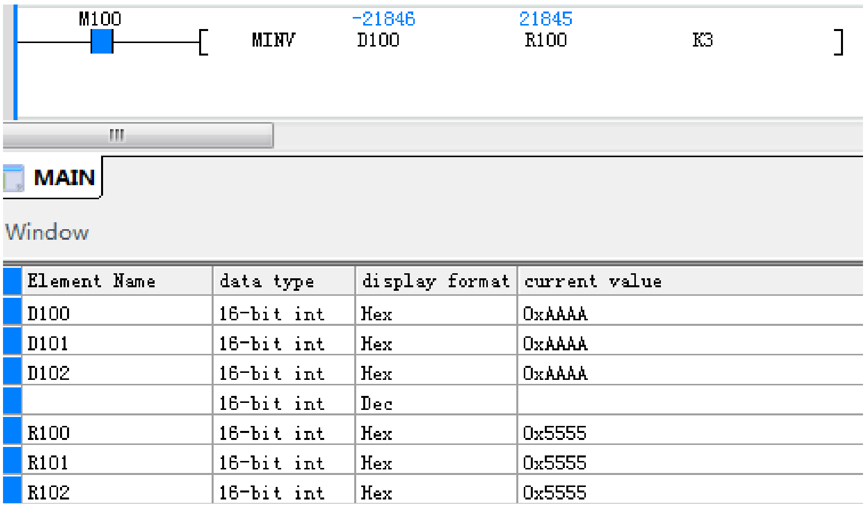
The bit pattern of the n bytes of data from head address [S] is inverted. The result is stored in elements from head address [D].

[S+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1

⇩ (bit inversion)

[D+0]	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	0
[+1]	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	0

◆ Application



MBC: Matrix bit status counting operation

◆ Overview

The MBC instruction counts bits in a matrix by status.

MBC S n D		Matrix bit status counting operation	Applicable model: H3U	
S	Matrix	Operand element in an operation	16-bit instruction (7 steps) MBC: Continuous execution MBCP: Pulse execution	
n	Data count	Number of data entries in an operation; value range: 1 to 256		
D	Operation result	Start number of elements that store the operation result		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Bits with a value of 0 or 1 are counted in a matrix composed of n 16-bit data entries from head address [S]. The result is stored in elements from head address [D].

When M8331 = ON, bits with a value of 1 are counted; otherwise, bits with value of 0 are counted.

M8332 is set to ON when the result is 0.

◆ Application

When M8331 = OFF and M100 = ON:

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x7777
D101	16-bit int	Hex	0x7777
D102	16-bit int	Hex	0x7777
	16-bit int	Dec	
M100	BOOL	Bin	ON
R100	16-bit int	Dec	12
M8331	BOOL	Bin	OFF

When M8331 = ON and M100 = ON:

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x7777
D101	16-bit int	Hex	0x7777
D102	16-bit int	Hex	0x7777
	16-bit int	Dec	
M100	BOOL	Bin	ON
D200	16-bit int	Dec	3
R100	16-bit int	Dec	36
M8331	BOOL	Bin	OFF

M8331 indicates whether bits with a value of 0 or 1 are counted. M8332 is set to ON when the counting result is 0.

4.6.2 Matrix Comparison

Matrix comparison	BKCMP=	Matrix equal-to comparison (S1 = S2)
	BKCMP>	Matrix greater-than comparison (S1 > S2)
	BKCMP<	Matrix less-than comparison (S1 < S2)
	BKCMP<>	Matrix not-equal-to comparison (S1 ≠ S2)
	BKCMP<=	Matrix less-than-or-equal-to comparison (S1 ≤ S2)
	BKCMP>=	Matrix greater-than-or-equal-to comparison (S1 ≥ S2)
	MCMP	Matrix comparison operation

BKCMP=, >, <, <>, <=, and >=: Matrix comparison

◆ Overview

These instructions perform matrix comparison based on comparison conditions.

<table>
<tr><th colspan="3">BKCMP# S1 S2 D n</th><th>Matrix comparison</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S1</td><td>Comparative value</td><td colspan="2">Comparative value, or number of the element that stores the comparative value</td><td rowspan="4">16-bit instruction (9 steps)
BKCMP#: Continuous execution
BKCMP#P: Pulse execution</td><td rowspan="4">32-bit instruction (17 steps)
DBKCMP#: Continuous execution
DBKCMP#P: Pulse execution</td></tr>
<tr><td>S2</td><td>Compared value</td><td colspan="2">Start number of elements that store the source data to be compared</td></tr>
<tr><td>D</td><td>Destination address</td><td colspan="2">Start number of elements that store the comparison result</td></tr>
<tr><td>n</td><td>Data count</td><td colspan="2">Number of data entries in an operation</td></tr>
</table>

Note: The # symbol can be =, >, <, <>, <=, or >=.

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

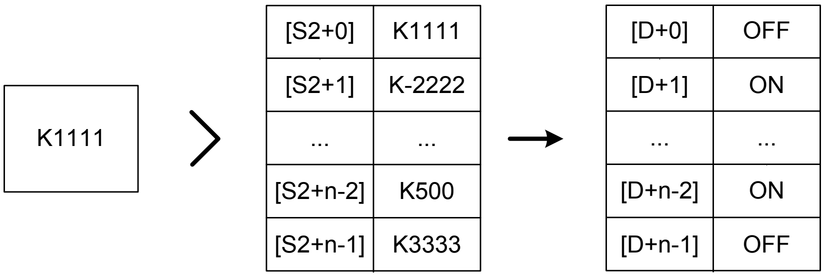
◆ Function

The n data entries (16- or 32-bit) from head addresses [S1] and [S2] are compared. The result is stored in n units (16- or 32-bit) from head address [D].

The BKCMP> instruction is used as an example.

[S1+0]	K1111		[S2+0]	K1111		[D+0]	OFF
[S1+1]	K1111		[S2+1]	K-2222		[D+1]	ON
...	...	>	...	...	→	...	...
[S1+n-2]	K1111		[S2+n-2]	K500		[D+n-2]	ON
[S1+n-1]	K1111		[S2+n-1]	K3333		[D+n-1]	OFF

A signed constant (16- or 32-bit) can be directly set in [S1]. The BKCMP> instruction is used as an example.



M8333 is set to ON when all of n results from head address [D] are ON.

An error is returned in the following conditions. The instruction is not executed. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when elements from head addresses [S1], [S2], and [D] are out of range.

2. Error 6705 is returned when 32-bit counters (C200 to C255) are used in 16-bit operation.

3. 32-bit counters are only applicable to 32-bit instructions (such as DBKCMP=, DBKCMP>, and DBKCMP<).

◆ Application

Element Name	data type	display format	current value
D0	16-bit int	Dec	1111
D1	16-bit int	Dec	-2222
D2	16-bit int	Dec	0
D3	16-bit int	Dec	500
D4	16-bit int	Dec	3333
M0	BOOL	Bin	OFF
M1	BOOL	Bin	ON
M2	BOOL	Bin	ON
M3	BOOL	Bin	ON
M4	BOOL	Bin	OFF

MCMP: Matrix comparison operation

◆ Overview

The MCMP instruction performs matrix comparison.

MCMP S1 S2 n D			Matrix comparison operation	Applicable model: H3U
S1	Matrix 1	Operand element 1 in an operation	16-bit instruction (9 steps) MCMP: Continuous execution MCMPP: Pulse execution	
S2	Matrix 2	Operand element 2 in an operation		
n	Data count	Number of data entries in an operation; value range: 1 to 256		
D	Pointer	Position number of the current bit		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) The bit patterns of n data entries from head addresses [S1] and [S2] in matrix format are compared starting from the ([D]+2)th bit to identify the conforming bit. The number of this bit is stored in [D].

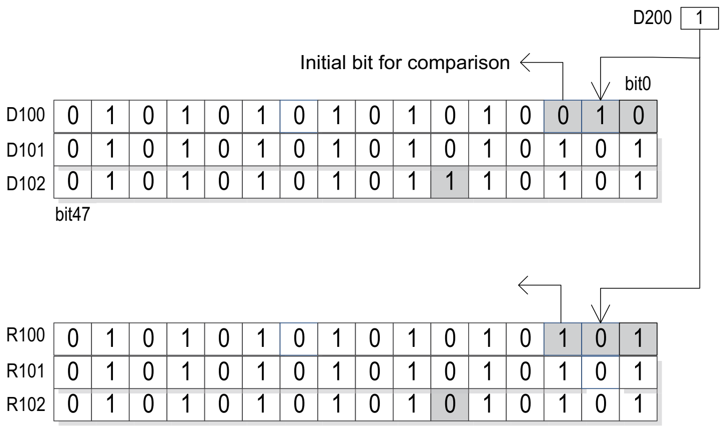
2) The matrix comparison flag M8320 determines the comparison rule. When M8320 = ON, same values are compared; when M8320 = OFF, different values are compared. When the conforming bit is found, comparison stops and the matrix bit found flag M8323 is set to ON. After the last bits from [S1] and [S2] are compared, the matrix search end flag M8321 is set to ON, the current number is stored in [D], and the instruction ends.

   If the conforming bit is the last bit, M8321 and M8323 are set to ON.

3) Searching starts from bit 0 when the next round of comparison is initiated or the matrix search start flag M8322 is set to ON.

4) The value in [D] ranges from 0 to (16n – 1). If [D] is out of range, the pointer error flag M8324 is set to ON and the instruction is not executed.

◆ Application



The initial value of D200 is set to 1, and comparison starts from the third bit (bit 2).

The following results are returned in sequence when M100 switches from OFF to ON:

D200 = 2, the matrix bit found flag M8323 is set to ON, and the matrix search end flag M8321 is set to OFF.

D200 = 37, the matrix bit found flag M8323 is set to ON, and the matrix search end flag M8321 is set to OFF.

200 = 47, the matrix bit found flag M8323 is set to OFF, and the matrix search end flag M8321 is set to ON.

D200 = 0, the matrix bit found flag M8323 is set to ON, and the matrix search end flag M8321 is set to OFF.

D200 = 1, the matrix bit found flag M8323 is set to ON, and the matrix search end flag M8321 is set to OFF.

- Flags

M8320: Matrix comparison flag. When it is set to OFF, different values are compared; when it is set to ON, same values are compared.

M8321: Matrix search end flag. It is set to ON after the last bits from [S1] and [S2] are compared.

M8322: Matrix search start flag. When it is set to ON, comparison starts from the first bit.

M8323: Matrix bit found flag. When the conforming bit is found, comparison stops and this flag is set to ON.

M8324: Matrix pointer error flag. It is set to ON when the pointer exceeds the range 0 to (16n – 1).

4.6.3 Matrix Read/Write

Matrix read/write	MBRD	Matrix bit read operation
	MBWR	Matrix bit write operation

MBRD: Matrix bit read operation

◆ Overview

The MBRD instruction reads bits in a matrix.

MBRD S n D			Matrix bit read operation	Applicable model: H3U
S	Matrix	Operand element in an operation	16-bit instruction (7 steps) MBRD: Continuous execution MBRDP: Pulse execution	
n	Data count	Number of data entries in an operation; value range: 1 to 256		
D	Pointer	Position number of the current bit		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

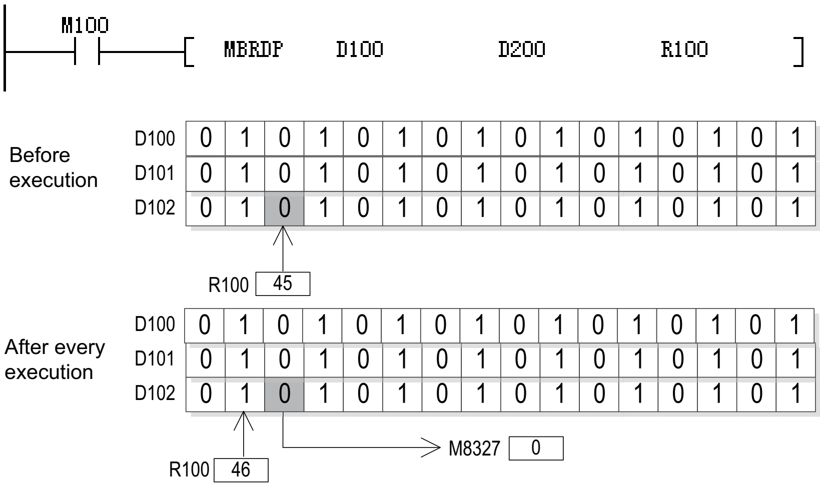
Note: The elements in gray background are supported.

◆ Function

1) The ([D]+1)th bit in [S] is read. The ON/OFF state of the read bit is stored in M8327 (carry flag of matrix shift output). After a bit is read, the pointer in [D] is incremented by 1 if the matrix pointer increment flag M8325 is set to ON. After the last bit is read, the matrix search end flag M8321 is set to ON, the pointer records the number of this bit, and the instruction ends. If M8326 = ON, the pointer is cleared and reading starts from the first bit (bit 0) in [S].

2) The value in [D] ranges from 0 to (16n – 1). If [D] is out of range, the pointer error flag M8324 is set to ON and the instruction is not executed.

◆ Application



The initial value of R100 is set to 45, and reading starts from bit 45 (the 46th bit).

M8325 is set to ON. The following results are returned in sequence when M100 switches from OFF to ON:

R100 = 45, the carry flag of matrix shift output M8327 is set to OFF, and the matrix search end flag M8321 is set to OFF.

R100 = 46, the carry flag of matrix shift output M8327 is set to ON, and the matrix search end flag M8321 is set to OFF.

R100 = 47, the carry flag of matrix shift output M8327 is set to OFF, and the matrix search end flag M8321 is set to OFF.

R100 = 47, the carry flag of matrix shift output M8327 remains OFF, and the matrix search end flag M8321 is set to ON.

● Flags

M8321: Matrix search end flag. It is set to ON when the last bit is read.

M8324: Matrix pointer error flag. It is set to ON when the pointer exceeds the range 0 to (16n – 1).

M8325: Matrix pointer increment flag, used to increment the pointer by 1.

M8326: Matrix pointer clearance flag, used to clear the pointer.

M8327: Carry flag of matrix shift output

MBWR: Matrix bit write operation

◆ Overview

The MBWR instruction writes bits in a matrix.

MBWR S n D			Matrix bit write operation	Applicable model: H3U
S	Matrix	Operand element in an operation	16-bit instruction (7 steps) MBWR: Continuous execution MBWRP: Pulse execution	
n	Data count	Number of data entries in an operation; value range: 1 to 256		
D	Pointer	Position number of the current bit		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

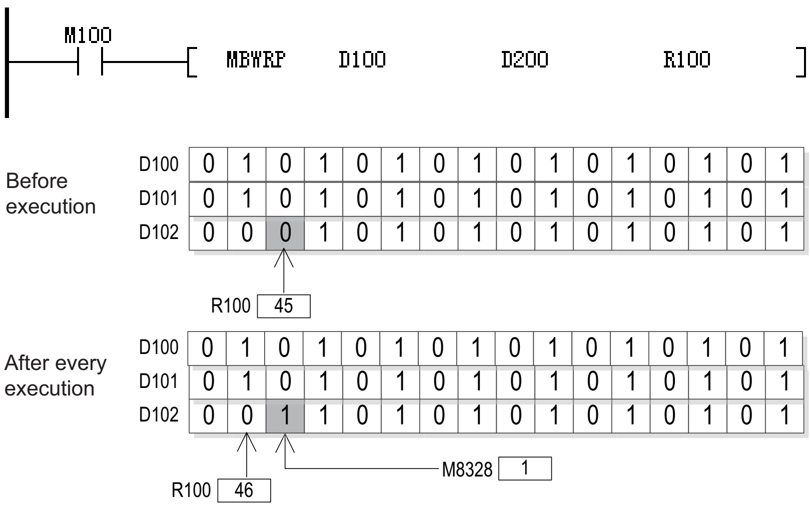
Note: The elements in gray background are supported.

◆ Function

1) The ON/OFF state of M8328 (placeholder flag of matrix shift input) is written to the ([D]+1)th bit in [S]. After a bit is written, the pointer in [D] is incremented by 1 if the matrix pointer increment flag M8325 is set to ON. After the last bit is written, the matrix search end flag M8321 is set to ON, the pointer records the number of this bit, and the instruction ends. If M8326 = ON, the pointer is cleared and writing starts from the first bit (bit 0) in [S].

2) The value in [D] ranges from 0 to (16n – 1). If [D] is out of range, the pointer error flag M8324 is set to ON and the instruction is not executed.

◆ Application



The initial value of R100 is set to 45, and writing starts from bit 45 (the 46th bit).

M8325 and M8328 (placeholder flag of matrix shift input) are both set to ON. The following results are returned in sequence when M100 switches from OFF to ON:

R100 = 45, bit 45 in D102 is set to 1, and the matrix search end flag M8321 is set to OFF.

R100 = 46, bit 46 in D102 is set to 1, and the matrix search end flag M8321 is set to OFF.

R100 = 47, bit 47 in D102 is set to 1, and the matrix search end flag M8321 is set to OFF.

R100 = 47, bit 47 in D102 remains 1, and the matrix search end flag M8321 is set to ON.

- Flags

M8321: Matrix search end flag. It is set to ON when the last bit is written.

M8324: Matrix pointer error flag. It is set to ON when the pointer exceeds the range 0 to (16n – 1).

M8325: Matrix pointer increment flag, used to increment the pointer by 1.

M8326: Matrix pointer clearance flag, used to clear the pointer.

M8328: Placeholder flag of matrix shift input

4.6.4 Matrix Rotation and Shift

Matrix rotation and shift	MBS	Matrix bit shift operation
	MBR	Matrix bit rotation operation

MBS: Matrix bit shift operation

◆ Overview

The MBS instruction shifts bits in a matrix.

MBS S D n		Matrix bit shift operation	Applicable model: H3U	
S	Matrix	Operand element in an operation	16-bit instruction (7 steps) MBS: Continuous execution MBSP: Pulse execution	
D	Operation result	Start number of elements that store the operation result		
n	Data count	Number of data entries in an operation; value range: 1 to 256		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

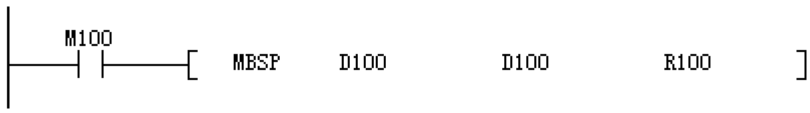
Note: The elements in gray background are supported.

◆ Function

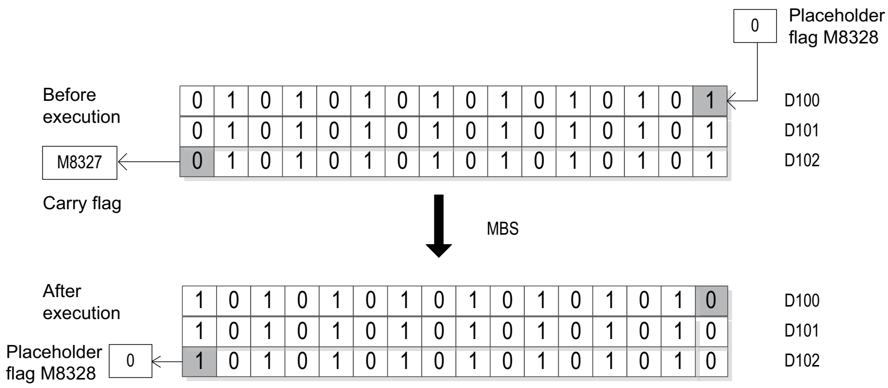
1) The bit pattern of a matrix composed of n 16-bit data entries in the range [S] to [S + n – 1] is shifted to the left or right. If M8329 is set to ON, the shift direction is right; if it is set to OFF, the shift direction is left. The bit position left blank after every shift, which is bit 0 in a left shift or bit (16n – 1) in a right shift, is filled with the placeholder flag M8328. The bit moved out of the matrix, which is bit (16n – 1) in a left shift or bit 0 in a right shift, is diverted to the carry flag M8327. The data after shift is stored in elements from head address [D].

2) The instruction of the pulse execution type (MBSP) is generally used.

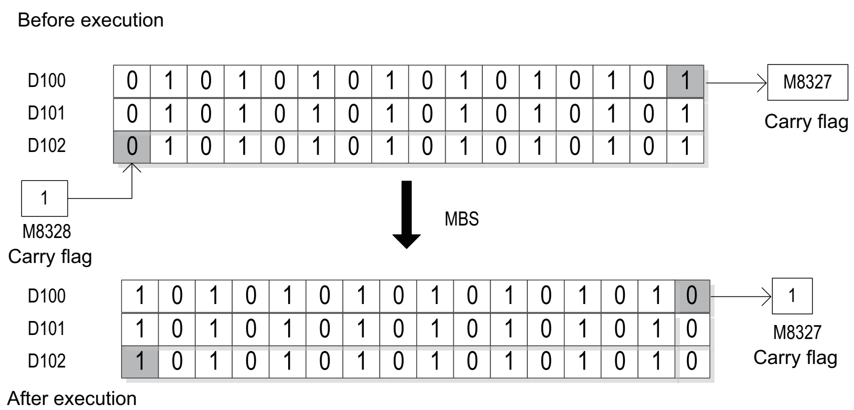
◆ Application



Assume that M8329 = OFF. The bit pattern of the following matrix is shifted to the left:



Assume that M8329 = ON. The bit pattern of the following matrix is shifted to the right:



- Flags

M8327: Carry flag of matrix shift output

M8328: Placeholder flag of matrix shift input

M8329: Matrix shift direction flag

MBR: Matrix bit rotation operation

◆ Overview

The MBR instruction rotates bits in a matrix.

MBR S D n			Matrix bit rotation operation	Applicable model: H3U	
S	Matrix	Operand element in an operation		16-bit instruction (7 steps) MBR: Continuous execution MBRP: Pulse execution	
D	Operation result	Start number of elements that store the operation result			
n	Data count	Number of data entries in an operation; value range: 1 to 256			

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

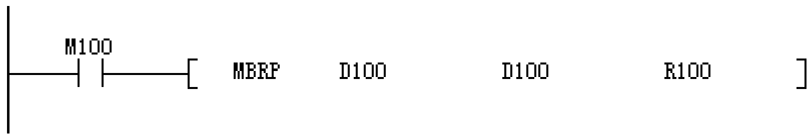
Note: The elements in gray background are supported.

◆ Function

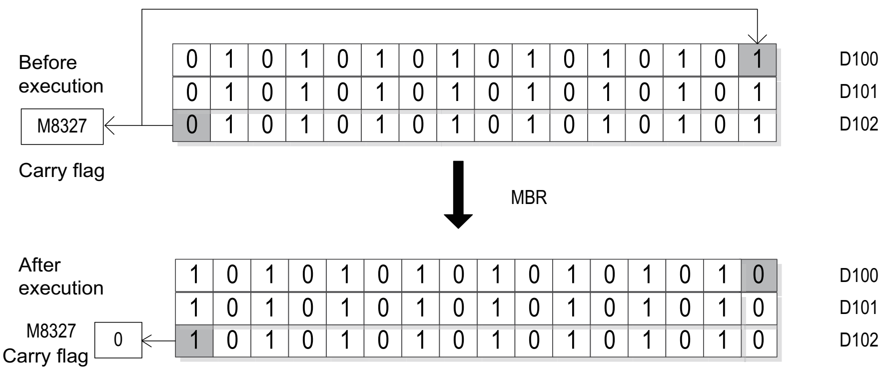
1) The bit pattern of a matrix composed of n 16-bit data entries in the range [S] to [S + n – 1] is rotated to the left or right. If M8329 is set to ON, the rotation direction is right; if it is set to OFF, the rotation direction is left. The bit position left blank after every rotation, which is bit 0 in a left rotation or bit (16n – 1) in a right rotation, is filled with the bit moved out of the matrix, which is bit (16n – 1) in a left rotation or bit 0 in a right rotation. This bit is also diverted to the carry flag M8327. The data after rotation is stored in elements from head address [D].

2) The instruction of the pulse execution type (MBRP) is generally used.

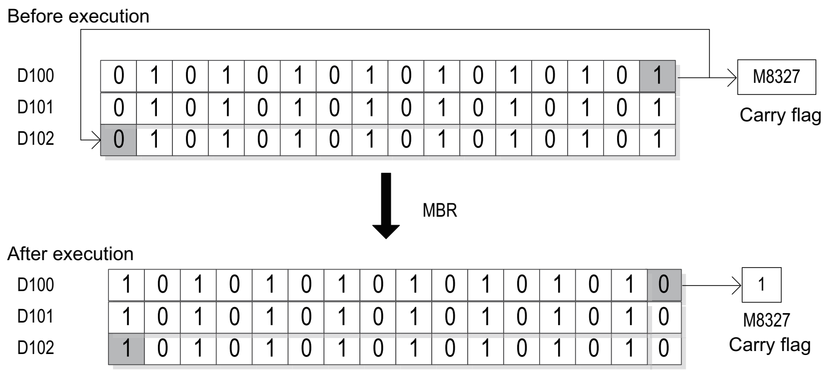
◆ Application



Assume that M8329 = OFF. The bit pattern of the following matrix is rotated to the left:



Assume that M8329 = ON. The bit pattern of the following matrix is rotated to the right:



- Flags

M8327: Carry flag of matrix shift output

M8329: Matrix shift direction flag

4.7 String Instructions

String instructions	STR	Conversion from BIN to string format
	VAL	Conversion from string to BIN format
	ESTR	Conversion from binary floating-point number to string format
	EVAL	Conversion from string to binary floating-point number format
	$+	String combination
	LEN	String length check
	INSTR	String retrieval
	RIGHT	String read right
	LEFT	String read left
	MIDR	Reading of one string from another
	MIDW	String replacement at any position
	$MOV	String transfer

STR: Conversion from BIN to string format

◆ Overview

The STR instruction converts BIN numbers into ASCII-encoded strings.

STR S1 S2 D		Conversion from BIN to string format	Applicable model: H3U	
S1	Character count	Start number of elements that store the total number of characters contained in a string after conversion	16-bit instruction (7 steps) STR: Continuous execution STRP: Pulse execution	32-bit instruction (13 steps) DSTR: Continuous execution DSTRP: Pulse execution
S2	Converted data	Number of the element that stores the BIN number to be converted		
D	Output	Start number of elements that store the string after conversion		

◆ Operands

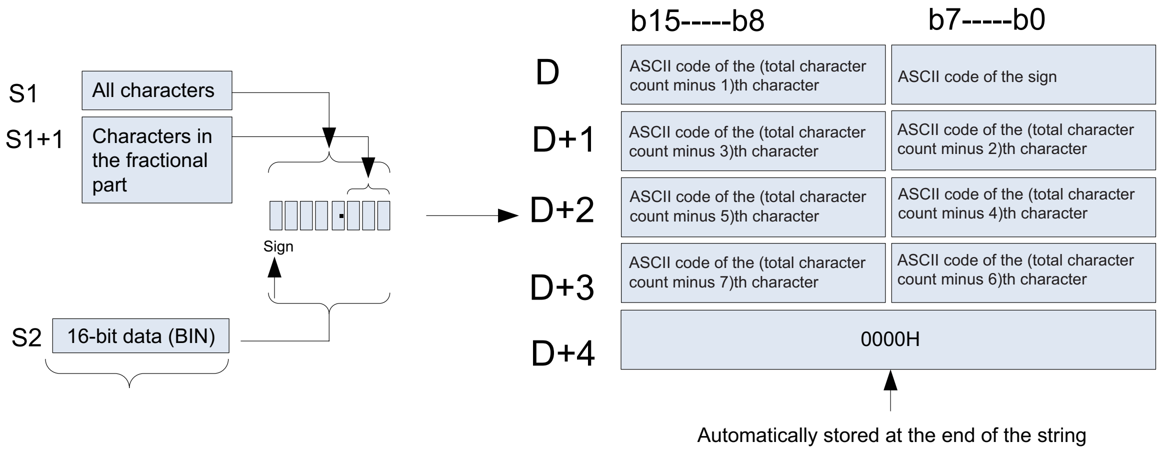
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

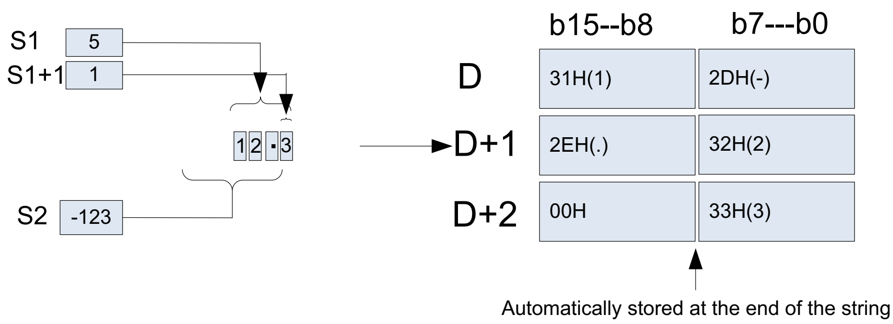
Note: The elements in gray background are supported.

◆ Function

1) 16-bit operation (STR and STRP)

- A decimal point is inserted into the 16-bit BIN number in [S2] to convert it to a string. The position of insertion is specified collectively by [S1] (which indicates the total number of characters) and [S1+1] (which indicates the number of characters in the fractional part). The result is stored in elements from head address [D].





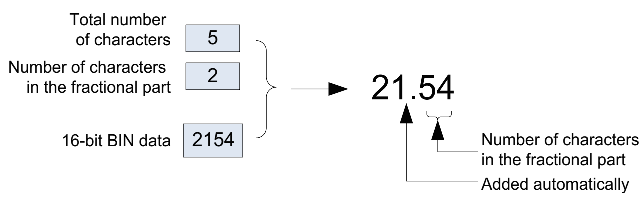
4

① The value in [S1] ranges from 2 to 8.

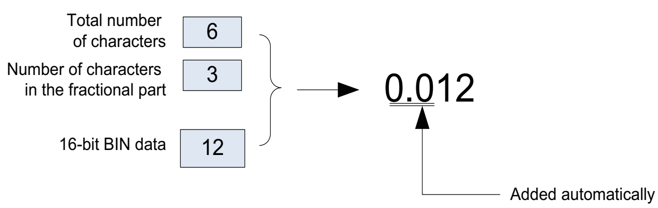
② The value in [S1+1] ranges from 0 to 5. The value in [S1+1] must be less than or equal to the value in [S1] minus 3.

③ The 16-bit BIN number ranges from –32,768 to +32,767. The string after conversion is stored in elements from head address [D].

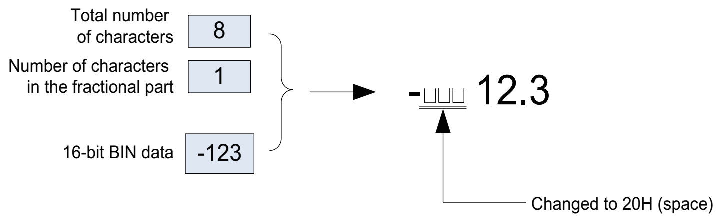
- When the 16-bit BIN number in [S2] is positive, the sign is encoded into 20H (space); when it is negative, the sign is encoded into 2DH (minus sign).
- When a number other than 0 is set in [S1+1], a decimal point (2EH) is inserted at the immediate position prior to a number of characters indicated by [S1+1].
- No decimal point is inserted when the value in [S1+1] is 0.



- If the value in [S1+1] is greater than the number of characters contained in the 16-bit BIN number in [S2], the system aligns characters to the right and then adds 0s (30H) on the left.



- With the decimal point and sign excluded, if the total number of characters indicated by [S1] is greater than the number of characters contained in the 16-bit BIN number in [S2], spaces (20H) are inserted between the sign and the value.
- An error will occur when the number of characters contained in the 16-bit BIN number in [S2] is greater.

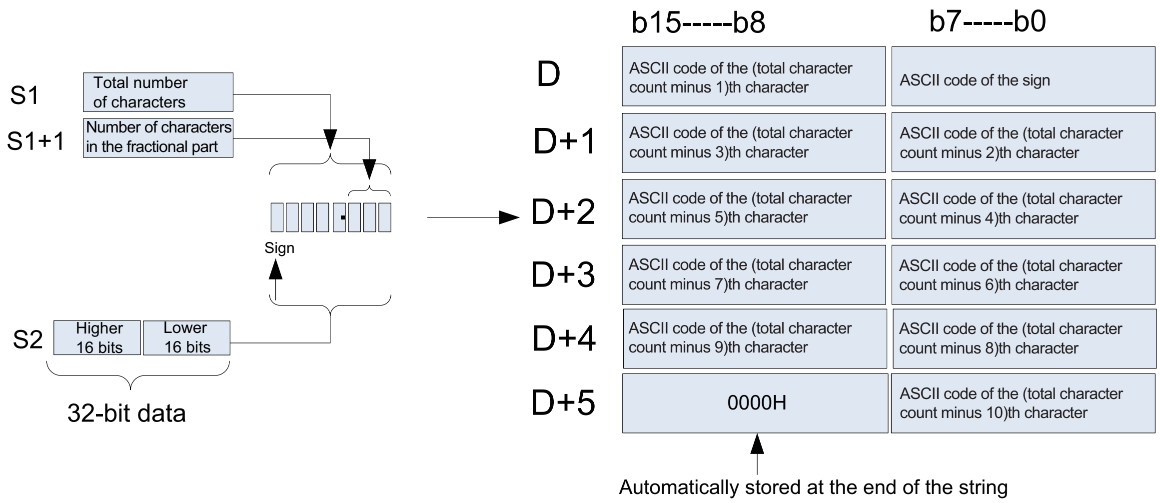


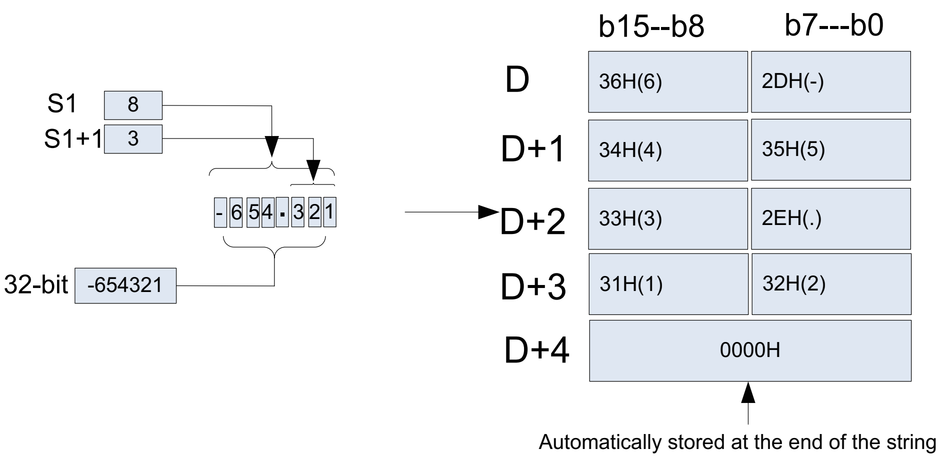
- The string after conversion is appended with 00H to indicate the end of the string.
- When the total number of characters is even, 0000H is stored in the element after the one that stores the last character. When the total number of characters is odd, 00H is stored in the higher byte (eight bits) of the element that stores the last character.

2. 32-bit operation (DSTR and DSTRP)

- A decimal point is inserted into the 32-bit BIN number in [S2+1, S2] to convert it to a string. The position of insertion is specified collectively by [S1] (which indicates the total number of characters) and [S1+1] (which indicates the number of characters in the fractional part). The result is stored in elements from head address [D].

4





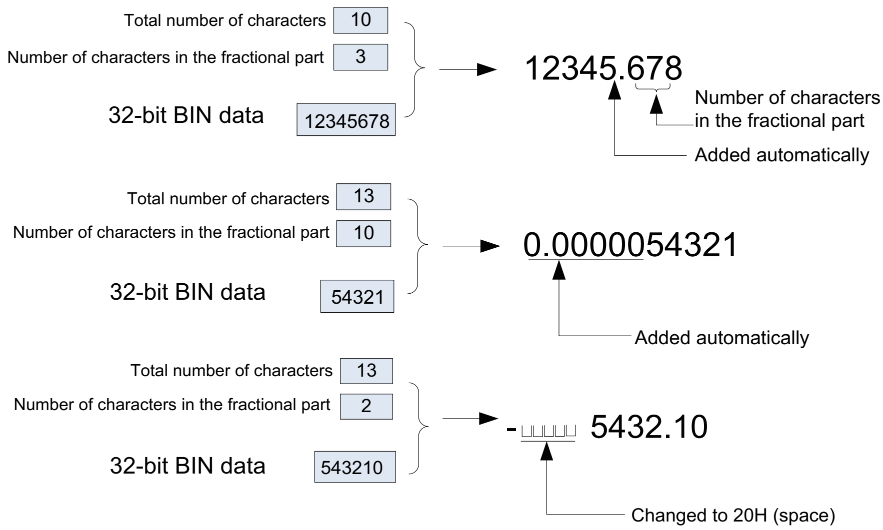
① The value in [S1] ranges from 2 to 13.

② The value in [S1+1] ranges from 0 to 10. The value in [S1+1] must be less than or equal to the value in [S1] minus 3.

③ The 32-bit BIN number ranges from –2,147,483,648 to +2,147,483,647.

④ The string after conversion is stored in elements from head address [D].

- When the 32-bit BIN number in [S2] is positive, the sign is encoded into 20H (space); when it is negative, the sign is encoded into 2DH (minus sign).
- When a number other than 0 is set in [S1+1], a decimal point (2EH) is inserted at the immediate position prior to a number of characters indicated by [S1+1]. No decimal point is inserted when the value in [S1+1] is 0.
- If the value in [S1+1] is greater than the number of characters contained in the 32-bit BIN number in [S2+1, S2], the system aligns characters to the right and then adds 0s (30H) on the left.
- With the decimal point and sign excluded, if the total number of characters indicated by [S1] is greater than the number of characters contained in the 32-bit BIN number in [S2+1, S2], spaces (20H) are inserted between the sign and the value.
- An error will occur when the number of characters contained in the 32-bit BIN number in [S2] is greater.
- The string after conversion is appended with 00H to indicate the end of the string.
- When the total number of characters is even, 0000H is stored in the element after the one that stores the last character. When the total number of characters is odd, 00H is stored in the higher byte (eight bits) of the element that stores the last character.



3. Errors

An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- The value in [S1] is out of range. (Error code: K6706)

	Setting Range
16-bit operation	2 to 8
32-bit operation	2 to 13

- The value in [S1+1] is out of range. (Error code: K6706)

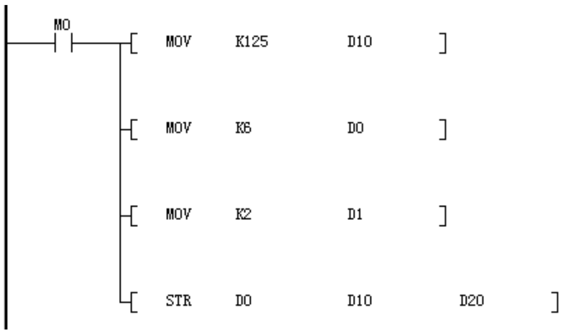
	Setting Range
16-bit operation	0 to 5
32-bit operation	0 to 10

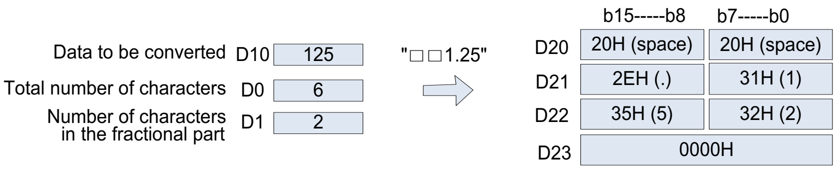
- The value in [S1+1] is greater than the value in [S1] minus 3. (Error code: K6706)
- The total number of characters (including the sign and decimal point) indicated by [S1] is less than the number of characters contained in the BIN number in [S2]. (Error code: K6706)

- Elements from head address [D] for string storage are out of range. (Error code: K6705)

◆ Application

When M0 = ON, the 16-bit BIN number in D10 is converted to a string by inserting a decimal point at the position collectively specified by D0 and D1. The result is stored in D20 to D23.





VAL: Conversion from string to BIN format

◆ Overview

The VAL instruction converts ASCII-encoded strings to BIN numbers.

VAL S D1 D2			Conversion from string to BIN format	Applicable model: H3U	
S	Converted data	Start number of elements that store the string to be converted		16-bit instruction (7 steps) VAL: Continuous execution VALP: Pulse execution	32-bit instruction (13 steps) DVAL: Continuous execution DVALP: Pulse execution
D1	Character count	Start number of elements that indicate the total number of characters contained in the string, as well as the number of characters in the fractional part			
D2	Output	Number of the element that stores the BIN number after conversion			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

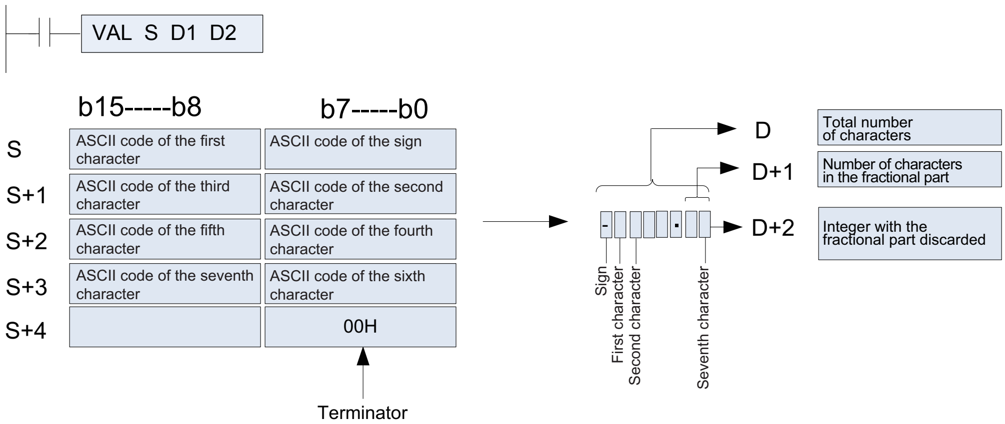
Note: The elements in gray background are supported.

◆ Function

1) 16-bit operation (VAL and VALP)

① The string stored in elements from head address [S] is converted to a 16-bit BIN number. The total number of characters contained in the string is stored in [D1], and the number of characters in the fractional part is stored in [D+1]. The BIN number is stored in [D2].

② During the string-to-BIN conversion, the data within the range from [S] to the element that stores 00H is processed as a string.



③ String to be converted

- The total number of characters contained in the string to be converted and its value (with the decimal point ignored) must be within the following ranges:

	Setting Range
Total number of characters	2 to 8
Number of characters in the fractional part	0 to 5
Value range (with the decimal point ignored)	–32,768 to +32,767 For example, 123.45 is processed as 12345.

- Types of characters to be converted

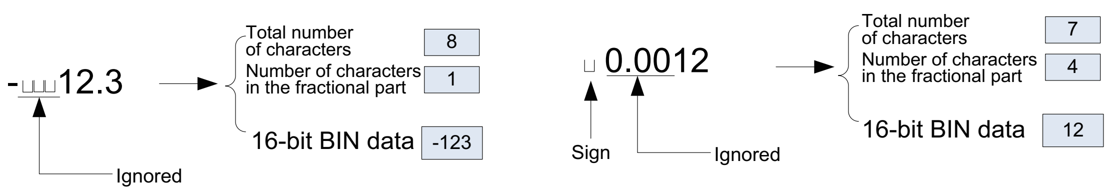
	Character Type
Positive number	Space (20H)
Negative number	Minus sign (2DH)
Decimal point	Period (2EH)
Digit	0 (30H) to 9 (39H)

④ [D1] indicates the total number of characters, including the digits, sign, and decimal point.

⑤ [D1+1] indicates the number of characters in the fractional part, that is, the characters to the right of the decimal point (2EH).

⑥ [D2] stores the 16-bit BIN number converted from a string with the decimal point ignored.

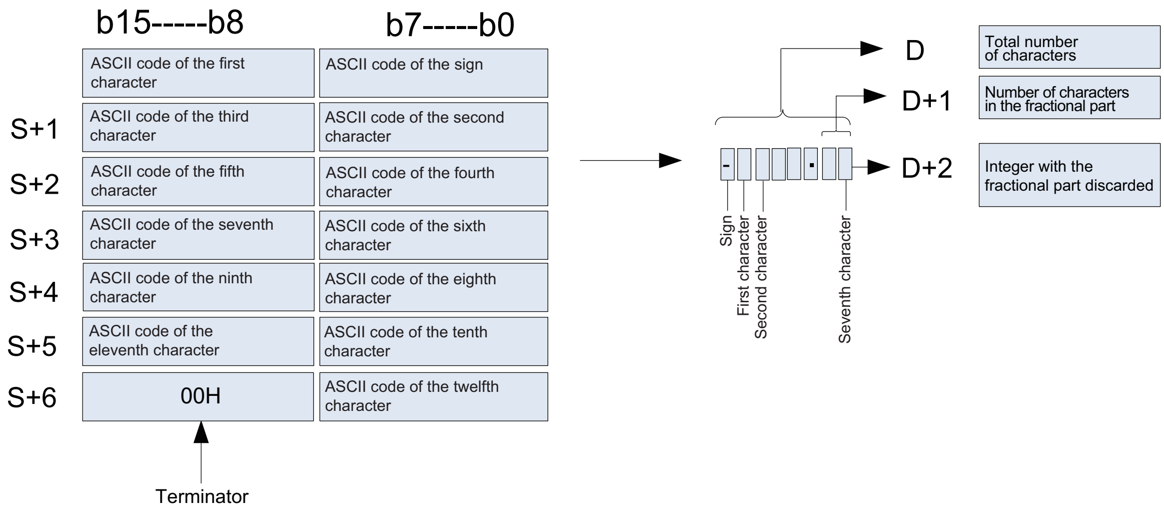
When a string is converted to a 16-bit BIN number, the spaces (20H) or 0s (30H) between the sign and a set of digits other than 0 are ignored.



2. 32-bit operation (DVAL and DVALP)

① The string stored in elements from head address [S] is converted to a 32-bit BIN number. The total number of characters contained in the string is stored in [D1], and the number of characters in the fractional part is stored in [D+1]. The BIN number is stored in [D2+1, D2].

② During the string-to-BIN conversion, the data within the range from [S] to the element that stores 00H is processed as a string.



③ String to be converted

- The total number of characters contained in the string to be converted and its value (with the decimal point ignored) must be within the following ranges:

	Setting Range
Total number of characters	2 to 8
Number of characters in the fractional part	0 to 10
Value range (with the decimal point ignored)	–2,147,483,648 to +2,147,483,647 For example, 123.45 is processed as 12345.

- Types of characters to be converted

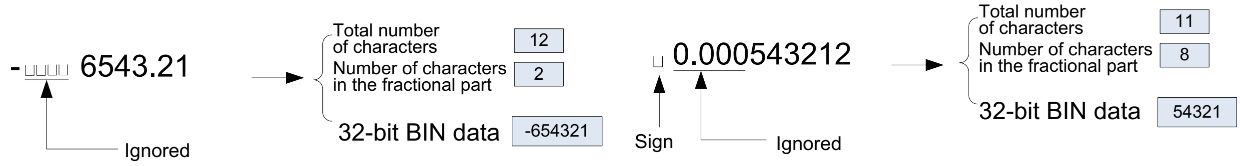
	Character Type
Positive number	Space (20H)
Negative number	Minus sign (2DH)
Decimal point	Period (2EH)
Digit	0 (30H) to 9 (39H)

④ [D1] indicates the total number of characters, including the digits, sign, and decimal point.

⑤ [D1+1] indicates the number of characters in the fractional part, that is, the characters to the right of the decimal point (2EH).

⑥ [D2+1, D2] stores the 32-bit BIN number converted from a string with the decimal point ignored.

When a string is converted to a 32-bit BIN number, the spaces (20H) or 0s (30H) between the sign and a set of digits other than 0 are ignored.



◆ Note

- The sign, which is encoded into 20H (space) or 2DH (minus sign), must be stored in the first byte (composed of the lower eight bits of the initial element from head address [S]).
- Only digits 0 (30H) to 9 (39H), spaces (20H), and decimal points (2EH) can be stored in the ASCII-encoded data area within the range from the second byte of [S] to the string end marker 00H. An operation error will occur when 2DH (minus sign) is stored after the second byte. (Error code: K6706)

◆ Errors

An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- The number of characters contained in the string from head address [S] is out of range. (Error code: K6706)

	Setting Range
16-bit operation	2 to 8
32-bit operation	2 to 13

- The number of characters in the fractional part of the string from head address [S] is out of range. (Error code: K6706)

	Setting Range
16-bit operation	0 to 5
32-bit operation	0 to 10

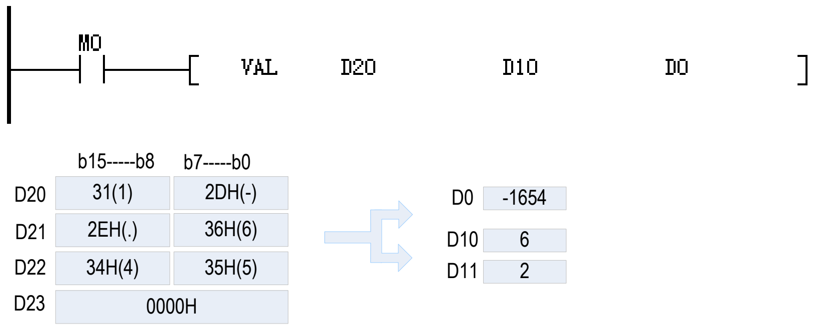
- The value in [D1+1] is greater than the value in [D1] minus 3. (Error code: K6706)
- The ASCII code of the sign is neither 20H (space) nor 2DH (minus sign). (Error code: K6706)
- The ASCII code of a character is out of the range 30H (0) to 39H (9) or is not 2EH (decimal point). (Error code: K6706)
- Multiple decimal points (2EH) are encoded for the string from head address [S]. (Error code: K6706)
- The BIN number after conversion is out of range. (Error code: K6706)

	Setting Range
16-bit operation	–32,768 to +32,767
32-bit operation	–2,147,483,648 to +2,147,483,647

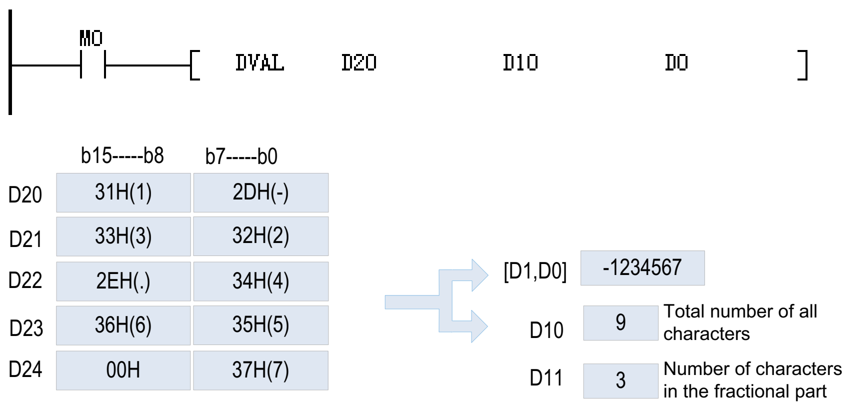
- 00H does not exist within the range of elements from head address [S]. (Error code: K6706)

◆ Application

1) After M0 is set to ON, the string stored in D20 to D22 is converted in integer format to a BIN number. The result is stored in D0.



2) After M0 is set to ON, the string stored in D20 to D24 is converted in integer format to a BIN number. The result is stored in [D1, D0].



ESTR: Conversion from binary floating-point number to string format

◆ Overview

The ESTR instruction converts binary floating-point numbers (real numbers) to ASCII-encoded strings containing the specified number of characters.

ESTR S1 S2 D		Conversion from binary floating-point number to string format	Applicable model: H3U	
S1	Operand	Start number of elements that store the binary floating-point number to be converted		32-bit instruction (13 steps) DESTR: Continuous execution DESTRP: Pulse execution
S2	Start number	Start number of elements that store the display format of the string after conversion		
D	Result	Start number of elements that store the string after conversion		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

4

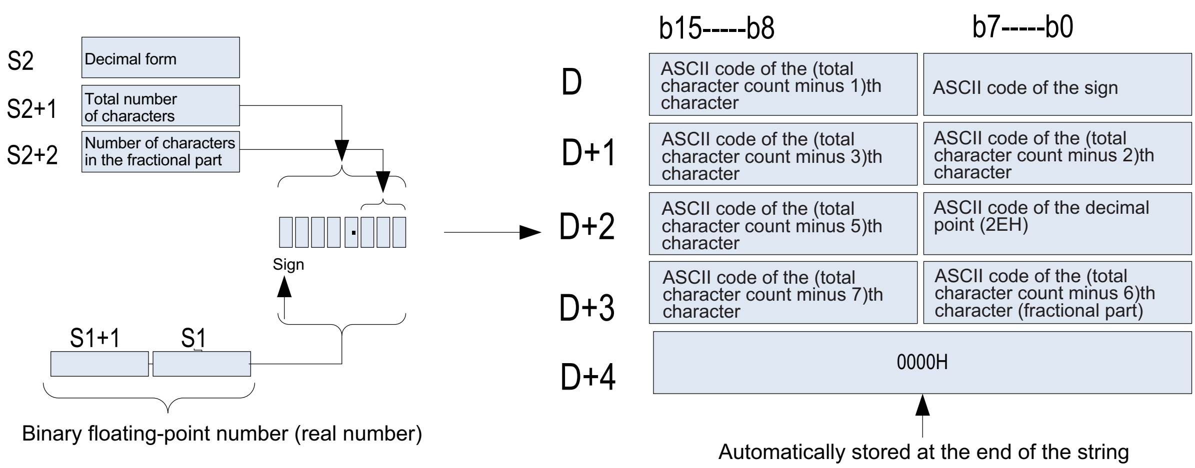
◆ Function

1) 32-bit operation (DESTR)

The binary floating-point number in [S+1, S] is converted to a string based on the content of [S2, S2+1, S2+2]. The result is stored in elements from head address D.

S2	0: Decimal form 1: Exponential form	The conversion result varies with different S2 forms.
S2+1	Total number of characters	Setting range: 2 to 24
S2+2	Number of characters in the fractional part	

2) Decimal form



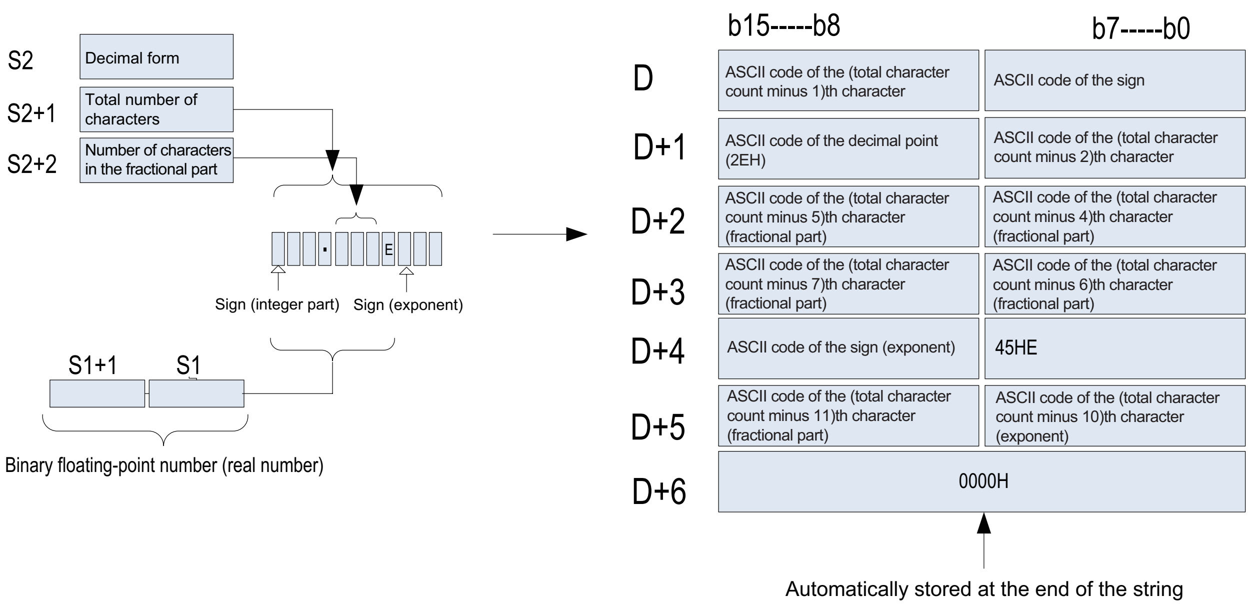
- The total number of characters (max.: 24) is specified in [S2+1] based on the following rules:

Total number of characters ≥ 2 when the number of characters in the fractional part is equal to 0

Total number of characters ≥ Number of characters in the fractional part + 3 when the number of characters in the fractional part is not 0

- The number of characters in the fractional part specified in [S2+2] ranges from 0 to 7. Ensure that the number of characters in the fractional part is not greater than the total number of characters minus 3.

3) Exponential form



- The total number of characters (max.: 24) is specified in [S2+1] based on the following rules:

Total number of characters ≥ 6 when the number of characters in the fractional part is equal to 0

Total number of characters ≥ Number of characters in the fractional part + 7 when the number of characters in the fractional part is not 0

- The number of characters in the fractional part specified in [S2+2] ranges from 0 to 7. Ensure that the number of characters in the fractional part is not greater than the total number of characters minus 7.

◆ Errors

An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- The value in [S1] is out of range. (Error code: K6706)

$$0,\ \pm 2^{-126} \leqslant [S1] < \pm 2^{128}$$

4

- The value in [S2] is neither 0 nor 1. (Error code: K6706)
- The total number of characters specified in [S2+1] is out of range. (Error code: K6706)
- Decimal form

Total number of characters ≥ 2 when the number of characters in the fractional part is equal to 0

Total number of characters ≥ Number of characters in the fractional part + 3 when the number of characters in the fractional part is not 0

- Exponential form

Total number of characters ≥ 6 when the number of characters in the fractional part is equal to 0

Total number of characters ≥ Number of characters in the fractional part +7 when the number of characters in the fractional part is not 0

- The number of characters in the fractional part specified in [S2+2] is out of range. (Error code: K6706)

Decimal form: Number of characters in the fractional part ≤ Total number of characters – 3

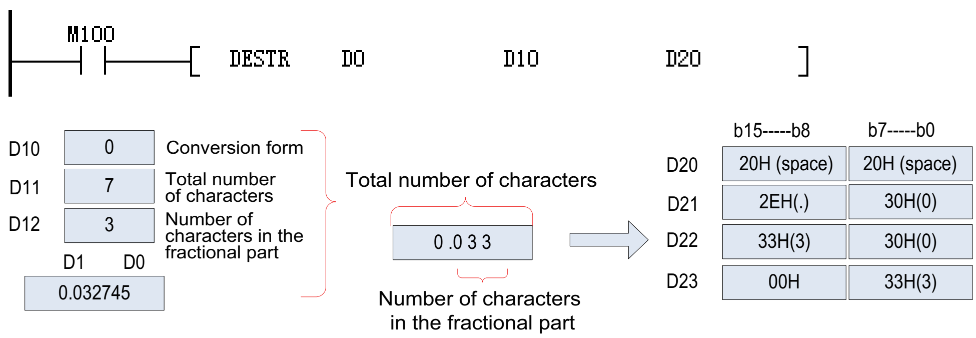
Exponential form: Number of characters in the fractional part ≤ Total number of characters – 7

- The elements from head address [D] for string storage are out of range. (Error code: K6705)
- The number of characters contained in the conversion result exceeds the designated total number of characters. (Error code: K6705)

◆ Application

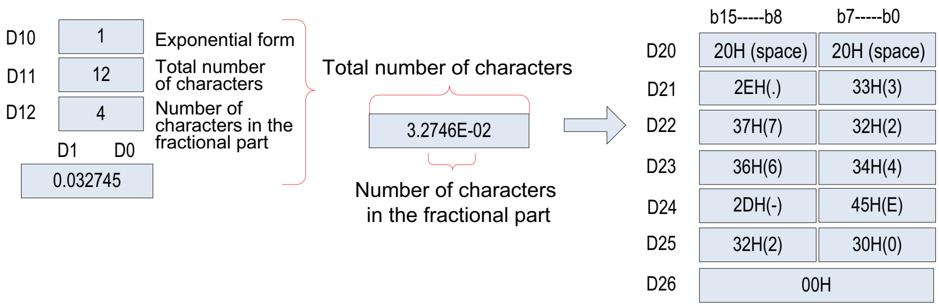
When M100 = ON, the binary floating-point number in D0 and D1 is converted based on the content (decimal form) of D10 to D12.The result is stored in elements from head address D20.

- Before execution of the instruction



Element Name	data type	display format	current value
D0	float	Dec	0.032745
D10	16-bit int	Dec	0
D11	16-bit int	Dec	7
D12	16-bit int	Dec	3
M100	BOOL	Bin	OFF
D20	16-bit int	Hex	0x2020
D21	16-bit int	Hex	0x2E30
D22	16-bit int	Hex	0x3330
D23	16-bit int	Hex	0x33

When M100 = ON, the binary floating-point number in D0 and D1 is converted based on the content (exponential form) of D10 to D12.The result is stored in elements from head address D20.



Element Name	data type	display format	current value
D0	float	Dec	0.032745
D10	16-bit int	Dec	1
D11	16-bit int	Dec	12
D12	16-bit int	Dec	4
M100	BOOL	Bin	OFF
D20	16-bit int	Hex	0x2020
D21	16-bit int	Hex	0x2E33
D22	16-bit int	Hex	0x3732
D23	16-bit int	Hex	0x3534
D24	16-bit int	Hex	0x2D45
D25	16-bit int	Hex	0x3230
D26	16-bit int	Hex	0x0

EVAL: Conversion from string to binary floating-point number format

◆ Overview

The EVAL instruction converts ASCII-encoded strings to binary floating-point numbers.

EVAL S D		Conversion from string to binary floating-point number format	Applicable model: H3U	
S	Operand	Start number of elements that store the string to be converted to a binary floating-point number		32-bit instruction (9 steps) DEVAL: Continuous execution DEVALP: Pulse execution
D	Result	Start number of elements that store the binary floating-point number after conversion		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

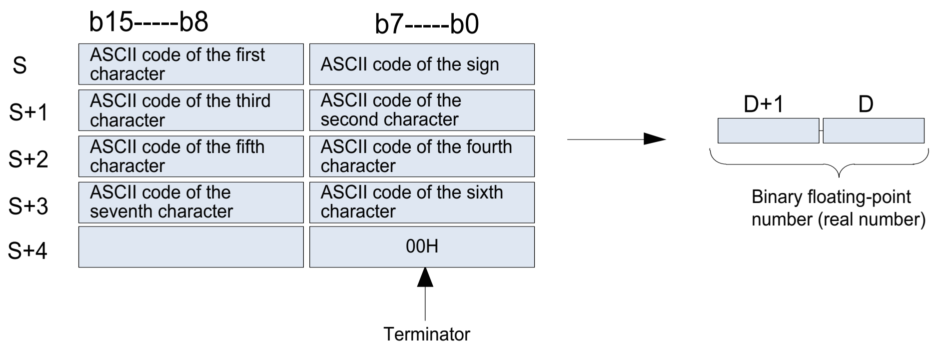
Note: The elements in gray background are supported.

◆ Function

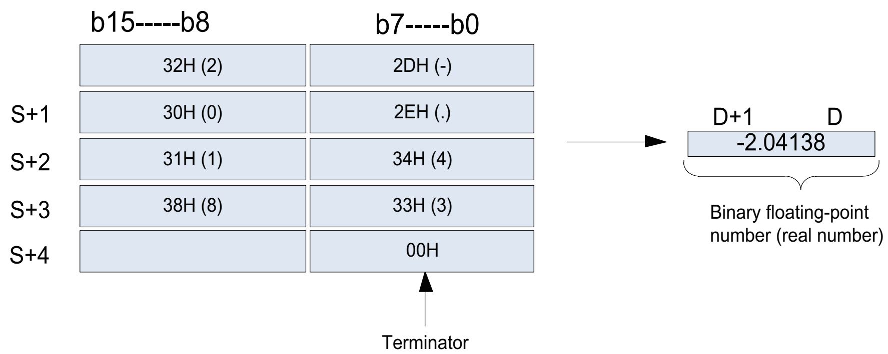
1) 32-bit operation (DESTR)

The string stored in elements from head address [S] is converted to a binary floating-point number. The result is stored in [D+1, D].

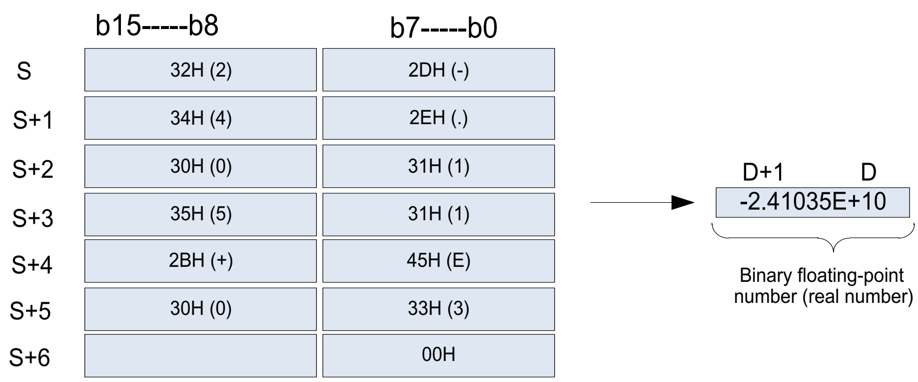
The string can be in decimal or exponential form.



● Decimal form



● Exponential form



4

If the source string contains more than seven characters excluding the sign, decimal point, and exponent, the eighth character and subsequent ones are discarded.

-2.563697453

Discarded

-1.35689004E-6

Discarded

- In decimal form, if the sign is encoded into 2BH (plus sign) in [S] or omitted, the string is converted to a positive number. If the sign is encoded into 2DH (minus sign), the string is converted to a negative number.

- In exponential form, if the sign of the exponent is encoded into 2BH (plus sign), the string is converted to a positive exponent after the sign is omitted. If the sign is encoded into 2DH (minus sign), the string is converted to a negative exponent.
- If the source string contains spaces (20H) or 0s (30H) between digits other than the initial 0, 20H or 30H is ignored when the string is converted.

- 01.245

gnored

- The source string can contain a maximum of 24 characters, including spaces (20H) and 0s (30H).

◆ Related elements

Element	Name	Description	
		Condition	Action
M8020	Zero flag	The conversion result is 0 (mantissa = 0).	The zero flag M8020 is set to ON.
M8021	Borrow flag	The absolute value of the conversion result is less than 2^{-126}.	The part of the value in D that is less than 2^{-126} (minimum absolute value of a 32-bit real number) is discarded, and the borrow flag M8021 is set to ON.
M8022	Carry flag	The absolute value of the conversion result is greater than or equal to 2^{128}.	The part of the value in D that is greater than 2^{128} (minimum absolute value of a 32-bit real number) is discarded, and the carry flag M8022 is set to ON.

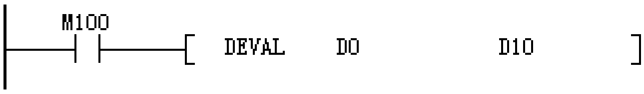
◆ Errors

An operation error occurs in the following conditions. The error flag M8067 is set ton ON to identify this error and the error code is stored in D8067.

- The integer and fractional parts contain characters out of the range 0 (30H) to 9 (39H). (Error code: K6706)
- The string from head address [S] contains two or more decimal points (2EH). (Error code: K6706)
- The exponent contains characters other than E (45H), plus sign (2BH), and minus sign (2DH), or multiple exponents exist. (Error code: K6706)
- Elements from head address [S] do not contain 00H. (Error code: K6705)
- The number of characters after [S] is 0 or exceeds 24. (Error code: K6705)

◆ Application

1) When M101 = ON, the string stored in elements from head address D0 is converted to a binary floating-point number (in decimal form). The result is stored in D10 and D11.



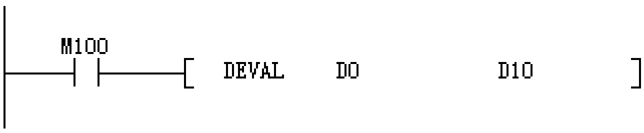
	b15-----b8	b7-----b0
D0	20H (space)	2DH(-)
D1	31H(1)	30H(0)
D2	32H(2)	2EH(.)
D3	34H(4)	33H(3)
D4	32H(2)	35H(5)
D5	00H	31H(1)

-□01.234521

⇨ D11 D10: -1.234521

Element Name	data type	display format	current value
D10	float	Dec	-1.234521
M100	BOOL	Bin	ON
D0	16-bit int	Hex	0x202D
D1	16-bit int	Hex	0x3130
D2	16-bit int	Hex	0x322E
D3	16-bit int	Hex	0x3433
D4	16-bit int	Hex	0x3235
D5	16-bit int	Hex	0x31

2) When M100 = ON, the string stored in elements from head address D0 is converted to a binary floating-point number (in exponential form). The result is stored in D10 and D11.



	b15-----b8	b7-----b0
D0	20H (space)	20H (space)
D1	2EH(.)	31H(1)
D2	33H(3)	32H(2)
D3	35H(5)	34H(4)
D4	2DH(-)	45H(E)
D5	32H(2)	30H(0)
D6	00H	00H

□□1.2345E-02

⇧ Space ignored

⇨ D11 D10: 0.012345

Element Name	data type	display format	current value
D10	float	Dec	0.012345
M100	BOOL	Bin	ON
D0	16-bit int	Hex	0x2020
D1	16-bit int	Hex	0x2E31
D2	16-bit int	Hex	0x3332
D3	16-bit int	Hex	0x3534
D4	16-bit int	Hex	0x2D45
D5	16-bit int	Hex	0x3230
D6	16-bit int	Hex	0x0

$+: String combination

◆ Overview

The $+ instruction connects strings together.

$+ S1 S2 D		String combination	Applicable model: H3U	
S1	String to be appended with another one	Start number of elements that store the source string to be appended with another string, or a directly designated string	16-bit instruction (7 steps) $+: Continuous execution $+P: Pulse execution	
S2	String to be appended to the source string	Start number of elements that store the string to be appended to the source string, or a directly designated string		
D	Connection result	Start number of elements that store the string after connection		

◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The string from head address [S2] is appended to the string from head address [S1]. The resulting string is stored in [D].

The characters in [S1] and [S2] are organized byte by byte. The first 00H byte indicates the string end.

[S1+0]	H2211
[S1+1]	H4433
[S1+2]	H0055

↑ 00H ignored

+

[S2+0]	H8877
[S2+1]	HAA99
[S2+2]	HCCBB

→

[D+0]	H2211
[D+1]	H4433
[D+2]	H7755
[D+3]	H9988
[D+4]	HBBAA
[D+5]	H00CC

The 00H that indicates the end of a string is ignored and the last character of this string is connected to a designated string. 00H is automatically appended to the new string.

- When the number of characters contained in the new string is odd, 00H is stored in the higher byte of the element that stores the last character.

- When the number of characters contained in the new string is even, 0000H is stored in the element after the one that stores the last character.
- When the value in [S1] or [S2] starts from 00H (which indicates the number of characters is 0), 0000H is stored in [D].

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6705 is returned when 00H is not found in the operand ranges of [S1] and [S2].
- Error 6705 is returned when the string after combination exceeds the operand range of [D].
- Error 6706 is returned when [S1] and [S2] have overlapping elements with [D].

◆ Application

MAIN

Window

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x2211
D101	16-bit int	Hex	0x4433
D102	16-bit int	Hex	0x55
D103	16-bit int	Hex	0x0
	16-bit int	Dec	
D200	16-bit int	Hex	0x8877
D201	16-bit int	Hex	0xAA99
D202	16-bit int	Hex	0xCCBB
D203	16-bit int	Hex	0xDD
	16-bit int	Hex	
R100	16-bit int	Hex	0x2211
R101	16-bit int	Hex	0x4433
R102	16-bit int	Hex	0x7755
R103	16-bit int	Hex	0x9988
R104	16-bit int	Hex	0xBBAA
R105	16-bit int	Hex	0xDDCC
R106	16-bit int	Hex	0x0

LEN: String length check

◆ Overview

The LEN instruction counts the characters (bytes) contained in a designated string..

LEN S D		String length check	Applicable model: H3U	
S	Checked data	Start number of elements that store the string whose characters will be counted	16-bit instruction (5 steps) LEN: Continuous execution LENP: Pulse execution	
D	Counting result	Number of the element that indicates the number of characters (bytes) contained in the string		

4

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

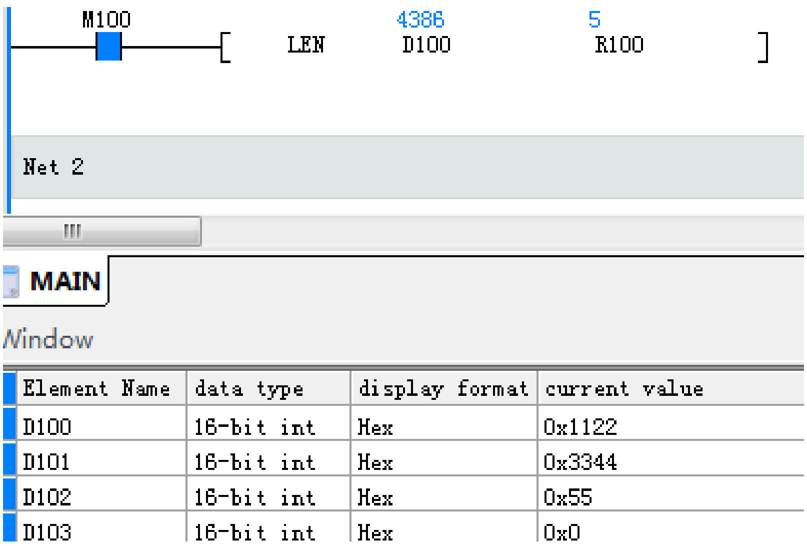
The characters contained in the string from head address S are counted. The result is stored in [D]. Characters are counted byte by byte within the range from [S] to the first 00H.

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned when 00H is not found within the value range from [S].

2. Error 6706 is returned when the number of characters is greater than 32,767.

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1122
D101	16-bit int	Hex	0x3344
D102	16-bit int	Hex	0x55
D103	16-bit int	Hex	0x0

INSTR: String retrieval

◆ Overview

The INSTR instruction retrieves one string from another.

INSTR S1 S2 D n		String retrieval	Applicable model: H3U	
S1	Source data	Start number of elements that store the string to be retrieved	16-bit instruction (7 steps) INSTR: Continuous execution INSTRP: Pulse execution	
S2	Retrieval source	Start number of elements that store the string from which another string will be retrieved		
D	Retrieval result	Start number of elements that store the retrieval result		
n	Retrieval start point	Position from which a string will be retrieved		

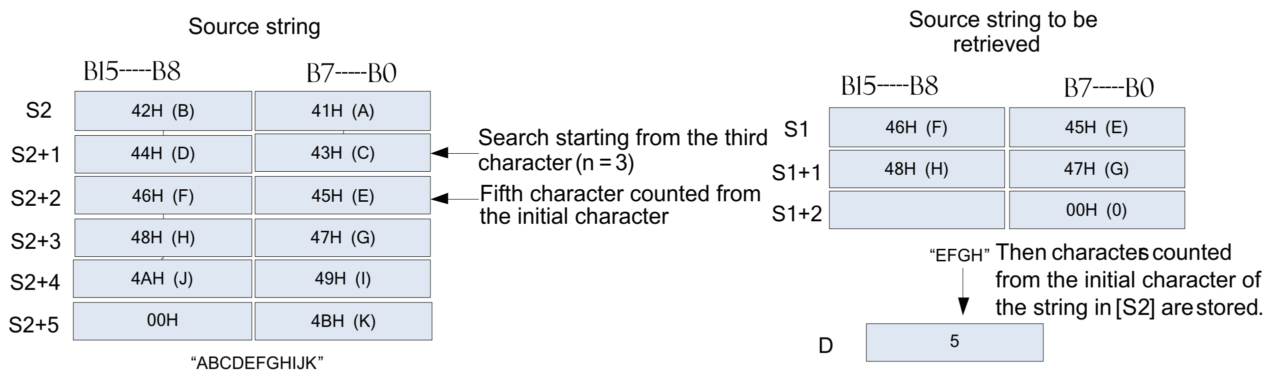
◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

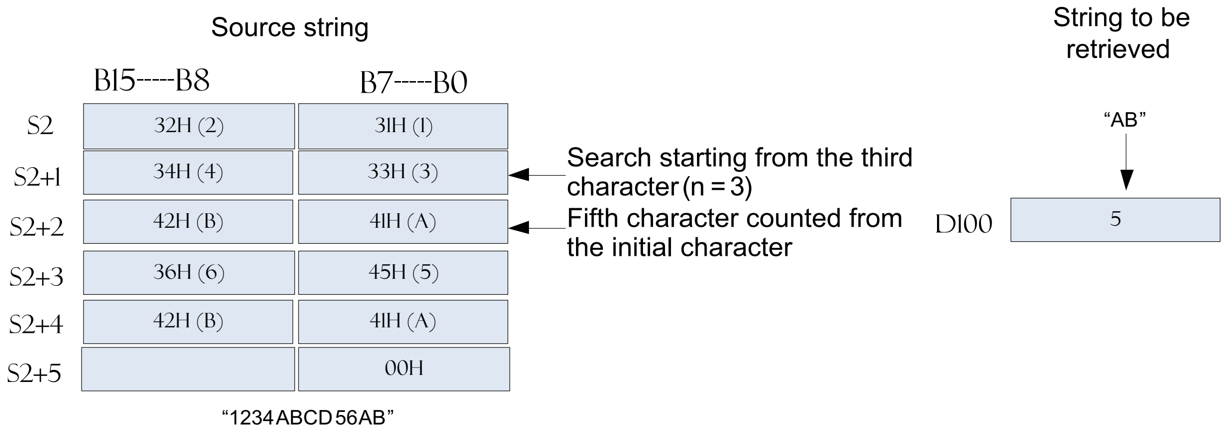
◆ Function

The INSTR instruction searches from the nth character on the left (beginning of a string) of a range of elements from the retrieval source [S2], to identify the same string as that stored in elements from head address [S1]. Information about the beginning (position of the first matched character on the left) of the retrieved string is stored in [D].



If no portion of the string from head address [S2] matches the string from head address [S1], 0 is stored in [D].

If n (retrieval start point) is negative or 0, the instruction is not executed.

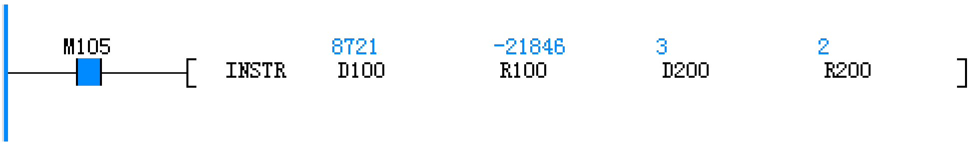


An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6706 is returned when the value of n (retrieval start point) is greater than the number of characters from head address [S2].

2. Error 6706 is returned when 00H is not found in elements from head address [S1] or [S2].

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x2211
D101	16-bit int	Hex	0x4433
D102	16-bit int	Hex	0x0
D103	16-bit int	Hex	0x0
	16-bit int	Dec	
D200	16-bit int	Hex	0x3
R200	16-bit int	Hex	0x2
	16-bit int	Hex	
	16-bit int	Hex	
	16-bit int	Hex	
R100	16-bit int	Hex	0xAAAA
R101	16-bit int	Hex	0x2211
R102	16-bit int	Hex	0x4433
R103	16-bit int	Hex	0xAAAA
R104	16-bit int	Hex	0xAAAA
R105	16-bit int	Hex	0x0

RIGHT: String read right

◆ Overview

The RIGHT instruction retrieves a designated number of characters at the right end of a string.

RIGHT S D n		String read right	Applicable model: H3U	
S	Source data	Start number of elements that store a string	16-bit instruction (7 steps) RIGHT: Continuous execution RIGHTP: Pulse execution	
D	Retrieval result	Start number of elements that store a retrieved string		
n	Retrieved character count	Number of characters to be retrieved		

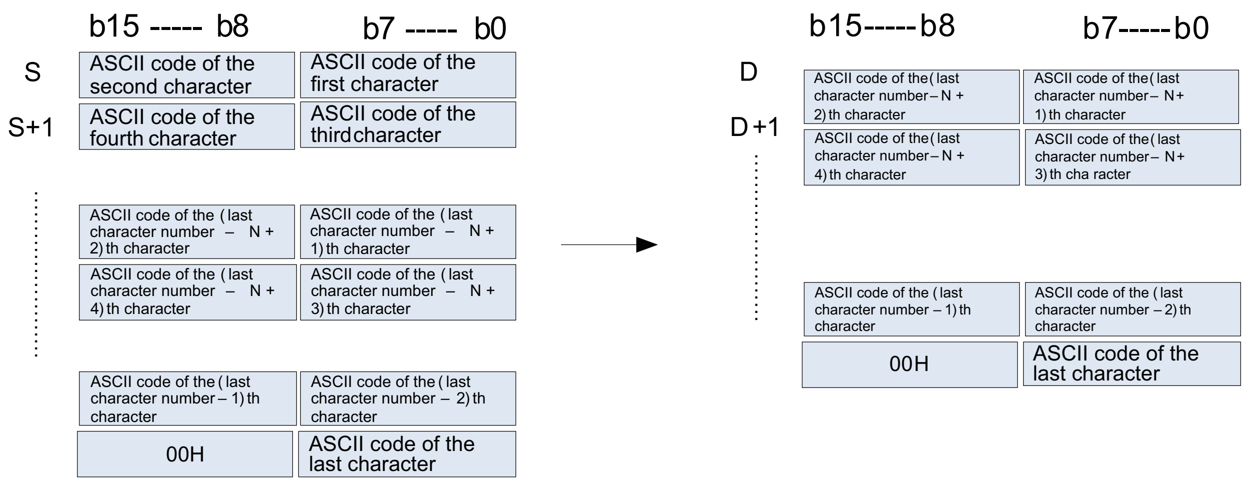
◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

n characters are retrieved at the right end of a string from head address [S]. The result is stored in elements from head address [D].



The retrieved string is appended with 00H when stored.

1. When the number of retrieved characters is odd, 00H is stored in the higher byte of the element that stores the last character.

2. When the number of retrieved characters is even, 0000H is stored in the element after the one that stores the last character.

3. When the number of retrieved bytes is 0, 0000H is stored in [D].

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

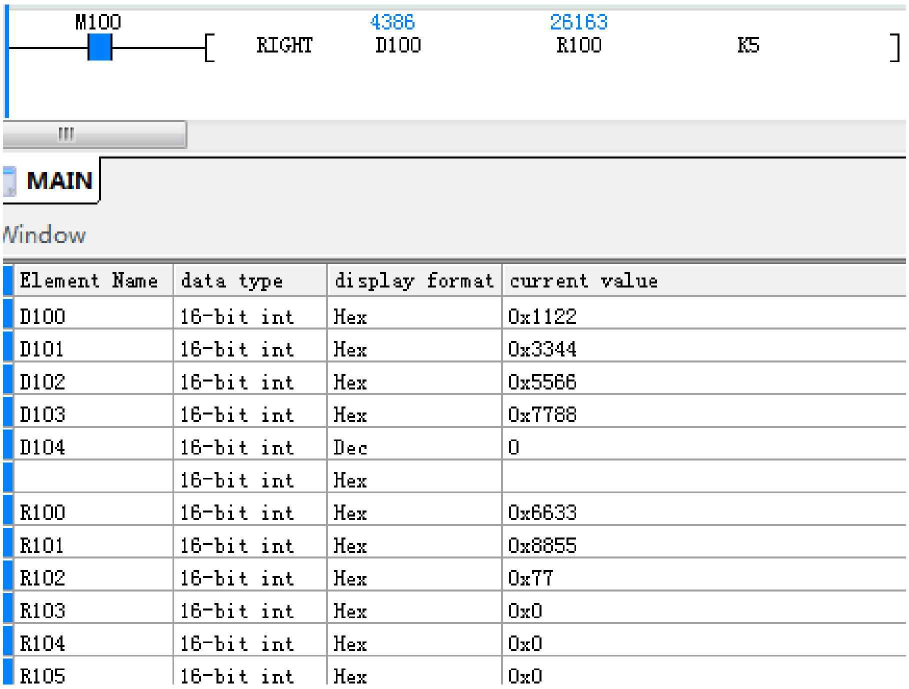
1. Error 6705 is returned due to out-of-range search when 00H is not found in elements from head address [S].

2. Error 6705 is returned when elements from head address [D] cannot fully store the n retrieved characters.

3. Error 6706 is returned when the value of n is greater than the number of characters stored in elements from head address [S].

4. Error 6706 is returned when n is negative.

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1122
D101	16-bit int	Hex	0x3344
D102	16-bit int	Hex	0x5566
D103	16-bit int	Hex	0x7788
D104	16-bit int	Dec	0
	16-bit int	Hex	
R100	16-bit int	Hex	0x6633
R101	16-bit int	Hex	0x8855
R102	16-bit int	Hex	0x77
R103	16-bit int	Hex	0x0
R104	16-bit int	Hex	0x0
R105	16-bit int	Hex	0x0

LEFT: String read left

◆ Overview

The LEFT instruction retrieves a designated number of characters at the left end of a string.

LEFT S D n		String read left	Applicable model: H3U	
S	Source data	Start number of elements that store a string	16-bit instruction (7 steps) LEFT: Continuous execution LEFTP: Pulse execution	
D	Retrieval result	Start number of elements that store a retrieved string		
n	Retrieved character count	Number of characters to be retrieved		

◆ Operands

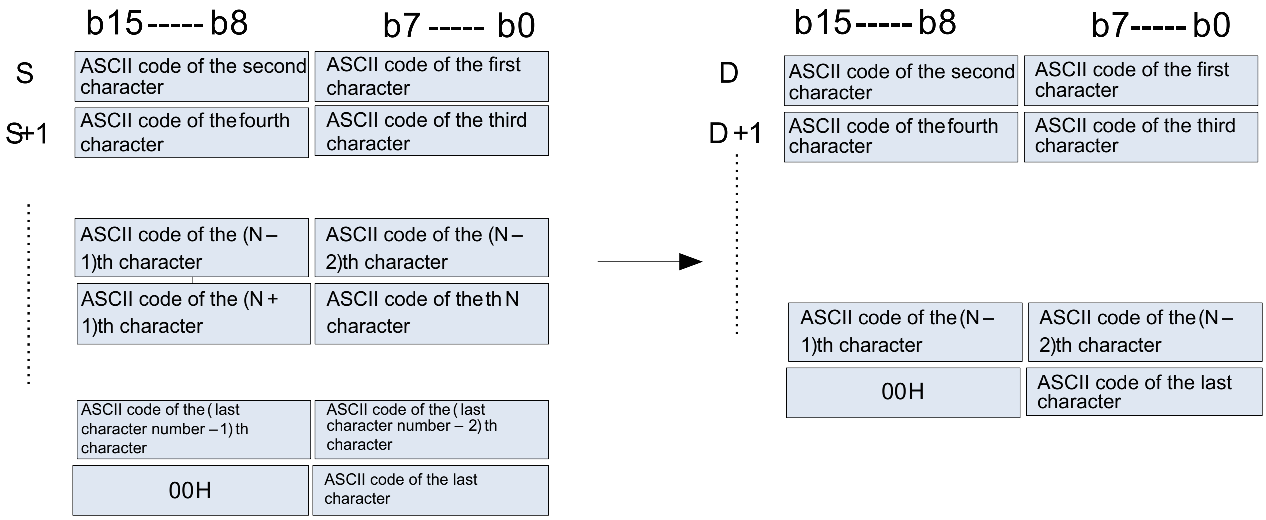
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

4

n characters are retrieved at the left end of a string from head address [S]. The result is stored in elements from head address [D]. The retrieved string is appended with 00H when stored.



1. When the number of retrieved characters is odd, 00H is stored in the higher byte of the element that stores the last character.

2. When the number of retrieved characters is even, 0000H is stored in the element after the one that stores the last character.

3. When the number of retrieved bytes is 0, 0000H is stored in [D].

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

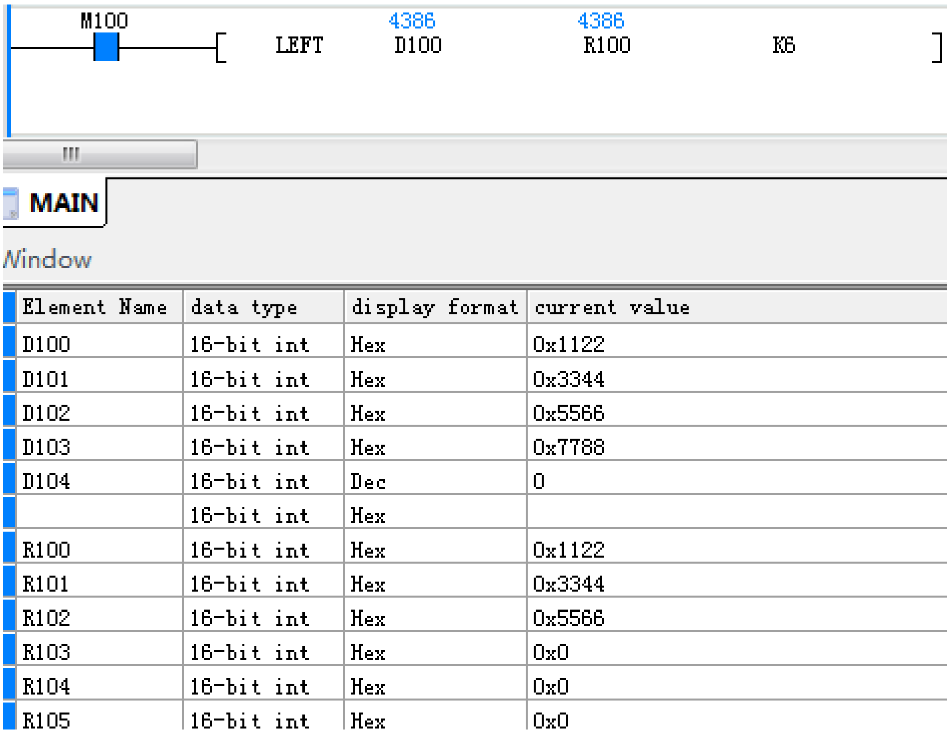
1. Error 6705 is returned due to out-of-range search when 00H is not found in elements from head address [S].

2. Error 6705 is returned when elements from head address [D] cannot fully store the n retrieved characters.

3. Error 6706 is returned when the value of n is greater than the number of characters stored in elements from head address [S].

4. Error 6706 is returned when n is negative.

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1122
D101	16-bit int	Hex	0x3344
D102	16-bit int	Hex	0x5566
D103	16-bit int	Hex	0x7788
D104	16-bit int	Dec	0
	16-bit int	Hex	
R100	16-bit int	Hex	0x1122
R101	16-bit int	Hex	0x3344
R102	16-bit int	Hex	0x5566
R103	16-bit int	Hex	0x0
R104	16-bit int	Hex	0x0
R105	16-bit int	Hex	0x0

MIDR: Reading of one string from another

◆ Overview

The MIDR instruction reads one string from another.

MIDR S1 D S2		Reading of one string from another	Applicable model: H3U
S1	Source data	Start number of elements that store a string	16-bit instruction (7 steps) MIDR: Continuous execution MIDRP: Pulse execution
D	Retrieval result	Start number of elements that store a retrieved string	
S2	Reading position	Start number of elements that indicate the start position of reading and the number of read characters S2: Position of the initial character to be read S2+1: Number of read characters	

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

A number (indicated by [S2+1]) of characters starting from the character indicated by [S2] are retrieved at the left end (beginning) of a string from head address [S1]. The result is stored in elements from head address [D].

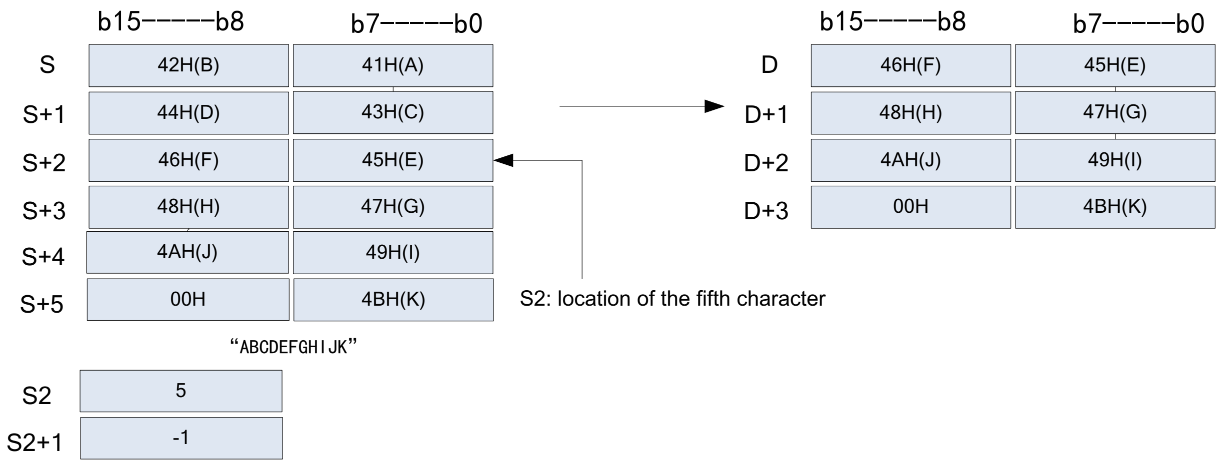
1. If [S2+1] is an odd number, 00H is stored in the higher byte of the element that stores the last character.

2. If [S2+1] is an even number, 0000H is stored in the element after the one that stores the last character.

The source string starts with [S1] and ends with 00H.

When the value in [S2+1] is 0, the instruction is not executed.

If the value in [S2+1] is –1, all data within the range from the character indicated by [S2] to the last character from head address [S1] is stored in elements from head address [D].



An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned due to out-of-range search when 00H is not found in elements from head address [S1].

2. Error 6706 is returned when the value in [S2] is greater than the number of characters from head address [S1].

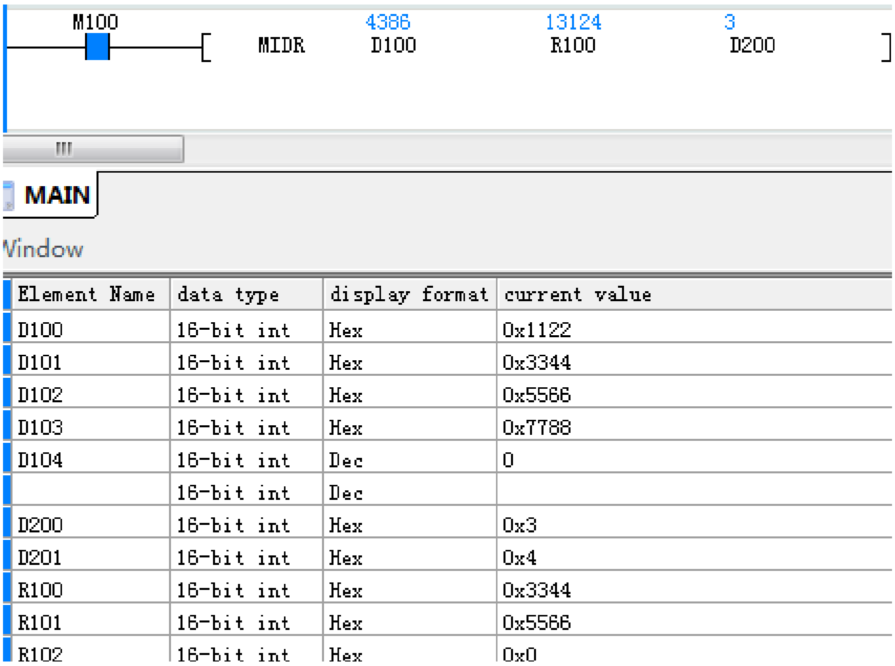
3. Error 6705 is returned when elements from head address [D] cannot fully store the read characters whose number is indicated by [S2+1].

4. Error 6706 is returned when the value of n in [S2] is negative.

5. Error 6706 is returned when the value in [S2+1] is –2 or less.

6. Error 6706 is returned when the value in [S2+1] exceeds the number of characters from head address [S1].

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1122
D101	16-bit int	Hex	0x3344
D102	16-bit int	Hex	0x5566
D103	16-bit int	Hex	0x7788
D104	16-bit int	Dec	0
	16-bit int	Dec	
D200	16-bit int	Hex	0x3
D201	16-bit int	Hex	0x4
R100	16-bit int	Hex	0x3344
R101	16-bit int	Hex	0x5566
R102	16-bit int	Hex	0x0

4

MIDW: String replacement at any position

◆ Overview

The MIDW instruction replaces one string with another contained in a designated string.

MIDW S1 D S2			String replacement at any position	Applicable model: H3U	
S1	Source data	Start number of elements that store the string that will replace another string		16-bit instruction (7 steps) MIDW: Continuous execution MIDWP: Pulse execution	
D	Replacement result	Start number of elements that store the string after replacement			
S2	Replacement position	Start number of elements that indicate the start position of replacement and the number of replaced characters S2: Initial character to be replaced S2+1: Number of replaced characters			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

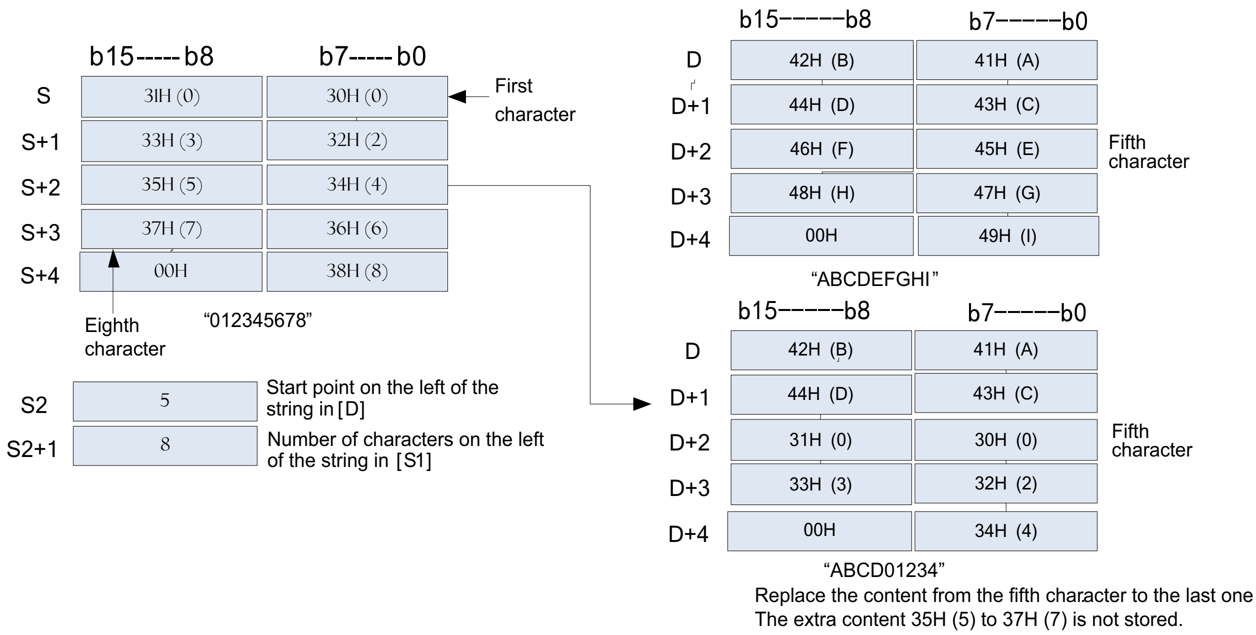
The character indicated by [S2] and subsequent characters contained in the string from head address [D] are replaced by a number (indicated by [S2+1]) of characters at the left end (beginning) of the string from head address [S1].

The source string starts with [S1] and ends with 00H.

When the value in [S2+1] is 0, the instruction is not executed.

When the value in [S2+1] is –1, all characters of the string from head address [S1] replace those of the string from head address [D].

If the value in [S2+1] is greater than the number of characters within the range from the character indicated by [S2] to the last one, only the characters within this range are replaced, and redundant characters of the source string are discarded.



An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned due to out-of-range search when 00H is not found in elements from head address [S1] or [D].

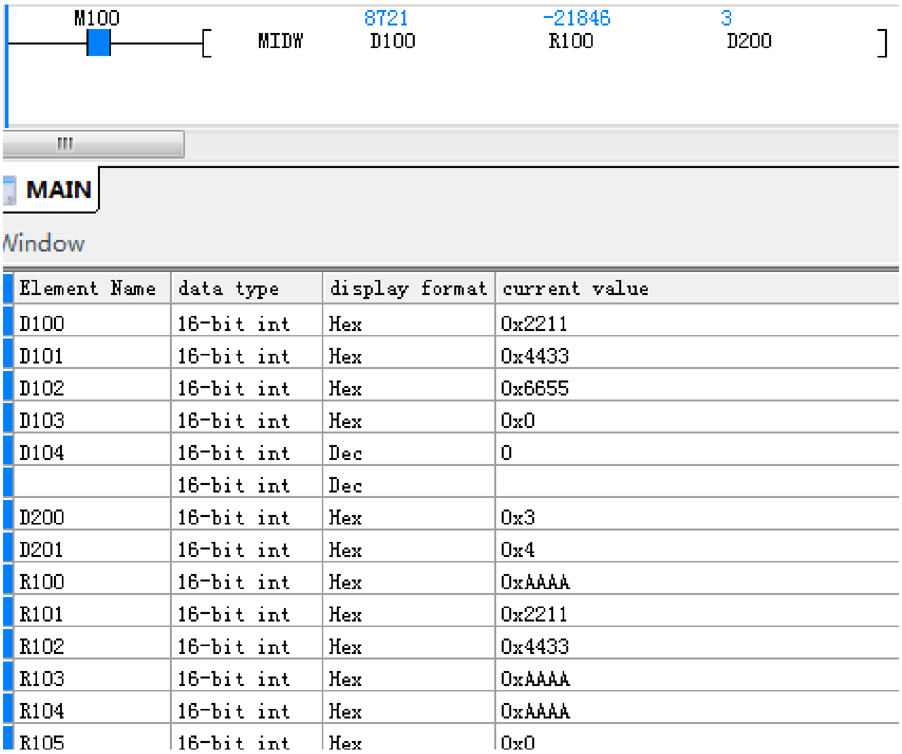
2. Error 6706 is returned when the value in [S2] is greater than the number of characters contained in the string from head address [D].

3. Error 6706 is returned when the value of n in [S2] is negative.

4. Error 6706 is returned when the value in [S2+1] is –2 or less.

5. Error 6706 is returned when the value in [S2+1] exceeds the number of characters from head address [S1].

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x2211
D101	16-bit int	Hex	0x4433
D102	16-bit int	Hex	0x6655
D103	16-bit int	Hex	0x0
D104	16-bit int	Dec	0
	16-bit int	Dec	
D200	16-bit int	Hex	0x3
D201	16-bit int	Hex	0x4
R100	16-bit int	Hex	0xAAAA
R101	16-bit int	Hex	0x2211
R102	16-bit int	Hex	0x4433
R103	16-bit int	Hex	0xAAAA
R104	16-bit int	Hex	0xAAAA
R105	16-bit int	Hex	0x0

$MOV: String transfer

◆ Overview

The $MOV instruction transfers strings.

$MOV S D			String transfer	Applicable model: H3U	
S	Source address	String (which contains a maximum of 32 characters) directly designated in the transfer source, or start number of elements that store the a string		16-bit instruction (5 steps) $MOV: Continuous execution $MOVP: Pulse execution	
D	Destination address	Start number of elements that store a string transferred to a destination			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The string from head address [S] is copied to elements from head address [D]. All characters of the string that starts from [S] and ends with the first 00H are transferred at a time, together with the terminator 00H or 0000H.

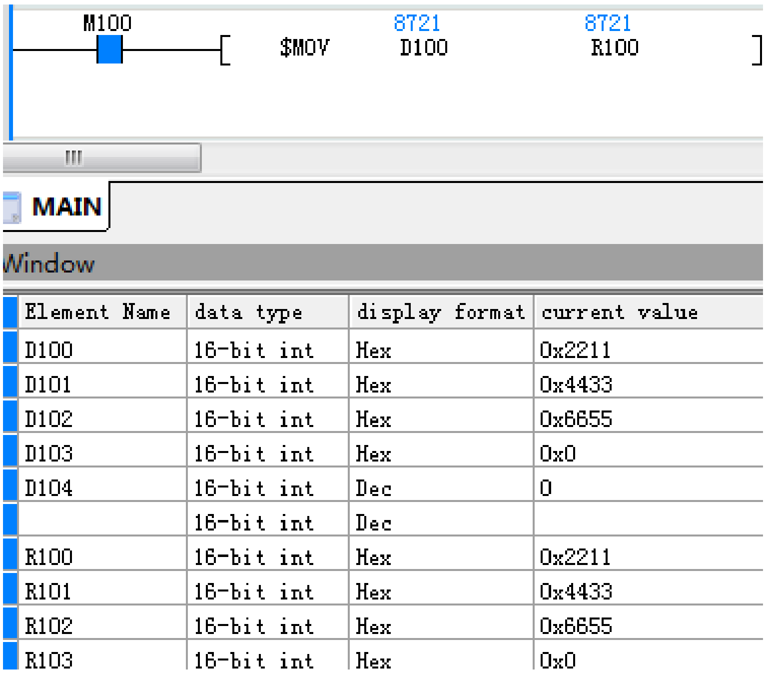
Batch copy is supported when the addresses of [S] and [D] overlap.

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

1. Error 6705 is returned due to out-of-range search when 00H is not found in elements from head address [S].

2. Error 6705 is returned when elements from head address [D] cannot fully store the transferred string.

◆ Application



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x2211
D101	16-bit int	Hex	0x4433
D102	16-bit int	Hex	0x6655
D103	16-bit int	Hex	0x0
D104	16-bit int	Dec	0
	16-bit int	Dec	
R100	16-bit int	Hex	0x2211
R101	16-bit int	Hex	0x4433
R102	16-bit int	Hex	0x6655
R103	16-bit int	Hex	0x0

4.8 Clock Instructions

Clock comparison output	TCMP	Clock data comparison
	TZCP	Clock data range comparison
Clock operations	TADD	Clock data addition operation
	TSUB	Clock data subtraction operation
Clock conversion	HTOS	Conversion from **hours:minutes:seconds** format to seconds
	STOH	Conversion from seconds to **hours:minutes:seconds** format
Clock read/write	TRD	Clock data read
	TWR	Clock data write
Timing	HOUR	Hour meter
	TTMR	Teaching timer
	STMR	Special timer
	DUTY	Timing pulse generation

4.8.1 Clock Comparison Output

TCMP: Clock data comparison

◆ Overview

The TCMP instruction compares the specified time (**hours:minutes:seconds**) with the time of an internal real-time clock and outputs the comparison result.

TCMP S1 S2 S3 S D			Clock data comparison	Applicable model: H3U
S1	Hours	Hours of the time to be compared, in the range 0 to 23	16-bit instruction (11 steps) TCMP: Continuous execution TCMPP: Pulse execution	
S2	Minutes	Minutes of the time to be compared, in the range 0 to 59		
S3	Seconds	Seconds of the time to be compared, in the range 0 to 59		
S	PLC time data head address	Head address of registers that store the current time value of a real-time clock, that is, the data read by the TRD or MOV instruction		
D	Comparison result	Head address of three consecutive variable units that store the comparison result		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The specified time (**hours:minutes:seconds**) is compared with the time of an internal real-time clock. The comparison result is output.

S1 represents hours, in the range 0 to 23.

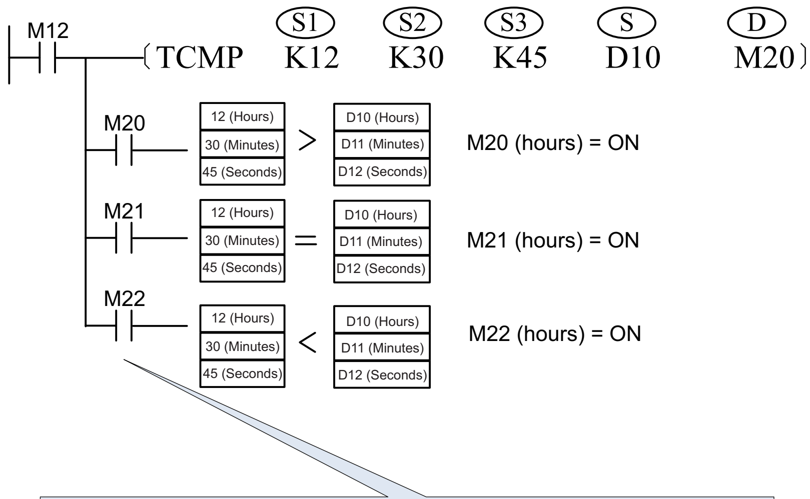
S2 represents minutes, in the range 0 to 59.

S3 represents seconds, in the range 0 to 59.

S is the head address of registers that store the current time value of a real-time clock, that is, the data read by the TRD or MOV instruction.

D is the head address of three consecutive variable units that store the comparison result.

Example:



When M12 = ON, then M20, M21, or M22 is set to ON.
The TCMP instruction is not executed when M12 switches from ON to OFF, and M20 to M22 remain in the state prior to M12 = OFF. Use the RST or ZRST instruction to clear the comparison result of M20 to M22.
Connect M20 to M22 in serial or parallel mode to acquire the ≥, ≤, and ≠ results.

TZCP: Clock data range comparison

◆ Overview

The TZCP instruction compares the time of an internal real-time clock with a specified time range. The result is stored in three consecutive variable units.

TZCP S1 S2 S D			**Clock data range comparison**	**Applicable model: H3U**
S1	Lower limit	Lower limit of a time range, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units	16-bit instruction (9 steps) TZCP: Continuous execution TZCPP: Pulse execution	
S2	Upper limit	Upper limit of a time range, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units		
S	PLC time data head address	Head address of registers that store the current time value of a real-time clock, that is, the data read by the TRD or MOV instruction		
D	Comparison result	Head address of three consecutive variable units that store the comparison result		

◆ Operands

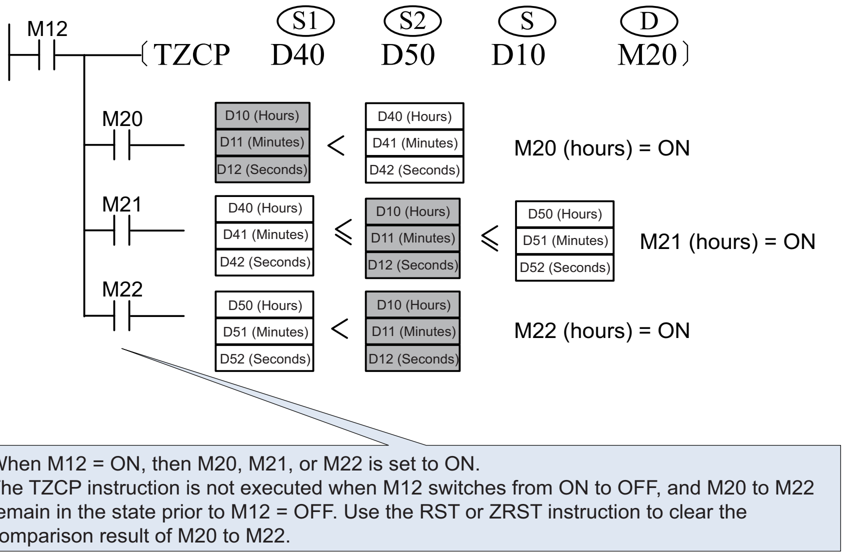
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The time of an internal real-time clock is compared with a time range defined by two time values in **hours:minutes:seconds** format. The result is stored in three consecutive variable units.

Example:



4.8.2 Clock Operations

TADD: Clock data addition operation

◆ Overview

The TADD instruction adds two time values in **hours:minutes:seconds** format together. The result is stored in designated variables.

TADD S1 S2 S3			Clock data addition operation	Applicable model: H3U
S1	Time augend	Time augend, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units	16-bit instruction (7 steps) TADD: Continuous execution TADDP: Pulse execution	
S2	Time addend	Time addend, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units		
D	Sum of two time values	Sum of two time values, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

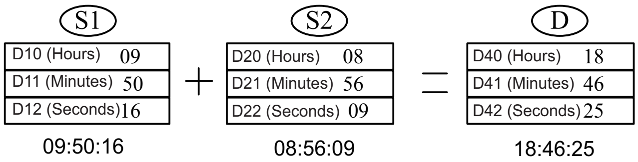
Two time values in **hours:minutes:seconds** format are added together. The result is stored in designated variables.

If the addition of two time values results in a value greater than 24 hours, the carry flag M8022 is set to ON and the actually displayed time is equal to the addition result minus 24:00:00.

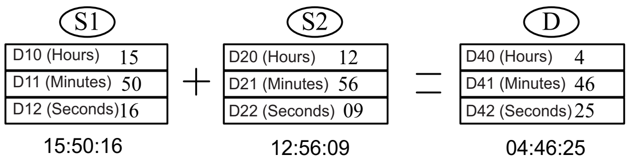
If the result is 00:00:00, the zero flag M8020 is set to 1.

Example:

Operation:



If the addition of two time values results in a value greater than 24 hours, the carry flag M8022 is set to ON.



TSUB: Clock data subtraction operation

◆ Overview

The TSUB instruction subtracts one time value from another in **hours:minutes:seconds** format. The result is stored in designated variables.

TSUB S1 S2 D		Clock data subtraction operation	Applicable model: H3U	
S1	Time subtrahend	Time subtrahend, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units	16-bit instruction (7 steps) TSUB: Continuous execution TSUBP: Pulse execution	
S2	Time minuend	Time minuend, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units		
D	Difference between two time values	Difference between two time values, which is represented by hours, minutes, and seconds respectively stored in three consecutive variable units		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

One time value is subtracted from another in **hours:minutes:seconds** format. The result is stored in designated variables.

If the subtraction of two time values results in a negative value, the borrow flag M8021 is set to ON and the actually displayed time is equal to the subtraction result plus 24:00:00.

If the subtraction gets 00:00:00, the zero flag M8020 is set to 1.

Example:

```
M11              (S1)  (S2)  (D)
─┤├──────(TSUB   D10   D20   D40 )
```

Operation:

S1		S2		D
D10 (Hours) 09	+	D20 (Hours) 08	=	D40 (Hours) 00
D11 (Minutes) 50		D21 (Minutes) 56		D41 (Minutes) 54
D12 (Seconds) 16		D22 (Seconds) 09		D42 (Seconds) 07
09:50:16		08:56:09		00:54:07

If the subtraction of two time values results in a negative value, the borrow flag M8021 is set to ON.

S1		S2		D
D10 (Hours) 09	+	D20 (Hours) 12	=	D40 (Hours) 20
D11 (Minutes) 50		D21 (Minutes) 56		D41 (Minutes) 54
D12 (Seconds) 16		D22 (Seconds) 09		D42 (Seconds) 07
09:50:16		12:56:09		20:54:07

4.8.3 Clock Conversion

HTOS: Conversion from hours:minutes:seconds format to seconds

◆ Overview

The HTOS instruction converts time values in **hours:minutes:seconds** format to seconds.

HTOS S D		**Conversion from hours:minutes:seconds format to seconds**	**Applicable model: H3U**	
S	Source data	Start number of elements that store the time in **hours:minutes:seconds** format to be converted	16-bit instruction (5 steps) HTOS: Continuous execution HTOSP: Pulse execution	32-bit instruction (9 steps) DHTOS: Continuous execution DHTOSP: Pulse execution
D	Result	Number of the element that stores the seconds converted from a time value		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit instruction

The time value in **hours:minutes:seconds** format stored in [S, S+1, S+2] is converted to seconds. The result is stored in D.

Hours' range: 0 to 9

Minutes' range: 0 to 59

Seconds' range: 0 to 59

2) 32-bit instruction

The time value in **hours:minutes:seconds** format stored in [S, S+1, S+2] is converted to seconds. The result is stored in [D, D+1].

Hours' range: 0 to 32,767

Minutes' range: 0 to 59

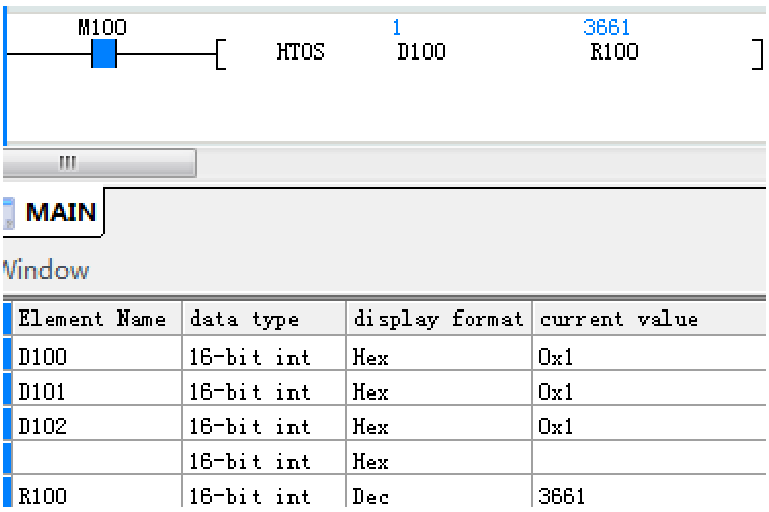
Seconds' range: 0 to 59

In 16- and 32-bit operations, an error occurs in the following conditions. The instruction is not executed and the error code is stored in D8067.

- Error 6705 is returned when the operands of the 16- and 32-bit instructions are out of range.
- Error 6706 is returned when the conversion result acquired by the 16-bit instruction is greater than 32,767.
- Error 6706 is returned when the time value in [S, S+1, S+2] is out of range.

◆ Application

The time value in **hours:minutes:seconds** format stored in D100, D101, and D102 is converted to seconds. The result is stored in R100.



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1
D101	16-bit int	Hex	0x1
D102	16-bit int	Hex	0x1
	16-bit int	Hex	
R100	16-bit int	Dec	3661

STOH: Conversion from seconds to hours:minutes:seconds format

◆ Overview

The STOH instruction converts time values represented in seconds to the **hours:minutes:seconds** format.

STOH S D		Conversion from seconds to hours:minutes:seconds format	Applicable model: H3U	
S	Source data	Number of the element that stores a time value represented in seconds	16-bit instruction (5 steps) STOH: Continuous execution STOHP: Pulse execution	32-bit instruction (9 steps) DSTOH: Continuous execution DSTOHP: Pulse execution
D	Result	Start number of elements that store the time in **hours:minutes:seconds** format after conversion		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) 16-bit Instruction

The time value represented in seconds stored in [S] is converted to the **hours:minutes:seconds** format. The result is stored in [D, D+1, D+2].

The value in [S] ranges from 0 to 32,767.

2) 32-bit Instruction

The time value represented in seconds stored in [S, S+1] is converted to the **hours:minutes:seconds** format. The result is stored in [D, D+1, D+2].

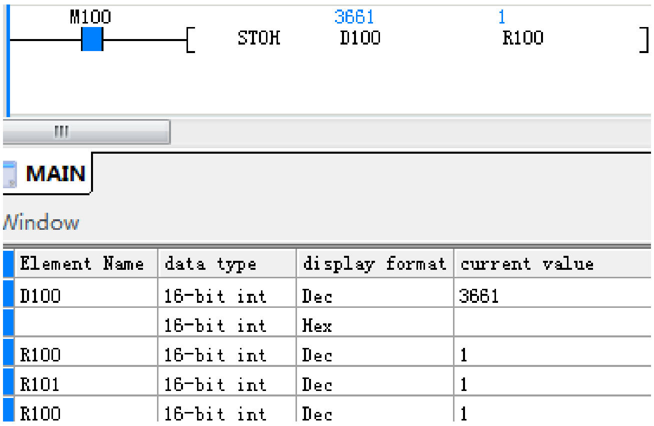
The value in [S, S+1] ranges from 0 to 117,964,799.

In 16- and 32-bit operations, an error occurs in the following conditions. The instruction is not executed and the error code is stored in D8067.

- Error 6705 is returned when the operands are out of range.
- Error 6706 is returned when the number of seconds to be converted is out of range.

◆ Application

The seconds in D100 are converted to the **hours:minutes:seconds** format. The result is stored in R100, R101, and R102.



4

4.8.4 Clock Read/Write

TRD: Clock data read

◆ Overview

The TRD instruction reads the current time and date of the internal real-time clock of the PLC. The time and date include the year, month, date, hours, minutes, seconds, and day, which are stored in designated registers.

TRD D			Clock data read	Applicable model: H3U
D	Time storage head address	Head address of seven consecutive variable units that store the year, month, date, hours, minutes, seconds, and day from the smallest to largest addresses	16-bit instruction (3 steps) TRD: Continuous execution TRDP: Pulse execution	

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

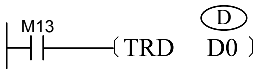
Note: The elements in gray background are supported.

◆ Function

The current time and date of the internal real-time clock of the PLC are read and stored in designated registers. The time and date include the year, month, date, hours, minutes, seconds, and day.

D is the head address of seven consecutive variable units that store the year, month, date, hours, minutes, seconds, and day from the smallest to largest addresses.

Example:



Operation:

Item	System Variable		D After Operation
Year (2000–2099)	D8018	→	D0
Month (1–12)	D8017	→	D1
Date (1–31)	D8016	→	D2
Hours (0–23)	D8015	→	D3
Minutes (0–59)	D8014	→	D4
Seconds (0–59)	D8013	→	D5
Day (0–6: Sun–Sat)	D8019	→	D6

Note: When you need to use the internal real-time clock of the PLC is used, use the TDR instruction to read the current time and date of the clock and store them in registers from head address D. Do not use the data stored in D8013 to D8019 directly.

TWR: Clock data write

◆ Overview

The TWR instruction sets the internal real-time clock of the PLC to a time value represented in year, month, date, hours, minutes, seconds, and day, which are stored from head address S.

<table>
<tr><th colspan="3">TWR D</th><th>Clock data write</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S</td><td>Time and date storage head address</td><td colspan="2">Head address of seven consecutive variable units that store the year, month, date, hours, minutes, seconds, and day from the smallest to largest addresses</td><td>16-bit instruction (3 steps)
TWR: Continuous execution
TWRP: Pulse execution</td><td></td></tr>
</table>

◆ Operands

<table>
<tr><th rowspan="2">Operand</th><th colspan="7">Bit Element</th><th colspan="15">Word Element</th></tr>
<tr><th colspan="7">System·User</th><th colspan="5">System·User</th><th colspan="5">Bit Designation</th><th colspan="2">Indexed Address</th><th colspan="2">Constant</th><th>Real Number</th></tr>
<tr><td>S</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modification</td><td>K</td><td>H</td><td>E</td></tr>
</table>

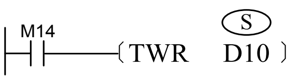
Note: The elements in gray background are supported.

◆ Function

The internal real-time clock of the PLC is set to a time value represented in year, month, date, hours, minutes, seconds, and day, which are stored from the head address S.

S is the head address of seven consecutive variable units that store the year, month, date, hours, minutes, seconds, and day from the smallest to largest addresses.

Example 1:



Operation:

Item	Data Source **D**		System Variable
Year (2000–2099)	D0	→	D8018
Month (1–12)	D1	→	D8017
Date (1–31)	D2	→	D8016
Hours (0–23)	D3	→	D8015
Minutes (0–59)	D4	→	D8014
Seconds (0–59)	D5	→	D8013
Day (0–6: Sun–Sat)	D6	→	D8019

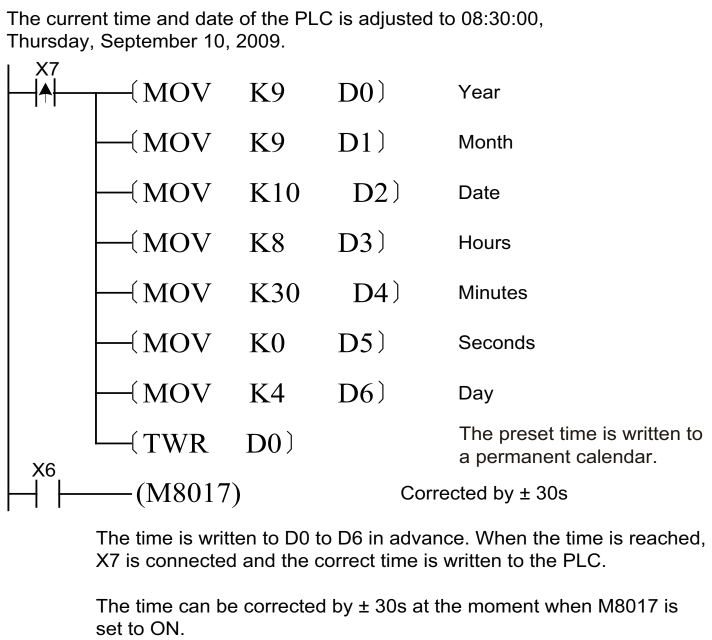
All the seven data entries year, month, date, hours, minutes, seconds, and day are used to set a new current value of the real-time clock. All the corresponding variables must be preset. If the day is not set, the default value 0 (Sunday) is applied. If the month is not set, then the corresponding variable is 0 and the PLC considers it to be incorrect, resulting in invalid clock change.

The clock is calibrated by ±30s each time M8017 is set to ON. When the number of seconds of the clock is within the range 1 to 29, the seconds are cleared while the minutes remain unchanged; when the number of seconds is within the range 30 to 59, the seconds are cleared while the number of minute is incremented by 1.

The clock stops when M8015 is set to ON.

The clock calibration method is described as follows.

Example 2:



Note: When you need to modify the clock, use the TWR instruction to write clock data to D8013–D8019. M8015 must be set when the MOV instruction is used to assign values to D8013–D8019.

4.8.5 Timing

HOUR: Hour meter

◆ Overview

When driving conditions are met, the HOUR instruction records time cumulatively. When the cumulative time reaches the preset value, a designated output becomes active.

HOUR S D1 D2		Hour meter	Applicable model: H3U	
S	Preset time	Preset time, measured in hours. When the cumulative time reaches the preset one, a designated output becomes active.	16-bit instruction (7 steps) HOUR: Continuous execution	32-bit instruction (13 steps) DHOURP: Pulse execution
D1	Cumulative time	Start number of units that store the cumulative time		
D2	Time reached flag	Variable unit that outputs a time reached alert. When the cumulative time reaches the preset value, this unit is active.		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

When driving conditions are met, the HOUR instruction records time cumulatively. When the cumulative time reaches the preset value, a designated output becomes active.

S is the preset time, measured in hours. When the cumulative time reaches the preset value, a designated output becomes active.

D1 is the start number of units that store the cumulative time.

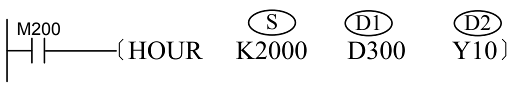
D2 is the variable unit that outputs a time reached alert. When the cumulative time reaches the preset value, this unit is active.

In 16-bit operation, the value in D1 ranges from K0 to K32,767, in the unit of hours. D1+1 stores the current time value less than 1 hour. The value ranges from K0 to K3599, in the unit of seconds. D1 occupies two units.

In 32-bit operation, the value stored in D1+1 and D1 ranges from K0 to K2,147,483,647, in the unit of hours. D1+2 stores the current time value less than 1 hour. The value ranges from K0 to K3599, in the unit of seconds. D1 occupies three units.

The time value in D1 cannot be negative. If D1 is specified as a register without support for retention upon power failure, the value in D1 is cleared when the PLC switches from STOP to RUN or when a power failure occurs. If you need to retain the current data in the case of a power failure, specify D1 as a register with support for retention upon power failure.

Example:



When M200 = ON, the time during which M200 remains ON is recorded cumulatively and stored in D300. If the time value is less than 1 hour, the equivalent value in the unit of seconds is recorded in D301. When the cumulative time in D200 reaches 2000 hours, Y10 output is set to ON. After the cumulative time counted from when the timing condition is met reaches the preset value in S, the cumulative time continues to increase. Timing stops when the current time value in D300 reaches 32,767 hours or the value in D301 reaches 3599s. To restart timing, clear the values in D300 and D301.

TTMR: Teaching timer

◆ Overview

The TTMR instruction measures the duration when driving conditions remain ON.

TTMR D n			Teaching timer	Applicable model: H3U	
D	Measured value	Duration when driving conditions remain ON		16-bit instruction (5 steps) TTMR: Continuous execution	
n	Measurement unit	Seconds when n = 0; 100 ms when n = 1; 10 ms when n = 2			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
D	X	Y	M	T	C	S	SM	**D**	**R**	T	C	**SD**	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modification**	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	**K**	**H**	E

Note: The elements in gray background are supported.

◆ Function

The key hold time of a designated input port is multiplied by n. The result is stored in the D variable and typically used for parameter setting.

D stores the product of multiplying the key hold time (in seconds) by n. The content of D remains unchanged after key release. D+1 stores the key press time and is reset to 0 after key release. The time in D+1 is measured in 100 ms.

n is the multiple. The actual multiple is calculated by 10n (n in the range 0 to 2).

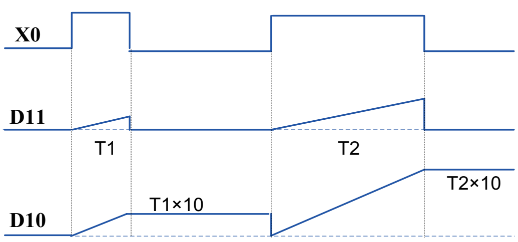
When n = K0, the actual multiple is x1.

When n = K1, the actual multiple is x10.

When n = K2, the actual multiple is x100.

Example 1:

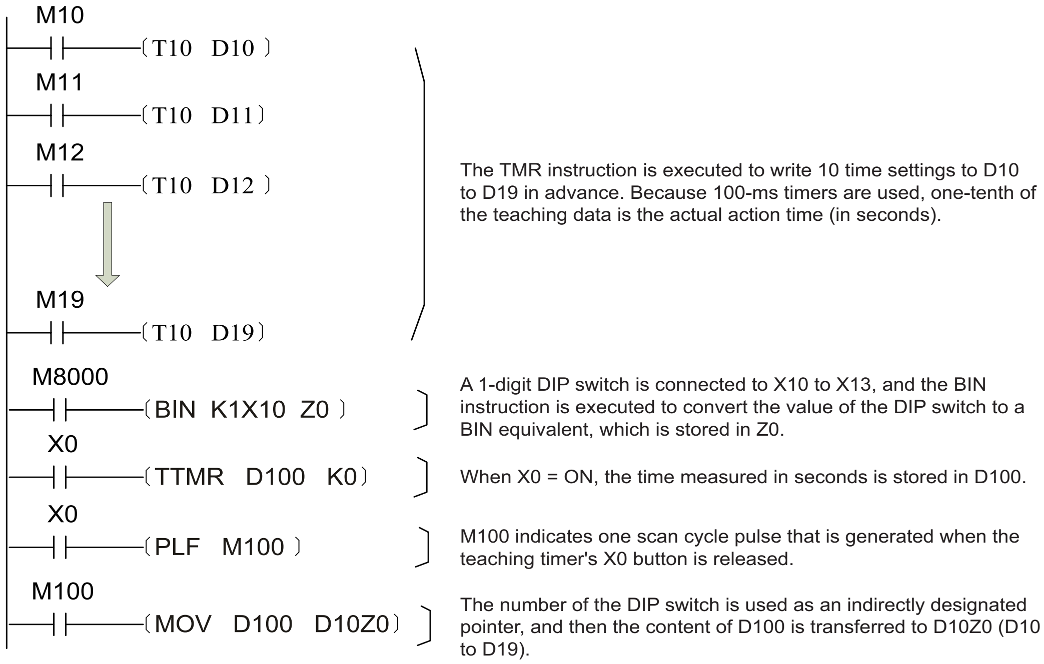
D10 = D11 when X10 is enabled. After X10 is disabled, the value in D10 remains unchanged whereas that in D11 changes to 0.



Assume that the key hold time of X10 is T seconds. The following table lists the relationships among D10, D11, and n.

n	D10	D11 (unit: 100 ms)
K0 (unit: s)	1 x T	D11 = D10 x 10
K1 (unit: 100 ms)	10 x T	D11 = D10
K2 (unit: 10 ms)	100 x T	D11 = D10/10

Example 2:



STMR: Special timer

◆ Overview

When driving conditions are met, the STMR instruction acquires four contacts with reference to the time value S2 of the timer S1, including delayed switch-off after power-off, switch-on upon power-off and delayed switch-off, switch-on upon power-on and delayed switch-off, and delayed switch-on after power-on and delayed switch-off after power-off.

STMR S m D			Special timer	Applicable model: H3U	
S	Timer	Timer number, in the range T0 to T199		16-bit instruction (7 steps) STMR: Continuous execution	
m	Preset time	Preset time of the timer, in the unit of 100 ms; value range: 1 to 32,767			
D	Contact head address	Head address of four consecutive elements that store contacts			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
m	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Four delay actions are generated based on the instruction flow.

S stores the number of the timer used for generating delay actions. Timers T0 to T199 are available.

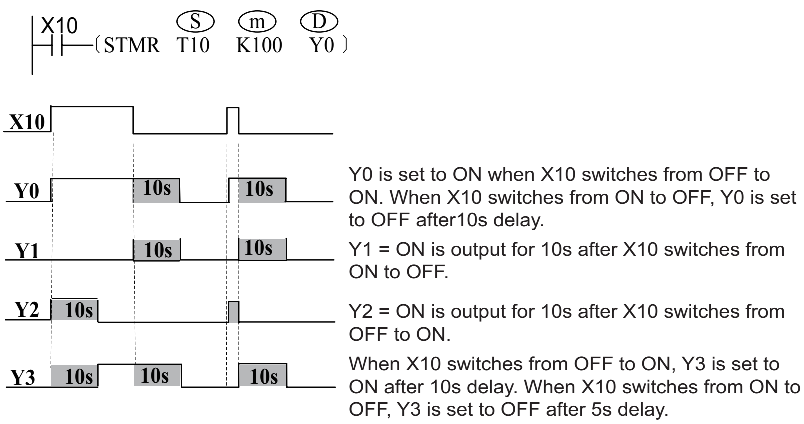
m is the preset delay value, in the unit of 100 ms. The value range is K1 to K32,767.

D is the start number of four consecutive elements that output the delay actions.

- Note:

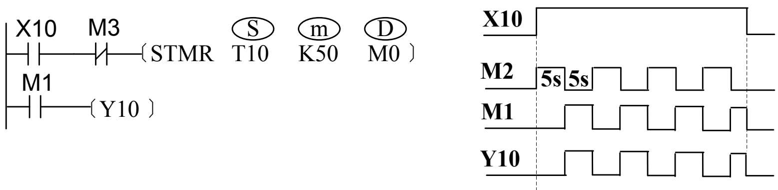
The timer used by this instruction and the corresponding outputs cannot be reused by other instructions.

◆ Example 1:



◆ Example 2:

Introduce a D element to the instruction flow for convenient output of an oscillator. (The ALT instruction achieves the same result.)



4

DUTY: Timing pulse generation

◆ Overview

The DUTY instruction considers the period required to complete a specified number of operations as one cycle and generates timed signals.

DUTY S1 S2 D			Timing pulse generation	Applicable model: H3U
S1	ON cycle count	Number of scan (operation) cycles mapping the ON state (n1 ≥ 0)	16-bit instruction (7 steps) DUTY: Continuous execution	
S2	OFF cycle count	Number of scan (operation) cycles mapping the OFF state (n2 ≥ 0)		
D	Output	Destination address (M8335–M8339) to which a timer outputs data		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1. Timers output the ON (after n1 scan cycles) and OFF (after n2 scan cycles) states.

2. The destination address that stores timer outputs is M8335–M8339. The scan cycle counting result is stored in D8335–D8339.

3. The value in D8335–D8339 is reset when it is equal to (n1 + n2) or the input flow switches from OFF to ON again.

The switch-off input flow cannot be stopped during execution.

4. When n1 = 0 and n2 ≥ 0, the value in [D] is fixed to OFF; when n1 > 0 and n2 = 0, the value in [D] is fixed to ON.

- An operation error occurs in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

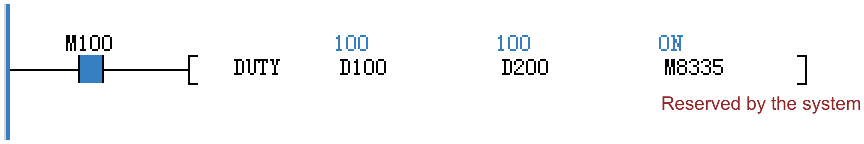
1. Error 6706 is returned when n1 and n2 are out of range.

2. Error 6705 is returned when the destination address [D] is beyond the range M8335–M8339.

3. Error 6711 is returned when multiple DUTY instructions use the same timer output destination address.

◆ Application

The following square wave results in a duty cycle of 1:1 when the M8335 output cycle is equal to 200 scan cycles.



4.9 High-speed Input, Pulse Positioning, and Communication Positioning

High-speed comparison	HSCS	(High-speed counter) Comparison setting
	HSCR	(High-speed counter) Comparison reset
	HSZ	(High-speed counter) Range comparison
	HSOS	High-speed interrupt comparison setting
	HSOR	High-speed interrupt comparison reset
	HSTP	High-speed comparative interrupt output
Pulse input	SPD	Pulse density detection
Pulse output	PWM	Pulse-width modulation output
	PLSY	Pulse output
	PLSR	Pulse output with acceleration/deceleration
Pulse positioning	PLSV	Variable-speed pulse output
	PLSV2	Variable-speed pulse output with acceleration/deceleration
	PLSN	Multi-speed pulse output
	DVIT	Interrupt positioning (extension)
	DRVI	Relative position positioning
	DRVA	Absolute position positioning
	ZRN	Zero return
	DSZR	DOG search zero return
Communication positioning (manipulator)	AXISENAB	Axis enabling
	AXISSTOP	Axis positioning stop
	AXISESTOP	Axis emergency stop (used for stopping the servo in case of exceptions)
	AXISDRVA	Absolute axis positioning
	AXISZRN	Axis zero return
	AXISJOGA	Axis jog
	AXISALMRST	Axis alarm reset
Refreshing	REF	Input/Output refreshing
	REFF	Input filter time adjustment

4

4.9.1 High-speed Comparison

High-speed comparison	HSCS	(High-speed counter) Comparison setting [Note]
	HSCR	(High-speed counter) Comparison reset [Note]
	HSZ	(High-speed counter) Range comparison [Note]
	HSOS	High-speed interrupt comparison setting (only applicable to the H3U-PM motion control model)
	HSOR	High-speed interrupt comparison reset (only applicable to the H3U-PM motion control model)
	HSTP	High-speed comparative interrupt output (only applicable to the H3U-PM motion control model)

Note: The usage of the HSCS, HSCR, and HSZ instructions varies between the H3U standard models and H3U-PM motion control model. For details, see "Chapter 5 High-speed Input" on Page 336

4.9.2 Pulse Input

Pulse input	SPD	Pulse density detection

For details, see "Chapter 5 High-speed Input" on Page 336.

4.9.3 Pulse Output

Pulse output	PWM	Pulse-width modulation output
	PLSY	Pulse output
	PLSR	Pulse output with acceleration/deceleration

For details, see "Chapter 6 Positioning and Interpolation" on Page 375.

4.9.4 Pulse Positioning

Pulse positioning	PLSV	Variable-speed pulse output
	PLSV2	Variable-speed pulse output with acceleration/deceleration
	PLSN	Multi-speed pulse output
	DVIT	Interrupt positioning (extension)
	DRVI	Relative position positioning
	DRVA	Absolute position positioning
	ZRN	Zero return
	DSZR	DOG search zero return

For details, see "Chapter 6 Positioning and Interpolation" on Page 375.

4.9.5 Communication Positioning

Communication positioning (manipulator)	AXISENAB	Axis enabling
	AXISSTOP	Axis positioning stop
	AXISESTOP	Axis emergency stop (used for stopping the servo in case of exceptions)
	AXISDRVA	Absolute axis positioning
	AXISZRN	Axis zero return
	AXISJOGA	Axis jog
	AXISALMRST	Axis alarm reset

1) Note:

The H3U integrated axis control (manipulator) instructions are executed based on CANlink3.0+IS620P communication. (The H3U version must be later than 24303-0000. In AutoShop, H3U-R must be selected for the PLC type.)

- Manually set the baud rate and station number of the servo driver.
- Manually set servo stop parameters.
- Manually set the home attaining mode of the servo driver and connect the corresponding proximity switch to the servo driver.
- These instructions occupy the special elements SM and SD of H3U. CANlink configuration occupies elements after D7200 and M7200.
- SM400 controls the data format of instruction parameters: ON indicates the floating-point number format whereas OFF indicates the integer format. The position and speed of axis control (manipulator) instructions must be given in the preset format. The current axis position and speed are displayed in the preset format. By default, SM400 is OFF.
- If the floating-point number format is used, the proportionality coefficient must be set for mechanical parameters. If the integer format is used, this setting is not required.
- The default format is integer. You can switch to the floating-point number format in system setting when necessary.
- The axis control (manipulator) instructions are not recommended in SFC.
- When the servo generates an alarm, the error flag of the corresponding instruction is set and the instruction is not executed.

2) Special registers

The following table lists the special registers used by H3U.

Element	Function	Remarks
SM400	Flag of instruction parameter format switching	OFF by default. It can be modified only during the initial running cycle. ON: The instruction parameters are in floating-point number format. OFF: The instruction parameters are in integer format.
M7200–M7679	Occupied by CANlink configuration	-
D7200–D7999	Occupied by CANlink configuration	-

4

Axis Number	Current Axis Position Display (Floating-point Number/Integer)	Current Axis Speed Display (Floating-point Number/Integer)	Setting of Axis Positioning Deviation Pulse Count (Integer)	---	Pulse Count Corresponding to Unit Mechanical Displacement (Floating-point Number)	Servo Rotational Speed Corresponding to Unit Mechanical Speed (Floating-point Number)
Axis 1	SD410, 411	SD412, 413	SD414	SD415	SD416, 417	SD418, 419
Axis 2	SD420, 421	SD422, 423	SD424	SD425	SD426, 427	SD428, 429
Axis 3	SD430, 431	SD432, 433	SD434	SD435	SD436, 437	SD438, 439
Axis 4	SD440, 441	SD442, 443	SD444	SD445	SD446, 447	SD448, 449
Axis 5	SD450, 451	SD452, 453	SD454	SD455	SD456, 457	SD458, 459
Axis 6	SD460, 461	SD462, 463	SD464	SD465	SD466, 467	SD468, 469
Axis 7	SD470, 471	SD472, 473	SD474	SD475	SD476, 477	SD478, 479
Axis 8	SD480, 481	SD482, 483	SD484	SD485	SD486, 487	SD488, 489
	...	...	...	...	...	...
Axis 16	SD560, 561	SD562, 563	SD564	SD565	SD566, 567	SD568, 569

Note:

1. Examples of unit mechanical displacement: 1 mm, 1°, and 1 radian.

2. Unit mechanical speeds can be measured in mm/s and revolution/min, which corresponds to the RPM unit of the servo.

3) Servo parameter setting

The following table lists the servo parameters that need to be set.

Servo Parameter	Value	Description
H02-00	1	Control mode: Position mode
H03-10	0	DI5 terminal function: 0
H05-00	2	Primary position instruction: Multi-segment position
H0C-09	1	VDI for virtual communication: Enabled
H0C-13	0 [Note]	Parameter number writing to EEPROM: Disabled
H0C-15	0	CAN communication protocol: CANlink
H11-00	1	Multi-segment position running mode: Cyclic running
H11-01	1	Segment quantity selection: One segment
H11-04	1	Displacement instruction type: Absolute position
H11-05	1	Start segment of cyclic running
H17-00	1	VDI1 terminal function
H17-02	18	VDI2 terminal function
H17-04	19	VDI3 terminal function
H17-06	28	VDI4 terminal function
H17-08	32	VDI5 terminal function
H17-10	34	VDI6 terminal function
H17-12	2	VDI7 terminal function
H0C-00	Depending on the actual condition	Axis number
H0C-08		Baud rate

Effective VDI		
H31-00	Bit Function	Description
Bit 0	S-ON	Enabled
Bit 1	JOGCMD+	Forward jog
Bit 2	JOGCMD-	Reverse jog
Bit 3	PosInSen	Multi-segment enabled
Bit 4	HomingStart	Home attaining
Bit 5	EmergencyStop	Emergency stop
Bit 6	ALM-RST	Alarm reset

Note: The parameter setting operation varies with different setting methods (panel setting and communication setting).

- When the panel setting method is used, servo parameters can be set to 0.
- When the communication setting method is used, set H0C-13 to 1 first. (The values of parameters except those in groups H0B and H0D are saved to EEPROM in real time.) After you complete the setting of other parameters, set H0C-13 to 0; otherwise, the setting of other parameters cannot be saved.

AXISENAB: Axis enabling

◆ Overview

The AXISENAB instruction enables axes based on CANlink.

AXISENAB S1			**Servo enabling**	**Applicable model:** H3U
S1	Axis number	Servo station number, in the range 1 to 16	16-bit instruction (3 steps) AXISENAB: Continuous execution	

◆ Operands

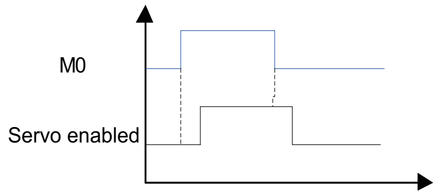
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ **Function**

The servo that corresponds to a designated station or axis number is enabled when the flow is active. The servo can be kept enabled during normal use. The axis number must be an immediate value.

◆ **Example**



Servo 1 is disabled when M0 = 0; it is enabled when M0 = 1.

NOTE

- This instruction can be executed only once for every axis.

AXISSTOP: Axis positioning stop

◆ **Overview**

The AXISSTOP instruction stops servo positioning based on CANlink.

AXISSTOP S1			**Servo positioning stop**	**Applicable model:** H3U	
S1	Axis number	Servo station number, in the range 1 to 16		16-bit instruction (3 steps) AXISSTOP: Continuous execution	

◆ **Operands**

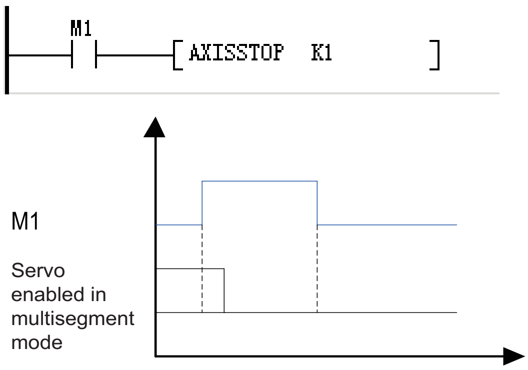
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ **Function**

The servo that corresponds to a designated station or axis number stops positioning based on the acceleration or deceleration time during positioning. This instruction can be executed when an exception occurs during positioning. The axis number must be an immediate value.

- Example



When M1 = 1, servo 1 stops positioning. (The internal multi-segment enabling function of the servo is disabled.)

NOTE

- This instruction can be executed only once for every axis.

AXISESTOP: Axis emergency stop (used for stopping the servo in case of exceptions)

◆ Overview

The AXISENAB instruction implements an emergency stop based on CANlink.

AXISESTOP S1			Servo emergency stop	Applicable model: H3U
S1	Axis number	Servo station number, in the range 1 to 16	16-bit instruction (3 steps) AXISESTOP: Continuous execution	

◆ Operands

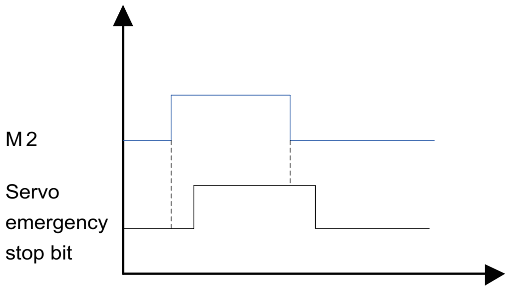
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Function

The servo that corresponds to a designated station or axis number implements an emergency stop. This instruction can be executed when an exception occurs during positioning. The axis number must be an immediate value.

The stop mode must be set on the servo driver.

◆ Example



When M2 = 0, servo 1 disables emergency stop; when M2 = 1, servo 1 enables emergency stop. (The internal emergency stop bit of the servo is driven.)

NOTE

- This instruction can be executed only once for every axis.

AXISDRVA: Absolute axis positioning

◆ Overview

The AXISDRVA instruction implements absolute axis positioning based on CANlink.

AXISDRVA S1 S2 S3 S4 S5 S6		Absolute axis positioning	Applicable model: H3U	
S1	Axis number	Servo station number, in the range 1 to 16	16-bit instruction (13 steps) AXISDRVA: Continuous execution	
S2	Position	Absolute position (which occupies two elements)		
S3	Speed	Positioning speed (which occupies four elements and cannot be the same for different instructions)		
S4	Acceleration/ Deceleration time	Acceleration/Deceleration time during positioning		
S5	Complete flag	Positioning complete flag		
S6	Error flag	Instruction error flag		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S5	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S6	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

4

◆ Function

① Axis number: The value ranges from K1 to K16. A maximum of 16 axes are supported. The servo station number must be set to the corresponding axis number. The axis number must be an immediate value.

② Position: In the actual condition, the pulse equivalent is sent to the servo when parameters are in integer or floating-point number format (which format to use is determined by the SM400 flag). If the integer format is used, the pulse unit is specified directly. For example, 1000 indicates 1000 pulses. If the floating-point number format is used, the mechanical unit is specified, and the proportion between mechanical displacement and pulse unit and that between mechanical speed and servo rotational speed must be set. If the customer's mechanical structure is configured with a correspondence between 1000 pulses and 1 mm feeding amount, the input of a floating-point number in the unit of 1.00 mm to this instruction results in the output of 1000 pulses. For details about the proportionality coefficient, see "A.2 Special Soft Element Register Range" on Page 704. Position data can be monitored at any time during running.

③ Speed: in integer or floating-point number format. For details about the units and conversion, see the preceding description. This operand occupies four consecutive word elements, two for the parameter and the other for the instruction state machine. If the speed is set to 0, the default value 200 is written.

④ Complete flag: Check this element after startup to determine whether positioning is completed. In normal cases, the positioning instructions with the same station number can be executed after positioning.

⑤ Error flag: It is set when an error occurs in the following conditions:

The driver is not enabled. (The AXISENAB instruction is disabled.)

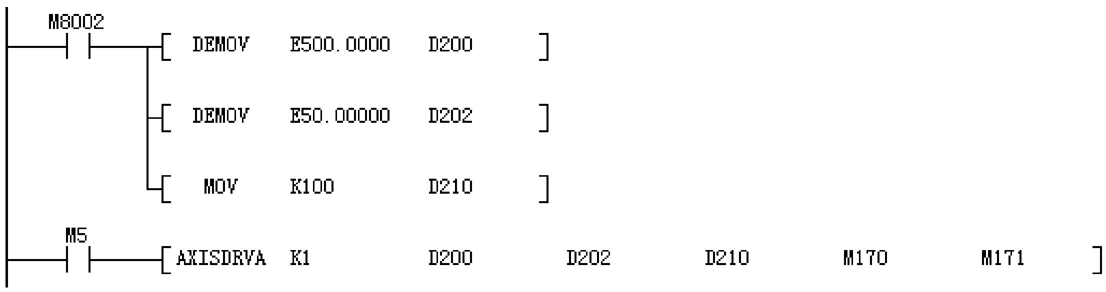
The speed or position is not written successfully.

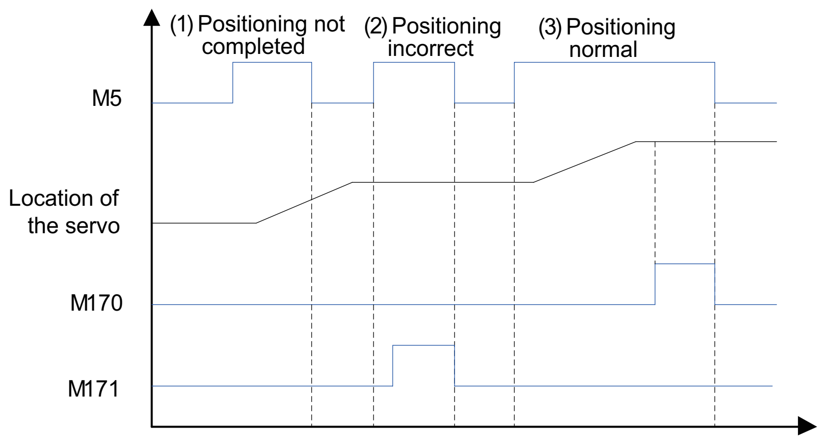
The driver generates an alarm.

Instructions conflict with each other. (The positioning, jog, positioning stop, and emergency stop instructions cannot be triggered at the same time.)

The complete flag and error flag are reset after the enable signal breaks off.

◆ Example





K1: Axis number

D200: Position

D202: Speed

D210: Acceleration/Deceleration time

M170: Positioning completed

M171: Positioning incorrect

1) If M5 is disabled during positioning, the servo continues movement toward a destination, but the complete flag M170 is not set.
2) If a servo data write error occurs when the instruction is executed, the servo stops movement and the error flag M171 is set. M171 is reset after M5 is disabled.
3) M170 is set after positioning is completed. It is reset when M5 is disabled.

NOTE

- The instruction can be called multiple times for the same axis, but the element that stores the speed parameter cannot be the same during each call.

AXISZRN: Axis zero return

◆ Overview

The AXISZRN instruction performs axis home attaining based on CANlink.

AXISZRN S1 S2 S3 S4		Servo zero return	Applicable model: H3U	
S1	Axis number	Servo station number, in the range 1 to 16	16-bit instruction (9 steps) AXISZRN: Continuous execution	
S2	Position offset	Origin position offset (which occupies four elements and cannot be the same for different instructions)		
S3	Complete flag	Completion of home attaining		
S4	Error flag	Instruction error flag		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Function

① Axis number: Number of the axis that requires home attaining, in the range K1 to K16.

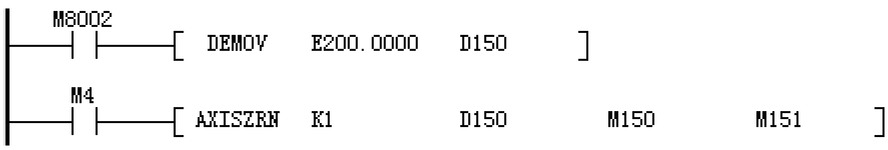
② Origin position offset: It can be set when necessary. It is typically set to 0 and stored in a D or R element. Speed: in integer or floating-point number format. For details about the units and conversion, see "AXISDRVA: Absolute axis positioning" on Page 288. This operand occupies four consecutive word elements, two for the parameter and the other for the instruction state machine.

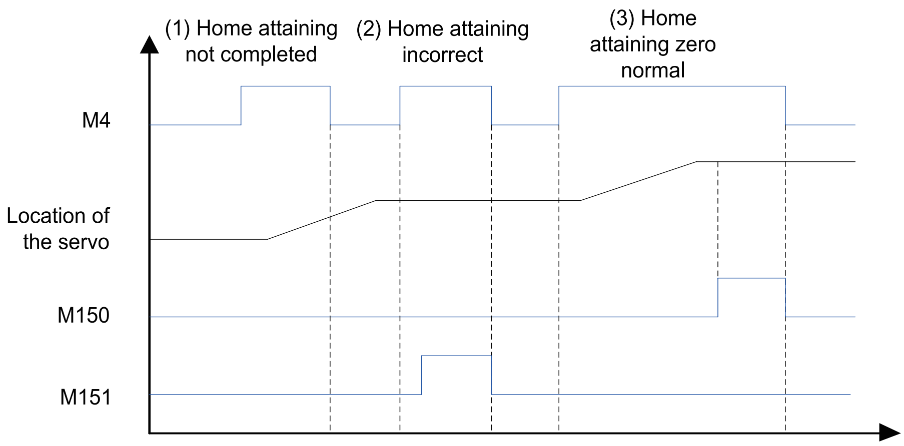
③ Complete flag: Indicates that home attaining is completed. It is stored in an S or M element.

④ Error flag: It is set when an error occurs in the following conditions:

- The driver is not enabled. (The AXISENAB instruction is disabled.)
- The position offset is not written successfully.
- The driver generates an alarm.
- Instructions conflict with each other. (The positioning, jog, positioning stop, and emergency stop instructions cannot be triggered at the same time.)
- The complete flag and error flag are reset after the enable signal breaks off.

◆ Example





K1: Axis number

D150: Servo origin offset, which is converted to the parameter number H05-36

M150: Positioning completed

M151: Positioning incorrect

When M4 = 0, home attaining is disabled for servo 1; when M4 = 1, home attaining is enabled for servo 1.

1) If M4 is disabled during positioning, the servo continues movement toward a destination, but the complete flag M150 is not set.

2) If a servo data write error occurs when the instruction is executed, the servo stops movement and the error flag M151 is set. M151 is reset after M4 is disabled.

3) M150 is set after positioning is completed. It is reset when M4 is disabled.

NOTE

- This instruction can be executed only once for every axis.

AXISJOGA: Axis jog

◆ Overview

The AXISJOGA instruction implements axis jog based on CANlink.

AXISJOGA S1 S2 S3 S4 S5		Axis jog	Applicable model: H3U	
S1	Axis number	Servo station number, in the range 1 to 16	16-bit instruction (11 steps) AXISJOGA: Continuous execution	
S2	Speed	Jog speed (which occupies four elements and cannot be the same for different instructions)		
S3	Forward jog	Jog in the forward direction		
S4	Reverse jog	Jog in the reverse direction		
S5	Error flag	Instruction error flag		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S5	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Function

① Axis number: Axis number of the servo that requires jog control.

② Speed: It is in integer or floating-point number format and stored in a D or R element. For details about the units and conversion, see "AXISDRVA: Absolute axis positioning" on Page 288. This operand occupies four consecutive word elements, two for the parameter and the other for the instruction state machine. If the speed is set to 0, the default value 100 is written.

③ Forward jog: It is stored in an M or S element. When the instruction is enabled and this bit is ON, the servo starts jog in the forward direction. When this bit is changed to OFF, the servo stops jog in the forward direction.

④ Reverse jog: It is stored in an M or S element. When the instruction is enabled and this bit is ON, the servo starts jog in the reverse direction. When this bit is changed to OFF, the servo stops jog in the reverse direction.

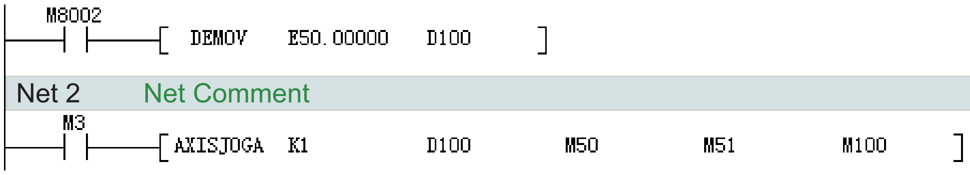
⑤ Error flag: It is set when an error occurs in the following conditions (the error flag of the AXISJOG instruction is stored in S4+1):

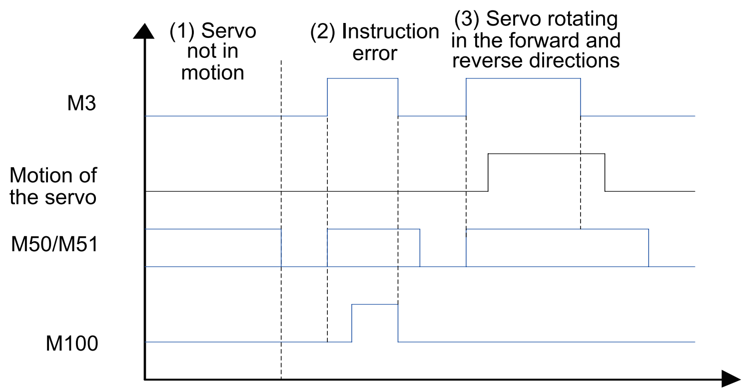
- The driver is not enabled. (The AXISENAB instruction is disabled.)
- The speed or position is not written successfully.
- The driver generates an alarm.
- Instructions conflict with each other. (The positioning, jog, positioning stop, and emergency stop instructions cannot be triggered at the same time.)
- The complete flag and error flag are reset after the enable signal breaks off.

Note 1: No action is taken when forward jog and reverse jog are both enabled.

Note 2: The speed is written only once when the instruction is enabled.

◆ Example





K1: Axis number

D100: Jog speed

M50: Forward jog

M51: Reverse jog

M100: Instruction error

When M3 = 0, jog is disabled for servo 1; when M3 = 1, jog is enabled for servo 1.

1) When M3 is disabled, forward/reverse jog control does not take effect.
2) If a servo data write error occurs when the instruction is executed, the servo stops movement and the error flag M100 is set. M100 is reset after M3 is disabled.
3) When M50 or M51 is triggered, the servo triggers the corresponding action. If M50 and M51 are both triggered, the servo is inactive.

NOTE

- The AXISJOG instruction has the same usage as the AXISJOGA instruction.
- The 24303-0000 software version can only use the AXISJOG instruction.
- The AXISJOG and AXISJOGA instructions differ in the number of operands. The error flag of the AXISJOG instruction is stored in S4+1, whereas that of the AXISJOGA instruction is stored in S5.

AXISALMRST: Axis alarm reset

◆ Overview

The AXISALMRST instruction resets axis alarms based on CANlink.

AXISALMRST S1			Axis alarm reset	Applicable model: H3U	
S1	Axis number	Servo station number, in the range 1 to 16		16-bit instruction (3 steps) AXISALMRST: Continuous execution	

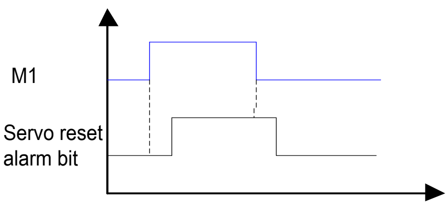
◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Function

The alarm of the servo corresponding to a designated station or axis number will be reset. Some servo alarms cannot be reset by this instruction. Use this instruction to reset an alarm after the corresponding fault is rectified. The axis number must be an immediate value.

◆ Example



When M1 = 1, alarm reset is enabled for servo 1. When M1 = 0, alarm reset is disabled for servo 1.

NOTE

- This instruction can be executed only once for every axis.
- This instruction is supported by 24304 and later versions.

4.9.6 Refreshing

Refreshing	REF	Input/Output refreshing
	REFF	Input filter time adjustment

REF: Input/Output refreshing

◆ Overview

The REF instruction refreshes the input or output image storage area immediately.

REF S n		Input/Output refreshing	Applicable model: H3U	
S	Bit element head address	Head address of input or output bit elements to be refreshed	16-bit instruction (5 steps) REF: Continuous execution REFP: Pulse execution	
n	Bit element count	Number of input or output bit elements to be refreshed		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The statuses of n elements from head address S are refreshed immediately. The following requirements are posed because of PLC's byte-based port access feature:

The address S must be a element number whose lowest bit is 0, such as X0, X10, ……Y0, and Y10.

The value of n must be a multiple of 8 (in the range 8 to 256).

In normal cases, the status of the input port X is read before a program executes scanning, and the status of the output port Y is refreshed in batches after scanning is completed (END is reached). This causes an I/O delay. Use the REF instruction if you need to input the latest information to applications or output operation results immediately.

- The REF instruction can be executed between the FOR and NEXT instructions and between CJ instructions.
- The REF instruction can be used to refresh the input and output of an interrupt subprogram to acquire the latest input information and promptly output operation results.
- The actual change delay of the input port status depends on the filter time of input elements. X0 to X7 have the digital filter function. The filter time is configurable (using the FNC51 REFF instruction) within the range 0 ms to 60 ms. Other I/O ports use hardware filter with a filter time of about 10 ms. For details about related parameters, see the **User Guide – H3U Series Programmable Logic Controller (PLC) – Higher Performance & Pulse Motion Control**. (Please visit http://www.inovance.cn/es to obtain the latest version.)
- The actual change delay of the output port status depends on the response time of output elements (for example, relays). The output contacts involved in output refresh will act after the response time of output relays (transistors) has elapsed. The response delay of the relay output type is about 10 ms (max.: 20 ms), that of the high-speed output ports of the transistor output type is about 10 μs, and that of the output ports of general points is about 0.5 ms. For details about related parameters, see the **User Guide – H3U Series Programmable Logic Controller (PLC) – Higher Performance & Pulse Motion Control**. (Please visit http://www.inovance.cn/es to obtain the latest version.)

◆ **Example 1:**

```
 X20
─┤├─(REF  X0  K16 )
```

When the preceding program is executed, if X20 = ON, the input statuses of X0 to X17 are read immediately and input signals are updated. No input delay is incurred.

◆ **Example 2:**

```
 X0
─┤├─(REF  Y0  K16 )
```

When the preceding program is executed, if X0 = ON, the statuses of Y0 to Y17 are refreshed and output signals are updated immediately, without waiting for the END instruction.

REFF: Input filter time adjustment

4

◆ Overview

When driving conditions are met, the REFF instruction sets the filter time constants of X0 to X7 input ports to n ms.

REFF n			Input filter time adjustment	Applicable model: H3U
n	Filter time	Unit: ms	16-bit instruction (5 steps) REFF: Continuous execution REFFP: Pulse execution	

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

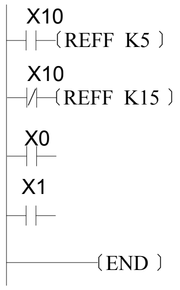
The filter time constants of X0 to X7 input ports are set to n ms.

In the PLC, X0 to X7 ports use digital filters. The default filter time constant is set by D8020. Use the REFF instruction to change the value in D8020 within the range 0 ms to 60 ms. Other X ports support only hardware RC filter. The filter time constant is about 10 ms, which cannot be modified.

When a high-speed counter is used or the X input interrupt function is enabled, the filter time of related ports is automatically adjusted to the shortest time, whereas the filter time of unrelated ports remains unchanged.

You can also use the MOV instruction to assign a value to D8020; then the filter time is changed.

◆ Example



When X10 = ON, the input filter time of X0 to X7 is set to 5 ms. When X10 = OFF, the input filter time of X0 to X7 is set to 15 ms.

4.10 Motion Control

H3U model interpolation	G90G01	2-axis linear absolute position interpolation
	G91G01	2-axis linear relative position interpolation
	G90G02	2-axis forward-arc absolute position interpolation
	G91G02	2-axis forward-arc relative position interpolation
	G90G03	2-axis reverse-arc absolute position interpolation
	G91G03	2-axis reverse-arc relative position interpolation
MC of the PM model	For details, see "Chapter 7 Motion Control" on Page 466	
G-code of the PM model	For details, see "Chapter 7 Motion Control" on Page 466	

4.10.1 H3U Model Interpolation

H3U model interpolation	G90G01	2-axis linear absolute position interpolation
	G91G01	2-axis linear relative position interpolation
	G90G02	2-axis forward-arc absolute position interpolation
	G91G02	2-axis forward-arc relative position interpolation
	G90G03	2-axis reverse-arc absolute position interpolation
	G91G03	2-axis reverse-arc relative position interpolation

The preceding instructions support only 32-bit operation. The pulse execution type is not supported.

For details, see "6.3 Interpolation Instruction" on Page 430.

4.10.2 MC of the PM Model

MC of the PM model	For details, see "Chapter 7 Motion Control" on Page 466.

4.10.3 G-code of the PM Model

G-code of the PM model	For details, see "Chapter 7 Motion Control" on Page 466.

4.11 Communication

Communication instructions	RS	Serial data transfer (For details, see "9.4 Modbus Protocol" on Page 598 and "9.5 Modbus Configuration and Usage" on Page 613.)
	MODBUS	Modbus communication (For details, see section "9.4 Modbus Protocol" on Page 598 and "9.5 Modbus Configuration and Usage" on Page 613.)
Verification	CCD	Check code
	CRC	CRC code calculation
	LRC	LRC code calculation

4

4.11.1 Communication Instructions

RS: Serial data transfer

◆ **Overview**

The RS instruction is used for data sending/reception during communication. The data of a designated register area is automatically sent to serial ports in sequence, and then the data received by serial ports is stored in a designated area. This achieves the effect of allowing the user program to directly access the communication buffer. Communication can be conducted using custom protocols by allowing the user program to process the sent/received data buffer.

RS S m D n			Serial data transfer	Applicable model: H3U
S	Data sending head address	Head address of the register area that stores the data to be sent	16-bit instruction (9 steps) RS: Continuous execution	
m	Sent data length	Length of the data to be sent, measured in bytes; value range: 0 to 256		
D	Data reception head address	Head address of the register area that stores the received data		
n	Received data length	Length of the received data, measured in bytes; value range: 0 to 256		

◆ **Operands**

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	**D**	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modification**	K	H	E
m	X	**Y**	M	T	C	S	SM	**D**	**R**	**T**	**C**	**SD**	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	**K**	**H**	E
D	X	Y	M	T	C	S	SM	**D**	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modification**	K	H	E
n	X	**Y**	M	T	C	S	SM	**D**	**R**	**T**	**C**	**SD**	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	**K**	**H**	E

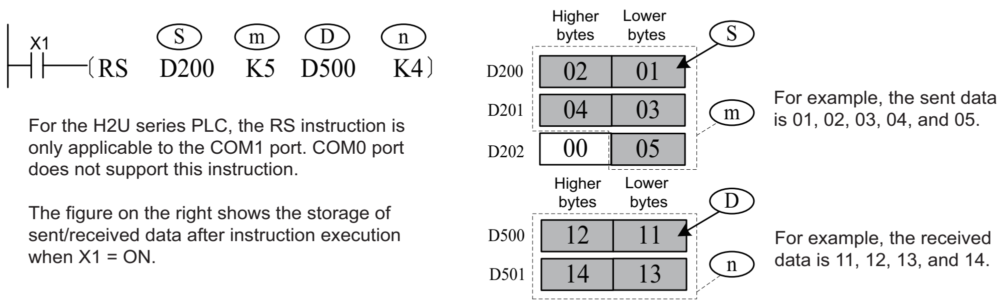
Note: The elements in gray background are supported.

◆ Function

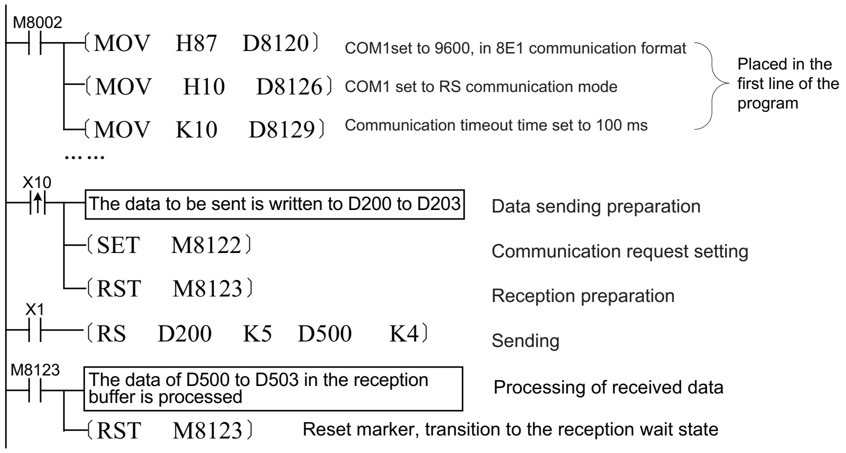
The RS instruction is used for data sending/reception during communication. The data of a designated register area is automatically sent to serial ports in sequence, and then the data received by serial ports is stored in a designated area. This achieves the effect of allowing the user program to directly access the communication buffer. Communication can be conducted using custom protocols by allowing the user program to process the sent/received data buffer.

Whether the RS instruction works in half-duplex or full-duplex mode is determined by bit 10 of D8120. Multiple RS instructions can be compiled by the user program, but only one instruction can be driven at a time. M8122 must be set before the RS instruction is driven.

◆ Example



Configuration and preparation of serial communication must be completed during programming so that communication can be conducted based on requirements. Related tasks include configuring the data sending/receiving mode of serial ports, baud rate, bits, parity bit, software protocols, and timeout conditions; preparing data for the sent/received data buffer; processing sending/reception labels. The preceding statement is used as an example. A complete RS communication configuration program is as follows:



4

MODBUS: Modbus communication

◆ Overview

The MODBUS instruction reads and writes data during Modbus communication.

MODBUS S1 S2 n D			Modbus communication	Applicable model: H3U
S1	Communication address and parameter number	Slave address (higher byte) and communication command (lower byte, defined in the Modbus protocol)		16-bit instruction (9 steps) MODBUS: Continuous execution
S2	Slave data head address	Head address of slave registers that store the data to be accessed		
n	Data length	Length of the data to be read or written		
D	Master data head address	Head address of units that store the read or written data (The length of occupied address units is determined by n.)		

4

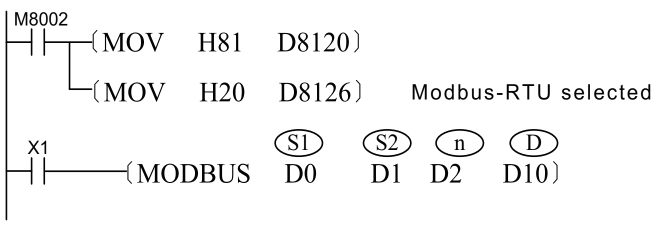
◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

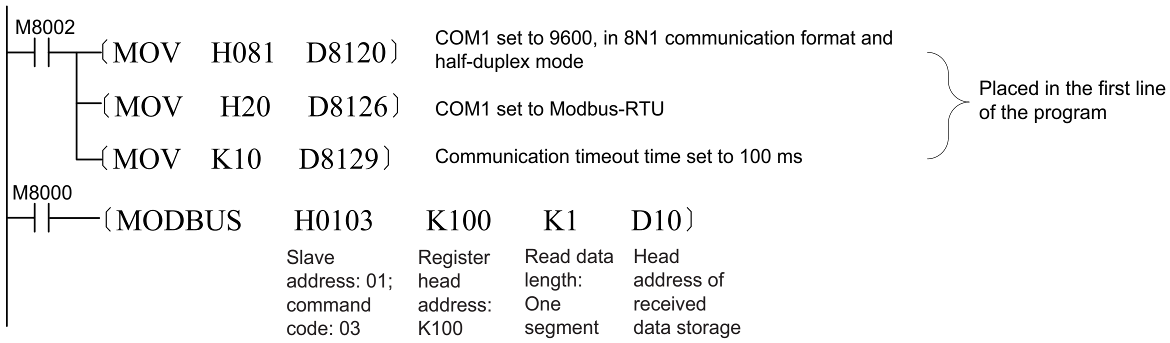
Note: The elements in gray background are supported.

◆ Function

For more intuitive and convenient operation, the RS instruction can be replaced with the MODBUS instruction during communication through a standard Modbus master station protocol.



Example 1:



The MODBUS instruction is used for the communication of Modbus-enabled slave devices (for example, MD320, MD300, and MD280 series AC drives). In example 1, the PLC continuously reads data from registers from head address K100 of slave 1 and stores the data in units from head address D10.

Usage:

Different from the RS instruction, multiple MODBUS instructions can be executed simultaneously in a program.

The MODBUS instruction does not require the processes of setting M8122 to ON and resetting M8123.

4.11.2 Verification

Verification	CCD	Check code
	CRC	CRC code calculation
	LRC	LRC code calculation

4

CCD: Check code

◆ Overview

When driving conditions are met, the CCD instruction calculates the checksum of the K data entries from head address S. The summation result is stored in D, and the XOR logical operation result is stored in D+1.

CCD S D n		Check code	Applicable model: H3U	
S	Data source	Head address of consecutive units that store the variables whose checksum will be calculated	16-bit instruction (7 steps) CCD: Continuous execution CCDP: Pulse execution	
D	Operation result	Summation result stored in D; XOR logical operation result stored in D+1		
n	Checked byte count	Number of bytes contained in checked variables		

◆ Operands

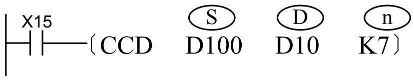
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Two types of checksum operation are performed on n variables from head address S. The summation result is stored in D, and the XOR logical operation result is stored in D+1. The string checksum operation ensures correct data transfer during communication.

- Example:

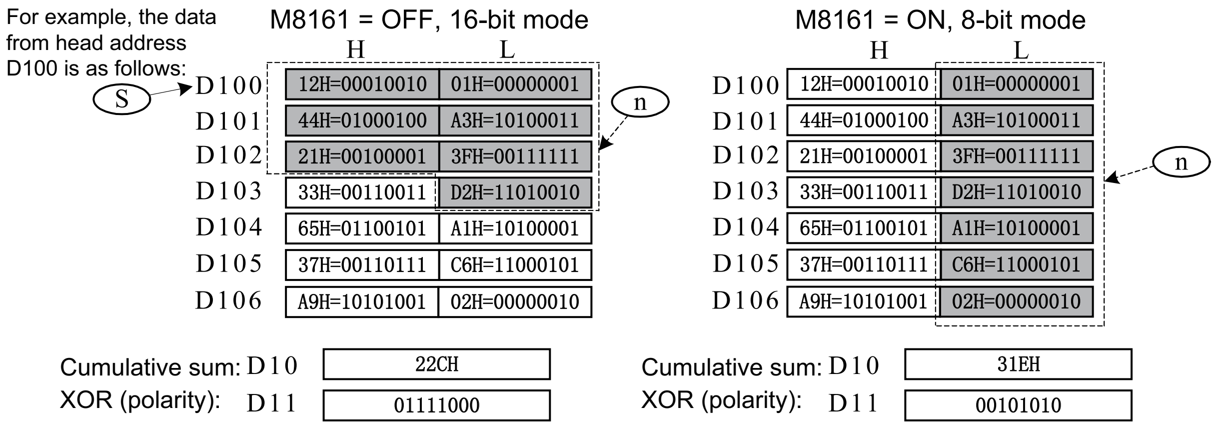


The M8161 flag determines which variable width mode to use. When M8161 = OFF, the 16-bit mode is enabled, whereby the higher and lower bytes of variables are taken for the operation. When M8161 = ON, the 8-bit mode is enabled, whereby only the lower bytes of variables are taken for the operation and the higher bytes are discarded. Therefore, the length of the actually used variable area is increased. See the following figure.

Summation is the process where the values of n variables are added together.

The XOR logical operation is described as follows:

1) The variables are converted to binary numbers.
2) Bit 0 = 1 occurrences in all variables are counted. If the counting result is an even number, the XOR operation result for bit 0 is 0; if the counting result is an odd number, the XOR operation result for bit 0 is 1.
3) Then bit 1 = 1 occurrences in all variables are counted. If the counting result is an even number, the XOR operation result for bit 1 is 0; if the counting result is an odd number, the XOR operation result for bit 1 is 1.
4) The counting proceeds to bit 2 through bit 7. The resulting binary number is converted to a hexadecimal equivalent, which is the XOR operation result (or called a polarity value).



The RS, HEX, ASCI, and CCD instructions share the M8161 flag. Pay attention to the flag processing during programming.

CRC: CRC code calculation

◆ Overview

A cyclic redundancy check (CRC) is a verification method used during communication. The CRC instruction is used to calculate the CRC code.

CRC S n D			CRC code calculation	Applicable model: H3U
S	Source data	Head address of elements that store the data for CRC code calculation	16-bit instruction (7 steps) CRC: Continuous execution CRCP: Pulse execution	
n	Data count	Number of operated data entries (K1 to K256)		
D	Result	Head address of elements that store the operation result		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

16-bit conversion mode: When M8161 = OFF, the higher eight bits and lower eight bits (n data points in total) from head address [S] are taken byte by byte for CRC code calculation. The result is stored in the higher eight bits and lower eight bits from head address [D].

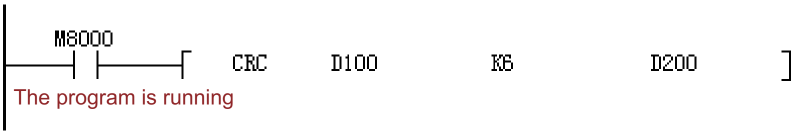
8-bit conversion mode: When M8161 = ON, the lower eight bits (n data points in total) from head address [S] are taken byte by byte for CRC code calculation. The lower eight bits of the result are stored in [D], and the higher eight bits are stored in [D+1].

An error is returned in the following condition. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

Error 6706 is returned when n is out of range.

◆ Application

M8161 = ON and the 8-bit conversion mode is enabled. The lower eight bits of elements D100 to D105 are taken for CRC code calculation. The result is stored in the lower eight bits of D200 and D201.



Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1
D101	16-bit int	Hex	0x2
D102	16-bit int	Hex	0x4
D103	16-bit int	Hex	0x20
D104	16-bit int	Hex	0x0
D105	16-bit int	Hex	0x12
	16-bit int	Dec	
M8161	BOOL	Bin	ON
D200	16-bit int	Hex	0xF8
D201	16-bit int	Hex	0xFD

M8161 = OFF and the 16-bit conversion mode is enabled. The lower eight bits of elements D100 to D105 are taken for CRC code calculation. The result is stored in the higher eight bits and lower eight bits of D200.

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x1
D101	16-bit int	Hex	0x2
D102	16-bit int	Hex	0x4
D103	16-bit int	Hex	0x20
D104	16-bit int	Hex	0x0
D105	16-bit int	Hex	0x12
	16-bit int	Dec	
M8161	BOOL	Bin	OFF
D200	16-bit int	Hex	0xB202
D201	16-bit int	Hex	0xFD

LRC: LRC code calculation

◆ Overview

The LRC instruction calculates the longitudinal redundancy check (LRC) code in ASCII mode.

LRC S n D			LRC code calculation	Applicable model: H3U	
S	Source data	Head address of elements that store the data for LRC code calculation		16-bit instruction (7 steps) LRC: Continuous execution LRCP: Pulse execution	
n	Data count	Number of operated data entries, which must be an even number; value range: K1 to K256			
D	Result	Register that stores the operation result			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

16-bit conversion mode: When M8161 = OFF, the higher eight bits and lower eight bits (n data points in total) from head address [S] are taken byte by byte for LRC code calculation. The result is stored in the higher eight bits and lower eight bits of [D].

8-bit conversion mode: When M8161 = ON, the lower eight bits (n data points in total) from head address [S] are taken byte by byte for LRC code calculation. The lower eight bits of the result are stored in [D], and the higher eight bits are stored in [D+1].

An error is returned in the following conditions. The error flag M8067 is set to ON to identify this error and the error code is stored in D8067.

- Error 6706 is returned when n is out of range.
- Error 6706 is returned when n is odd.

◆ Application

1) 16-bit mode (M8161 = OFF)

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x3130
D101	16-bit int	Hex	0x3330
D102	16-bit int	Hex	0x3132
D103	16-bit int	Hex	0x3230
D104	16-bit int	Hex	0x3030
D105	16-bit int	Hex	0x3230
	16-bit int	Dec	
M8161	BOOL	Bin	OFF
D200	16-bit int	Hex	0x3734

4

2) 8-bit mode (M8161 = ON)

Element Name	data type	display format	current value
D100	16-bit int	Hex	0x30
D101	16-bit int	Hex	0x31
D102	16-bit int	Hex	0x30
D103	16-bit int	Hex	0x33
D104	16-bit int	Hex	0x32
D105	16-bit int	Hex	0x31
D106	16-bit int	Hex	0x30
D107	16-bit int	Hex	0x32
D108	16-bit int	Hex	0x30
D109	16-bit int	Hex	0x30
D110	16-bit int	Hex	0x30
D111	16-bit int	Hex	0x32
D112	16-bit int	Hex	0x0
M8161	BOOL	Bin	ON
D200	16-bit int	Hex	0x44
D201	16-bit int	Hex	0x37

The LRC code is acquired by calculating the two's complement of the sum of values within the range from the communication address to the end of data content.

For example: 01 H + 03 H + 21 H + 02 H + 00 H + 02 H = 29 H, and the two's complement of the sum is D7H (which corresponds to the ASCII codes 44H and 37H).

4.12 Peripheral Instructions

PID calculation	PID	PID calculation
Bit switch access	TKY	Ten key input
	HKY	Hexadecimal key input
	DSW	Digital switch
	DECO	Data decoding
	ENCO	Data encoding
7-segment LED display	SEGD	7-segment decoding
	SEGL	Seven segment with latch
Other peripheral instructions	ASC	ASCII code conversion
	PR	ASCII code printing
	IST	Status initialization
	MTR	Input matrix
	PRUN	Octal bit transfer
	ARWS	Arrow switch
	ABSD	Absolute cam control mode
	INCD	Incremental cam control mode
	ROTC	Rotary table control
	GRY	Gray code conversion
	GBIN	Gray code inverse conversion

4.12.1 PID Calculation

PID: PID calculation

◆ Overview

The PID instruction performs PID calculation to control the parameters of a close-loop control system.

PID S1 S2 S3 D		PID calculation	Applicable model: H3U	
S1	Target value	Target PID setting	16-bit instruction (9 steps) PID: Continuous execution	
S2	Feedback value	Actually measured feedback value		
S3	Operation parameter	Start number of units that store the operation result		
D	Output value	Unit that stores the PID output value		

◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

1) PID calculation selection

Unit	Function	Meaning		Remarks
S3 + 0	Sampling cycle			
S3 + 1	Calculation mode selection	0x00**	Incremental PID instruction	Higher-8-bit control instruction operation mode
		0x01**	Position-type PID instruction	
		0x02**	Special instruction (with the same function as the Inovance AC drive (customized version 307)	
S3 + 25…S3 + 30	Occupied by internal operation	-		-

The meaning of the S3 unit varies with different functional instructions.

- Incremental PID instruction

The following table lists the functions and setting methods of the parameters from head address S3.

Unit	Function	Setting
S3	Sampling time (TS)	The sampling time ranges from 1 ms to 32,767 ms and must be greater than the PLC's scan cycle.
S3+1	Action direction (ACT)	Bit 0 = 0: forward action; bit 0 = 1: reverse action Bit 1 = 0: input variable alarm disabled; bit 1 = 1: input variable alarm enabled Bit 2 = 0: output variable alarm disabled; bit 2 = 1: output variable alarm enabled Bit 3: unavailable Bit 4 = 0: auto-tuning not executed; bit 4 = 1: auto-tuning executed (The current version does not provide auto-tuning for the moment.) Bit 5 = 0: upper/lower output limits invalid; bit 5 = 1: upper/lower output limits valid Bits 6 to 15: unavailable Do not set both bits 5 and 2 to ON.
S3+2	Input filter constant (α)	Value range: 0 to 99, in percent. When it is set to 0, no input filter is processed.
S3+3	Proportional gain (Kp)	Value range: 1 to 32,767, in percent.
S3+4	Integral time (T1)	Value range: 0 to 32,767 (x 100 ms). When it is set to 0, it is processed as ∞ (no integral).
S3+5	Differential gain (KD)	Value range: 0 to 100, in percent. When it is set to 0, no differential gain is processed.
S3+6	Differential time (TD)	Value range: 0 to 32,767 (x 10 ms). When it is set to 0, no differential is processed.
S3+(7–19)	Occupied by the internal processing of PID calculation. Clear these units before initial running.	
When bit 1 = 1 and bit 2 or bit 5 = 1 in <ACT>, S3+(20–24) are occupied and defined as follows:		
S3+20	Input variable (incremental) alarm value	Value range: 0 to 32,767. This parameter is valid when bit 1 = 1 in <ACT>.
S3+21	Input variable (decremental) alarm value	Value range: 0 to 32,767. This parameter is valid when bit 1 = 1 in <ACT>.
S3+22	Output variable (incremental) alarm value	Value range: 0 to 32,767. This parameter is valid when bit 2 = 1 and bit 5 = 0 in <ACT>.
		Value range of the upper output limit: –32,768 to +32,767. This parameter is valid when bit 1 = 0 and bit 5 = 1 in <ACT>.
S3+23	Output variable (decremental) alarm value	Value range: 0 to 32,767. This parameter is valid when bit 2 = 1 and bit 5 = 0 in <ACT> of S3+1.
		Value range of the lower output limit: –32,768 to +32,767. This parameter is valid when bit 1 = 0 and bit 5 = 1 in <ACT>.
S3+24	Alarm output	Bit 0 input variable (incremental) overflow Bit 1 input variable (decremental) overflow Bit 2 output variable (incremental) overflow Bit 3 output variable (decremental) overflow This parameter is valid when bit 1 = 1 or bit 2 = 1 in <ACT>.
S3+25	Occupied by the internal processing of PID calculation	

- Position-type PID instruction (0x01** selected for S3+1)

Address	Name	Setting Range	Meaning
S3 + 0	Sampling cycle	1 to 32,767, in ms	PID calculation cycle; default value: 10
S3 + 1	Control mode	-	0x0100: Forward direction (default) 0x0101: Reverse direction
S3 + 2	Proportional gain Kp1	0 to 32,767, in percent	Proportional gain; default value: 0
S3 + 3	Integral gain Ki1	0 to 32,767, in percent	Integral gain; default value: 0
S3 + 4	Differential gain Kp1	0 to 32,767, in percent	Differential gain; default value: 0
S3 + 5	Deviation dead zone	0 to 32,767	0: Disabled Non-0: Deviation is zero if the deviation value is less than the specific value. Default value: 0
S3 + 6	Upper output limit	–32,768 to +32,767	Maximum output value
S3 + 7	Lower output limit	–32,768 to +32,767	Minimum output value
S3 + 8	Upper integral limit	–32,768 to +32,767	Maximum cumulative integral value ※1
S3 + 9	Lower integral limit	–32,768 to +32,767	Minimum cumulative integral value ※1
S3 + 10 S3 + 11	Cumulative integral	-	32-bit floating-point number
S3 + 12	Last output	–32,768 to +32,767	Used for differential calculation
S3 + 13	Kp2	0 to 32,767, in percent	Default value: 0
S3 + 14	Ki2	0 to 32,767, in percent	Default value: 0
S3 + 15	Kd2	0 to 32,767, in percent	Default value: 0
S3 + 16	Parameter switching condition	-	0: No switching 1: Switching based on deviation 2: User-defined ※2
S3 + 17	Lower deviation limit E1	–32,768 to +32,767	Deviation starting point or user-defined switching starting point
S3 + 18	Upper deviation limit E2	–32,768 to +32,767	Deviation end point or user-defined switching end point
S3 + 19	User-defined switching reference	–32,768 to +32,767	Switching reference when the parameter switching condition is set to 2
S3 + 20 S3 + 21 S3 + 22 S3 + 23 S3 + 24 S3 + 25 S3 + 26	Occupied by internal operation	-	-

Note:

※1: When the upper and lower integral limits are set to 0, the upper limit +32,737 and lower limit -32,768 take effect.

※2: When (S3+16) = 0, (S3+17) to (S3+19) are invalid.

2) Principle of position-type PID calculation

- PID calculation formula:

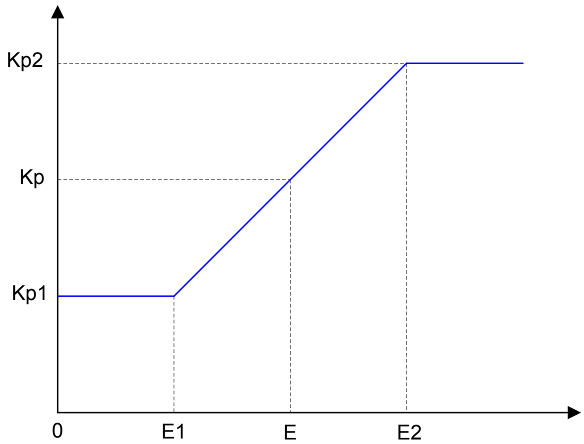
u(k) = Kp x e(k) + Ki x T x ∑e(i) + (Kd/T) x [Pv(k) – Pv(k-1)]

u(k)	Current output value	Pv(k-1)	Feedback value at the last time point
e(k)	Current deviation	T	Sampling time
∑e(i)	Current cumulative integral	Kp	Proportional gain
Sv(k)	Current value	Ki	Integral gain
Pv(k)	Current feedback value	Kd	Differential gain

Forward direction: e(k) = Sv(k) – Pv(k)

Reverse direction: e(k) = Pv(k) – Sv(k)

3) Principle of parameter switching (proportional gain Kp used an as example)



Kp1	(S3 + 2)
Kp2	(S3 + 13)
E1	(S3 + 17)
E2	(S3 + 18)
E	Switching reference

When E ≤ E1, Kp = Kp1.

When E1 < E < E2, Kp = (Kp2 – Kp1) x E/(E2 – E1).

When E ≥ E2, Kp = Kp2.

S3 + 16	0	No switching
	1	E = |Sv – Pv|
	2	E = S3 + 19

4) **Special instruction (0x02** selected for S3+1, with the same operation principle as the Inovance AC drive (customized version 307)**

Address	Name	Setting Range	Meaning	AC Drive Function Code	Wire Take-up Parameter	Wire Take-off Parameter	Wire Drawing Machine Parameter
S3 + 0	Sampling time	1 to 32,767, in ms	PID calculation cycle	-	10	10	10
S3 + 1	Mode setting	-	0x0200: Forward direction 0x0201: Reverse direction	-	-	-	-
S3 + 2	Default parameter selection	-	0: No initialization 1: Wire take-up parameters 2: Dynamic wire take-off 3: Wire drawing machine parameters	-	1	2	3
S3 + 3	Feedback range setting	0 to 32,767	AND feedback range setting	FA-04	1000	1000	1000
S3 + 4	Output range	0 to 32,767	Output range		10,000	10,000	10,000
S3 + 5	Reverse maximum output	0 to 32,767	Reverse maximum output ※1		10,000	10,000	10,000
S3 + 6	Output range selection	-	0: Relative to the maximum range 1: Relative to the main output (D+1)	F0-05	0	0	1
S3 + 7	Auxiliary output range	0 to 32,767, in percent	Valid when (S3+6) = 1	F0-06	-	-	70
S3 + 8	Proportional gain Kp1	0 to 32,767 (0.1%)	Proportional gain; default value: 0	FA-05	100	150	45
S3 + 9	Integral time Ti1	0 to 32,767 (0.01s)	Integral gain; default value: 0	FA-06	120	130	200
S3 + 10	Differential time Td1	0 to 32,767 (0.001s)	Differential gain; default value: 0	FA-07	150	0	0
S3 + 11	Deviation limit	0 to 32,767 (0.1%)	Maximum calculation deviation	FA-09	0	0	0
S3 + 12	Differential limit	0 to 32,767 (0.01%)	Maximum differential limit	FA-10	50		
S3 + 13	PID reference change time	0 to 32,767, in ms	When this parameter is enabled, the reference value will equal the setting after the specified time has elapsed.	FA-11	5000	0	0
S3 + 14	Proportional gain Kp2	0 to 32,767 (0.1%)	Default value: 0	FA-15	-	-	-
S3 + 15	Integral time Ti2	0 to 32,767 (0.01s)	Default value: 0	FA-16	-	-	-
S3 + 16	Differential time Td2	0 to 32,767 (0.001s)	Default value: 0	FA-17	-	-	-
S3 + 17	Parameter switching condition		0: No switching 1: Switching based on deviation 2: User-defined ※2	FA-18	-	-	-
S3 + 18	Lower deviation limit	0 to 32,767 (0.1%)	Deviation starting point or user-defined switching starting point	FA-19	-	-	-

Address	Name	Setting Range	Meaning	AC Drive Function Code	Wire Take-up Parameter	Wire Take-off Parameter	Wire Drawing Machine Parameter
S3 + 19	Upper deviation limit	0 to 32,767 (0.1%)	Deviation end point or user-defined switching end point	FA-20	-	-	-
S3 + 20	User-defined switching reference	0 to 32,767 (0.1%)	Switching reference when the parameter switching condition is set to 2		-	-	-
S3 + 21	Initial output	0 to 32,767 (0.1%)	Initial value after PID startup	FA-21	0	0	0
S3 + 22	Initial output hold time	0 to 32,767, in ms	Time during which the initial value remains unchanged	FA-22	0	0	0
S3 + 23	Output deviation limit	0 to 32,767 (0.1%)	Range of every deviation change		0	0	0
S3 + 24 … S3 + 30	Internal operation	-	-	-	-	-	-

Address	Name	Meaning
D + 0	Total output	PID calculation element + (D + 1)
D + 1	Main output	User-designated main output (AC drive dominant frequency) This value is set to 0 for pure PID.

Note:

※1: Maximum negative value of PID output. For example, if this parameter is set to 100, the maximum negative output value is –100.

※2: See the parameter switching principle of the position-type PID instruction.

- PID calculation formula

$u(k) = Kp \{e(k) + T/Ti \times \sum e(i) + Td/T \times [e(k) - e(k-1)]\}$

u(k)	Current output value	∑e(i)	Current cumulative integral
Kp	Proportional gain	T	Sampling time
e(k)	Current deviation	Ti	Integral time
e(k-1)	Deviation at the last time point	Td	Differential time
Sv(k)	Current value	Ki	Integral gain
Pv(k)	Current feedback value	Kd	Differential gain

Forward direction: e(k) = Sv(k) – Pv(k); reverse direction: e(k) = Pv(k) – Sv(k)

- For details about parameter switching, see the position-type PID description.
- Main output application

When (S3 + 6) = 0, (D + 1) is forcibly set to 0.

When (S3 + 6) = 1, (S3 + 7) is enabled. The maximum PID element is equal to (S3 + 1) percent of (D + 1). Final (D + 0) = PID element + Main output (D + 1)

4

4.12.2 Bit Switch Access

Bit switch access	TKY	Ten key input
	HKY	Hexadecimal key input
	DSW	Digital switch
	DECO	Data decoding
	ENCO	Data encoding

TKY: Ten key input

◆ Overview

The TKY instruction assigns a 4- or 8-digit value to a word element through 10 consecutive bit elements and drives the corresponding bit elements to act.

TKY S D1 D2		Ten key input	Applicable model: H3U	
S	Input port head address	Initial key input port (for example, X port), with 10 consecutive units occupied	16-bit instruction (7 steps) TKY: Continuous execution	32-bit instruction (13 steps) DTKY: Continuous execution
D1	Data storage unit	Address at which key input data is stored		
D2	Input signal indicator	Head address of 11 consecutive bit elements that store the actions corresponding to key input		

◆ Operands

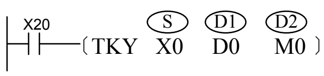
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

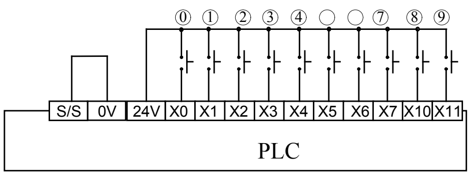
◆ Function

Ten consecutive bit variable units (for example, X input ports) are designated to represent the 0 to 9 keys in decimal format. When key press actions are initiated (status: ON), 4-digit decimal values in the range 0 to 9999 can be input based on the action sequence. If the 32-bit instruction is used, 8-digit decimal values in the range 0 to 99,999,999 can be input.

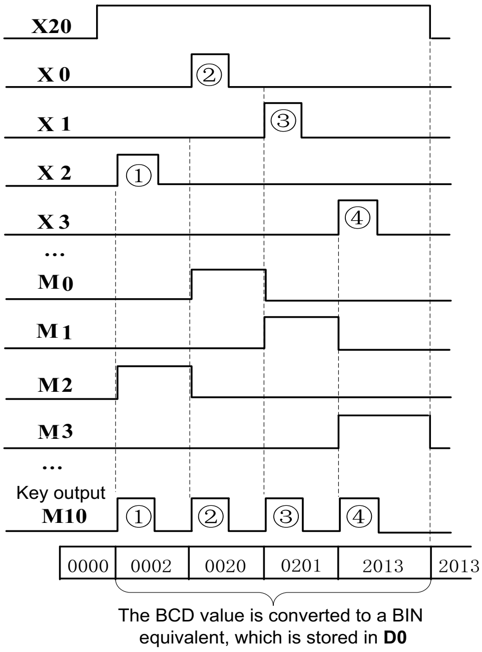
● Example



The following figure shows the hardware wiring diagram.



To input 2013, press keys ② ,⓪, ① , and ③ (X2, X0, X1, and X3) in sequence. The following figure shows the actions of the PLC's internal variables.



- Based on parameter setting, X0 to X11 represent numeric keys 0 to 9, and M0 to M9 indicate the key status. When any key is pressed, the key output unit M10 is set.
- The key value (for example, 2013) is converted to a BIN value and then stored in the specified D1 (D0 = 0x7DD), which remains unchanged even when the driver flow switches to OFF.
- When multiple keys are pressed, the first detected key is active. When the input number has more than four digits, the first digit overflows and only the last four digits are retained.

If the 32-bit instruction DTKY is used, D1 occupies a 32-bit variable, that is, D1 and D0 in the preceding example, which correspond to the higher and lower bytes respectively.

HKY: Hexadecimal key input

◆ Overview

The HKY instruction assigns a 4- or 8-digit hexadecimal value to a word element through a 4x4 input matrix and drives the corresponding bit elements to act.

HKY S D1 D2 D3		Hexadecimal key input	Applicable model: H3U	
S	Input port head address	Head address of four consecutive bit elements occupied by the key input port	16-bit instruction (9 steps) HKY: Continuous execution	32-bit instruction (17 steps) DHKY: Continuous execution
D1	Gating bit element head address	Head address of four consecutive bit elements occupied by the gating port		
D2	Data storage unit	Address of the word element that stores key inputs in the range 0 to 9999, or 0 to 99,999,999 when the 32-bit instruction is used.		
D3	Input signal indicator	Head address of eight consecutive bit elements that store the actions corresponding to key input		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

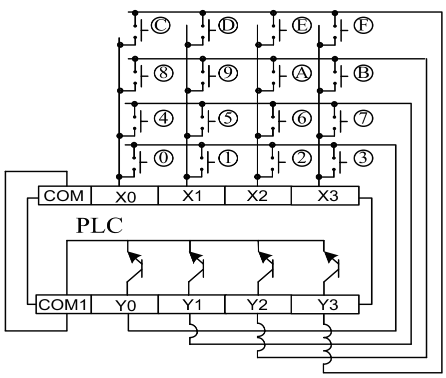
Sixteen keys in a 4x4 matrix are read to represent the 0 to 9 keys in decimal format and the A to F function keys. When key press actions are initiated (status: ON), 4-digit decimal values in the range 0 to 9999 can be input or the A to F function keys can be operated based on the action sequence. If the 32-bit instruction is used, 8-digit decimal values in the range 0 to 99,999,999 can be input or the A to F function keys can be operated based on the action sequence.

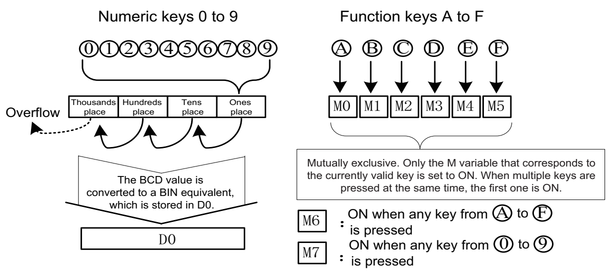
This instruction is only applicable to PLCs of the transistor output type.

- Example

```
      X20       (S)   (D1)  (D2)  (D3)
 ─┤├──────[HKY  X0    Y0    D0    M0 ]
```

The wiring diagram and parameter response description are as follows:





1. A controller of the MT transistor type is required due to the great output delay of relays.
2. When the drive flow X20 is OFF, D0 remains unchanged, and M0 to M7 are set to OFF.
3. It takes eight scan cycles to complete a key scan operation. After scanning, M8029 is set by one scan cycle.

Because it takes several cycles to complete key scanning, enable the constant scan mode or timing interrupt processing to avoid the impact of X port filter.

- Extended function

When the special variable M8167 is set to ON, this instruction stores keys⓪to⑨in hexadecimal format in D2.

DSW: Digital switch

◆ Overview

The DSW instruction reads digital switch setting.

DSW S D1 D2 n		Digital switch	Applicable model: H3U
S	Input port head address	Start number of X ports used for key scan input If n = 1, four consecutive X ports are occupied; if n = 2, eight consecutive X ports are occupied.	16-bit instruction (9 steps) DSW: Continuous execution
D1	Gating bit element head address	Start number of Y ports used for key scan output. Four consecutive Y ports are occupied.	
D2	Data storage unit	Unit that stores key inputs, in the range D0 to 9999.	
n	Data count	Number of digital switch sets; optional values: 1 and 2.	

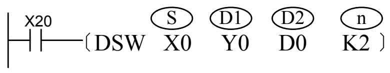
◆ Operands

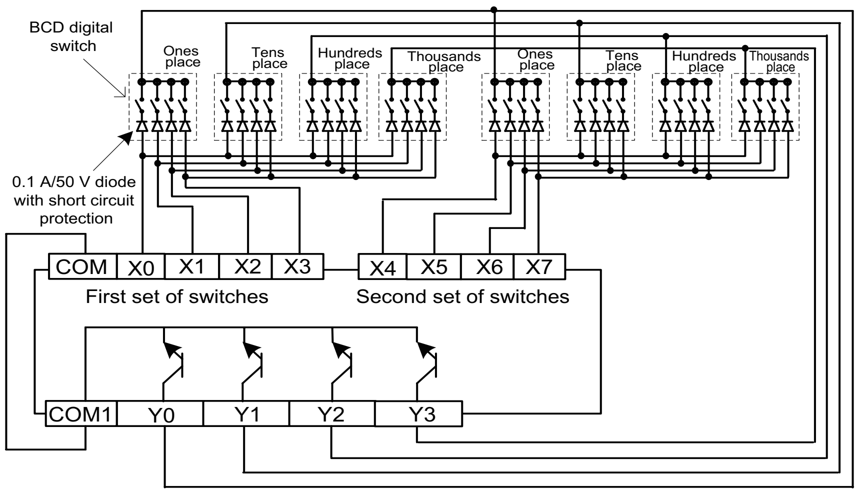
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

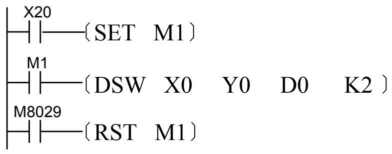
The statuses of matrix-type switches are read and stored in a designated unit. Each set of switches consists of four BCD switches. A maximum of two sets can be read. Example:





When X20 = ON, the system scans and reads the digital switch setting.
1. The values of the first set of digital switches are converted to BIN values, which are stored in D0.
2. The values of the second set of digital switches are converted to BIN values, which are stored in D1.
3. M8029 is set by one scan cycle after a reading loop.

Usage: Digital switches can be detected only by PLCs of the transistor output type. It takes multiple scan cycles to complete a digital switch read operation. If a read operation is initiated in key press mode, the following statement is recommended to ensure reading cycle completeness:



DECO: Data decoding

◆ Overview

The DECO instruction is used for data decoding.

DECO S D n			Data decoding	Applicable model: H3U	
S	Data source to be decoded	Address of the word element that stores the source data, or head address of bit elements		16-bit instruction (7 steps) DECO: Continuous execution DECOP: Pulse execution	
D	Decoding result	Address of the word element that stores the decoding result, or head address of bit elements			
n	Decoded bit length	Number of consecutive bits from head address S to be decoded			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The values of the last n bits from head address S are calculated. The result is used as a bit pointer. The corresponding bits of D are set to 1, and other bits are cleared.

The lower n (n ≤ 4) bits from head address S are decoded to the destination address. When n ≤ 3, the destination higher bits are set to 0.

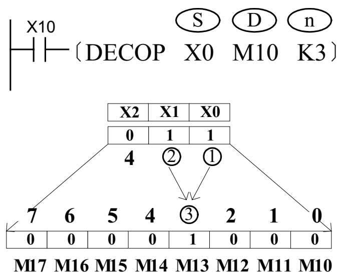
When n = 0, the instruction is not executed. When n is out of the range 0 to 8, an operation error will occur.

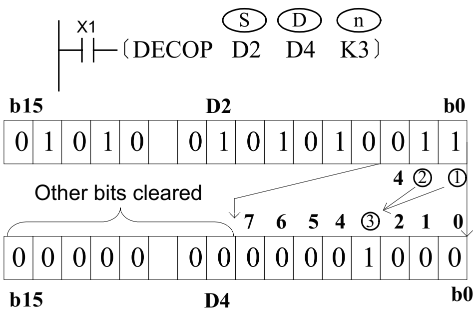
When n = 8, if D is a bit element, it has 256 bits.

When the driver input is OFF, the instruction is not executed but the ongoing decoding output continues.

The instruction of the pulse execution type is generally used.

Example:





ENCO: Data encoding

◆ Overview

When driving conditions are met, the ENCO instruction converts active (ON) bit elements from head address S or active (ON) bits in a word element to binary numbers. The result is stored in D. The number of bits from head address S is specified by 2^n.

ENCO S D n			Data encoding	Applicable model: H3U	
S	Source data to be encoded	Input data to be encoded, or head address of bit elements		16-bit instruction (7 steps) ENCO: continuous execution ENCOP: Pulse execution	
D	Encoding result	Address of the word element that stores the encoding result			
n	Encoded bit length	Number of data bits in D			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

The values of the last n bits from head address S are calculated. The result is used as a bit pointer. The corresponding bits of D are set to 1, and other bits are cleared.

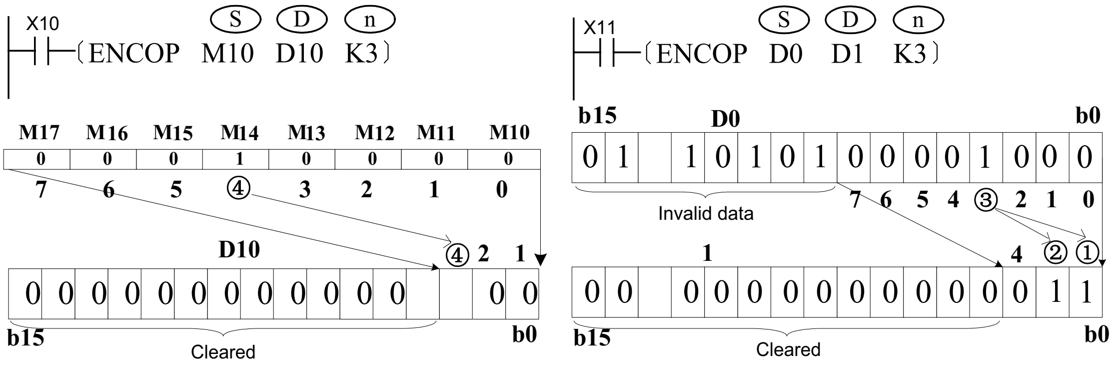
When multiple bits from head address S are 1, only the first higher bit among those bits is calculated. When all bits from head address S are 0, an operation error will occur.

When the driver input is OFF, the instruction is not executed but the ongoing encoding output continues.

When n = 8, if S is the head address of bit elements, the number of data bits is 256.

The instruction of the pulse execution type is generally used.

Example:



4.12.3 7-segment LED Display

7-segment LED display	SEGD	7-segment decoding
	SEGL	Seven segment with latch

SEGD: 7-segment decoding

◆ Overview

When driving conditions are met, the lower four bits of S1 are decoded into 7-segment display codes. The result is stored in the lower eight bits of D1.

SEGD S D			7-segment decoding	Applicable model: H3U	
S	Data source	Data source to be decoded (the lower four bits b0 to b3 of the BIN content are taken)		16-bit instruction (5 steps) SEGD: Continuous execution SEGDP: Pulse execution	
D	Decoding result	Variable that stores the 7-segment display codes after decoding			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

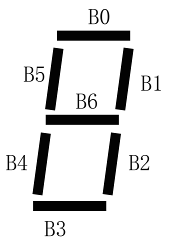
Note: The elements in gray background are supported.

◆ Function

The lower four bits of the data source are decoded into 7-segment display codes. The result is stored in the lower eight bits of the destination variable. Example:

When X20 = ON, the lower four bits of D0 are decoded. The result is output to ports Y10 to Y17.

The following table is used for decoding. The table is provided by the PLC and does not need to be prepared manually.

Data HEX	Data BIN	7-segment LED Display Combination	Internal Decoding Table Value B7	B6	B5	B4	B3	B2	B1	B0	Decoded Character
0	0000	 Each bit corresponding to one segment 1 = Segment ON 0 = Segment OFF		0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

SEGL: Seven segment with latch

◆ Overview

The SEGL instruction drives 4- or 8-digit latched 7-segment LED displays by using 8 or 12 Y ports. The display mode is scan-driven.

SEGL S D n			Seven segment with latch	Applicable model: H3U	
S	Data source	Data to be displayed (The value is converted to a BCD equivalent before being sent to a 7-segment LED display.)		16-bit instruction (7 steps) SEGL: Continuous execution	
D	Driver port head address	Head address of Y ports for display driving			
n	Related setting	Setting based on the number of displayed data sets as well as signal positive and negative logic			

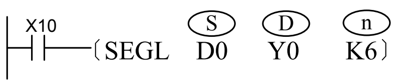
◆ Operands

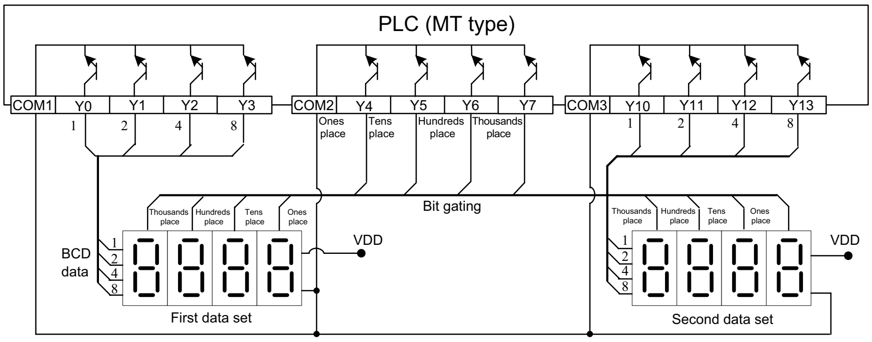
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Eight or 12 Y ports are used to drive 4- or 8-digit latched 7-segment LED displays. The display mode is scan-driven. Example:





In the preceding hardware wiring diagram, the first 7-segment LED display shows the content of D0, and the second one shows the content of D1. A program running error will occur if the readings of D0 or D1 exceed 9999.

The 7-segment LED displays shown in the wiring diagram feature data display latch, 7-segment decoding and driving, and a negative logic model. (When the input port is at low level, the input data is 1 or gated.) During display processing, ports Y4 to Y7 of the PLC perform cyclic scan automatically. Only one port is ON at a time and functions as a bit gating signal. The data on ports Y0 to Y3 is the BCD data sent to corresponding bits. When the bit gating signal switches from ON to OFF, the data is latched to 7-segment LED displays, which display digits after internal decoding and driving. The PLC system performs the same processing on Y4 to Y7 in sequence until all four bits are processed. Y10 to Y13 are the data output ports of the second 4-digit 7-segment LED display and share the bit gating wire with Y4 to Y7. The processing method is the same. Display is processed simultaneously for both 7-segment LED displays. In the example, if D0 = K2468 and D1 = K9753, the first 7-segment LED display shows "2 4 6 8", and the second one shows "9 7 5 3".

It takes 12 scan cycles to complete a display refresh operation. The M8029 flag is set to ON after processing is completed.

Set n as follows based on the positive and negative logic of the PLC and 7-segment codes:

n = 0–3 when there is one 4-digit data set. n = 4–7 when there are two 4-digit data sets.

Number of Displayed Data Sets	One set				Two sets			
Y Data Output Polarity	PNP		NPN		PNP		NPN	
Gating and Data Polarity	Same	Inverse	Same	Inverse	Same	Inverse	Same	Inverse
Value of n	0	1	2	3	4	5	6	7

The value n determines whether the transistor output polarity of the PLC and the input polarity of 7-segment LED displays are the same or inverse.

The Y output polarity of the H1U and H2U series PLCs of the transistor output type is NPN. This instruction can be used twice at most in a program.

Usage:

Because relays are not suitable for high-frequency scan output, this instruction is only applicable to PLCs of the transistor output type.

4.12.4 Other Peripheral Instructions

Other peripheral instructions	ASC	ASCII code conversion
	PR	ASCII code printing
	MTR	Input matrix
	PRUN	Octal bit transfer
	ARWS	Arrow switch
	ABSD	Absolute cam control mode
	INCD	Incremental cam control mode
	ROTC	Rotary table control
	GRY	Gray code conversion
	GBIN	Gray code inverse conversion

ASC: ASCII code conversion

◆ Overview

When driving conditions are met, the ASC instruction converts the string S1 input by a PC to ASCII codes. The result is stored in registers from head address D1.

ASCII S D			ASCII code conversion	Applicable model: H3U	
S	Data source	Alphanumeric string (which consists of up to eight characters) to be converted to ASCII codes		16-bit instruction (11 steps) ASCII: Continuous execution	
D	Conversion result	Start number of four (M8161 = 0) or eight (M8161 = 1) consecutive variable units that store the ASCII codes after conversion			

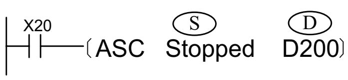
◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	Alphanumeric string that is input manually																					
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Example:



The figure on the right shows the values assigned to D200 to D203 when X20 = ON.

	Higher bytes	Lower bytes
D200	54 (t)	53 (S)
D201	50 (p)	4F (o)
D202	45 (e)	50 (p)
D203	20	44 (d)

If the special register M8161 is set to ON, each ASCII character after conversion occupies a 16-bit variable, as shown in the following figure. The higher byte of each variable is set to 0.

	Higher bytes	Lower bytes
D200	00	53（S）
D201	00	54（t）
D202	00	4F（o）
D203	00	50（p）
D204	00	50（p）
D205	00	45（e）
D206	00	44（d）
D207	00	20

Appendix: ASCII code mapping table

Decimal	ASCII (Hexadecimal)
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39

Code	ASCII (Hexadecimal)
STX	02
ETX	03

English Letter	ASCII (Hexadecimal)	English Letter	ASCII (Hexadecimal)
A	41	N	4E
B	42	O	4F
C	43	P	50
D	44	Q	51
E	45	R	52
F	46	S	53
G	47	T	54
H	48	U	55
I	49	V	56
J	4A	W	57
K	4B	X	58
L	4C	Y	59
M	4D	Z	5A

PR: ASCII code printing

◆ Overview

The PR instruction outputs the values stored in designated variable units synchronously byte by byte through Y output ports.

PR S D			ASCII code printing	Applicable model: H3U	
S	Data source	Head address of variable units that store the data to be output		16-bit instruction (5 steps) PR: Continuous execution	
D	Output port head address	Start number of Y output ports			

◆ Operands

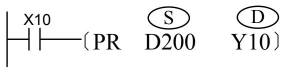
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

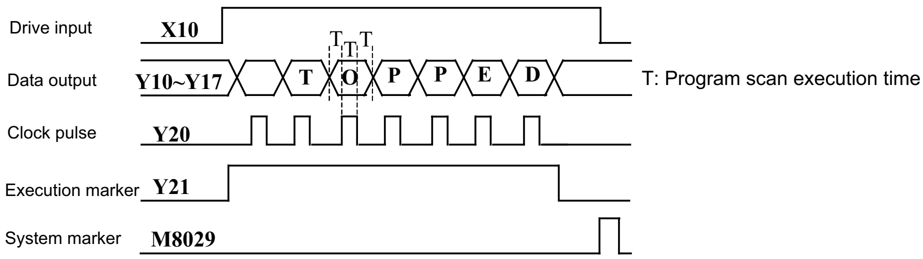
◆ Function

The values stored in designated variable units are output synchronously byte by byte through Y output ports.

Example:



If the ASCII codes stored in D200 to D203 correspond to "STOPPED", the corresponding output port signals and time sequence are as follows:



When X020 is valid, the ASC instruction is executed to convert the string "STOPPED" to ASCII codes. The result is stored in D200 to D203.

When X021 is valid, the PR instruction is executed to print the ASCII codes in D200 to D203 in the S -> T -> O -> P -> P -> E -> D sequence to an external display through ports Y10 to Y17.

Usage:

- This instruction is only applicable to PLCs of the transistor output type.
- When the driver signal X10 switches to OFF, the printing process is terminated. Printing resumes when X10 switches back to ON.
- Printing stops at the presence of characters 00, and subsequent content is not processed.
- When M8027 = OFF, serial output is performed with a fixed length of eight characters; when M8027 = ON, serial output is performed with a length of 1 to 16 characters.
- When M8027 = OFF, M8029 does not take action after the flow becomes inactive.
- When M8027 = ON, the complete flag M8029 is set to ON after the driver flow signal becomes inactive.

This instruction is executed based on scan cycles (indicated by T in the preceding figure). If the scan cycle is short, use the constant scan mode; if the scan cycle is long, execute this instruction within the timing interrupt program.

MTR: Input matrix

◆ Overview

The MTR instruction is used to form an input matrix.

MTR S D1 D2 n		Input matrix	Applicable model: H3U	
S	Input operand head address	Head address of eight consecutive input bit elements with numbers whose lowest bit is 0, such as X0 and X10	16-bit instruction (9 steps) MTR: Continuous execution	
D1	Output operand head address	Head address of n (2 to 8) consecutive output bit elements with numbers whose lowest bit is 0, such as Y0 and Y10		
D2	Input status storage head address	Head address of elements that store the status of input bit elements (the lowest bit of each storage element number must be 0, such as Y0, M0, and S0)		
n	Input column count	Number of input matrix columns, that is, the number of Y outputs to be scanned		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

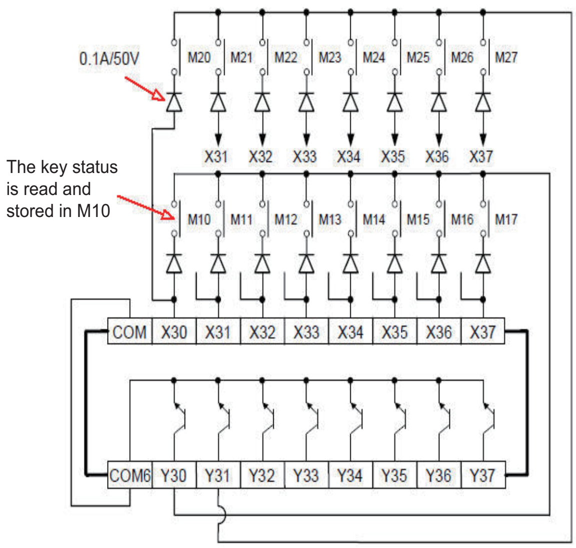
Note: The elements in gray background are supported.

◆ Function

This instruction is only applicable to PLCs of the transistor output type. Eight X ports and several Y ports form an input matrix network to increase the number of input signal channels. This instruction typically uses the ON contact M8000. Example:

M8000 —| |— (MTR X30 Y30 M10 K2)

The following figure shows the wiring diagram.



Considering the 10-ms response delay of the X input filter, the output of Y30 and Y31 is interrupted every 20 ms to enable instant input/output processing.

The M8029 flag is set to ON after every automatic read operation is completed.

A maximum of 64 scan inputs can be acquired using eight X inputs and eight transistor Y outputs. However, it takes 160 ms (= 20 ms x 8 columns) to read all inputs, which is impractical for high-speed input. Therefore, ports after X20 are typically used for scan input.

PRUN: Octal bit transfer

◆ Overview

The PRUN instruction batch copies consecutive bit variables from head address S1 to a set of bit variables from head address D1. The copy operation uses the octal width unit.

PRUN S D			Octal bit transfer	Applicable model: H3U	
S	Transferred bit head address	Head address of bit variables to be copied. The ones place of the address must be 0, such as X10 and M20.		16-bit instruction (5 steps) PRUN: Continuous execution PRUNP: Pulse execution	32-bit instruction (9 steps) DPRUN: Continuous execution DPRUNP: Pulse execution
D	Received bit head address	Head address of destination bit variables in a copy operation. The ones place of the address must be 0, such as M30 and Y10.			

4

◆ Operands

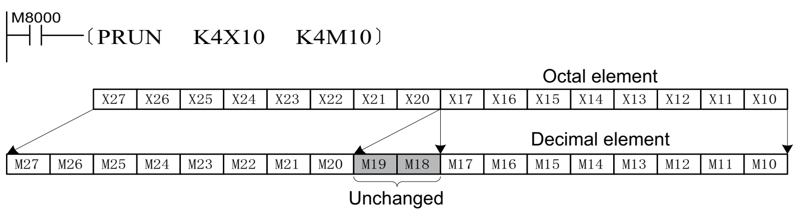
Operand	Bit Element							Word Element														
	System·User							System·User				Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

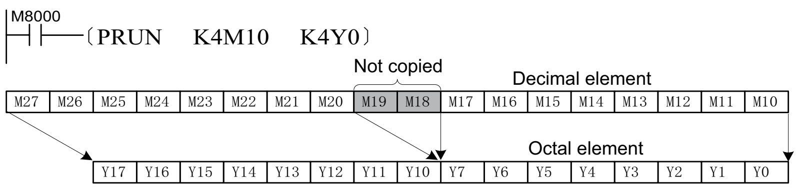
◆ Function

Consecutive bit variables from head address S are batch copied to a set of bit variables from head address D. The copy operation uses the octal width unit. In Kn, n ranges from 1 to 8.

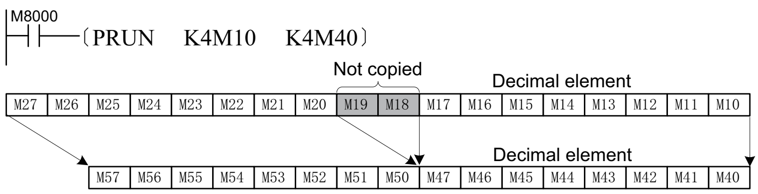
Example 1:



Example 2:



Example 3:



ARWS: Arrow switch

◆ Overview

The ARWS instruction enables easy editing of parameters stored in registers. X ports are used to provide edit keys, and Y ports are used to drive 4-digit 7-segment LED displays.

ARWS S D1 D2 n			Arrow switch	Applicable model: H3U	
S	Input port head address	Head address of four consecutive units that store key inputs		16-bit instruction (9 steps) ARWS: Continuous execution	
D1	Data storage address	Variable that is displayed and modified (Only one 16-bit variable is displayed.)			
D2	Output port head address	Head address of eight consecutive Y ports used for 7-segment LED display driving			
n	Related setting	Signal logic setting. For details, see the description of n in "SEGL: Seven segment with latch" on Page 320.			

◆ Operands

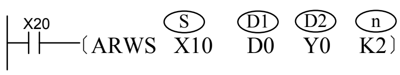
Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

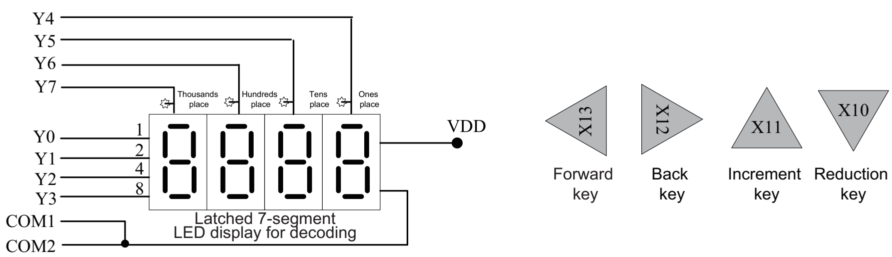
◆ Function

Parameters stored in registers can be edited easily by using X ports to provide edit keys and Y ports to drive 4-digit 7-segment LED displays.

Example:



The following figure shows the hardware wiring diagram. The PLC is of the transistor output type.



Operation:

1) The 7-segment LED display shown in the figure displays the value in D0, in the range 0 to 9999. Press X10 to X13 to modify the value.

2) When X20 = ON, the cursor is located at the thousands place. Each time the back key (X12) is pressed, the designated digit is switched through thousands place -> hundreds place -> tens place

-> ones place -> thousands place in sequence. If the forward key (X13) is pressed, the switching sequence is reversed. The cursor position is indicated by the LED indicator connected to the gating pulse signal (Y004 to Y007).

3) Each time the increment key (X11) is pressed, the content pointed by the cursor is changed through 0 -> 1 -> 2 -> ……8 -> 9 -> 0 -> 1 in sequence. Each time the decrement key (X10) is pressed, the content pointed by the cursor is changed through 0 -> 9 -> 8 -> 7 -> …… -> 1 -> 0 -> 9 in sequence. The modified value takes effect immediately.

Usage:

If the user program has a short scan duration, use the constant scan mode, or start scanning at a fixed interval within a timing interrupt.

ABSD: Absolute cam control mode

4

◆ Overview

When driving conditions are met, the ABSD instruction compares a data table from head address S1 with the current value of a selected counter (S2) to control the ON/OFF states of K bit elements from head address D.

ABSD S1 S2 D n		Absolute cam control mode	Applicable model: H3U	
S1	Data table head address	Head address of word elements that store a data table	16-bit instruction (9 steps) ABSD: Continuous execution	32-bit instruction (17 steps) DABSD: Continuous execution
S2	Counter C	Number of a selected counter		
D	Output bit element head address	Head address of n consecutive output bit elements (variable units) that store the comparison result		
n	Output bit element count	Number of output bit elements, or number of data entries of a table		

◆ Operands

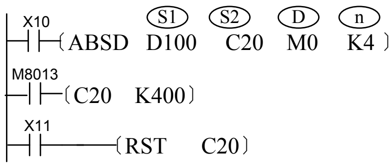
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

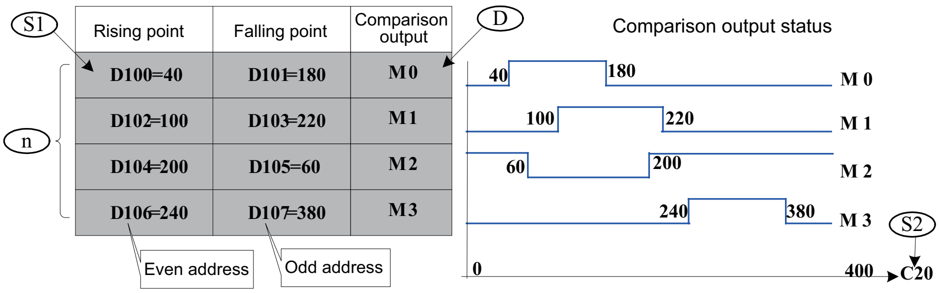
◆ Function

Multi-segment comparison is performed for cam control. The data table and counter used by comparison are configured in absolute mode. This instruction is executed during the scan process of the main program. The comparison result is affected by scan delay. In 32-bit operation, S1, S2, and D point to 32-bit variables, and n is calculated based on 32-bit variable width.

Example:



Assume that related variables are assigned values as follows. When X10 = ON, the execution result is shown in the following figure.



Usage:

- Before the ABSD instruction is executed, use the MOV instruction to assign values to variables of the related table.
- Even if the DABSD instruction uses a high-speed instruction, the comparison result D is affected by the scan delay of the user program. The HSZ instruction for high-speed comparison can be used for applications that require timely response.

INCD: Incremental cam control mode

◆ Overview

When driving conditions are met, the INCD instruction compares a data table from head address S1 with the current value of a pair of selected counters (S2) to control the ON/OFF states of K bit elements from head address D.

<table>
<tr><th colspan="3">INCD S1 S2 D n</th><th>Incremental cam control mode</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S1</td><td>Data table head address</td><td colspan="2">Head address of word elements that store a data table</td><td rowspan="4">16-bit instruction (9 steps)
INCD: Continuous execution</td><td rowspan="4"></td></tr>
<tr><td>S2</td><td>Counter C</td><td colspan="2">Numbers of two consecutive counters (The adjacent S2+1 unit indicates the number of counter reset times after comparison.)</td></tr>
<tr><td>D</td><td>Output bit element head address</td><td colspan="2">Head address of n consecutive output bit elements (variable units) that store the comparison result</td></tr>
<tr><td>n</td><td>Output bit element count</td><td colspan="2">Number of output bit elements, or number of data entries of a table</td></tr>
</table>

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

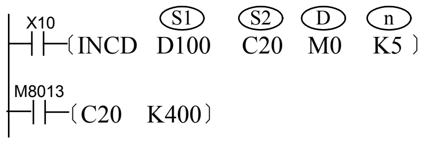
Note: The elements in gray background are supported.

◆ Function

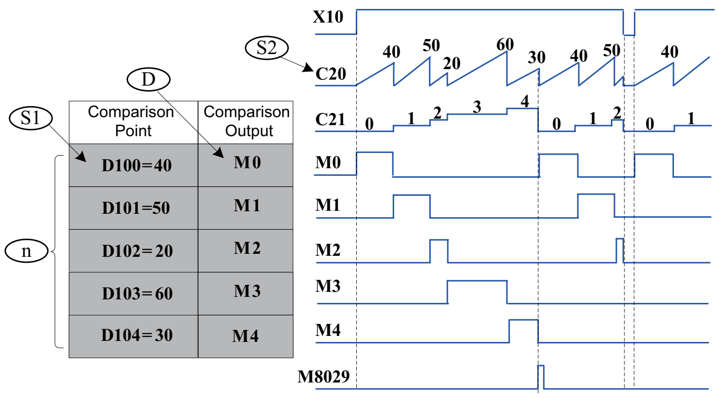
Multi-segment comparison is performed for cam control. The data table and counters used by comparison are configured in incremental mode. This instruction is executed during the scan process of the main program. The comparison result is affected by scan delay.

The complete flag M8029 is set to ON after comparison of n data entries is completed.

Example:



Assume that related variables are assigned values as follows. When X10 = ON, the execution result is shown in the following figure.



Usage:

- Before the INCD instruction is executed, use the MOV instruction to assign values to variables of the related table.
- The comparison result is affected by the scan delay of the user program. The HSZ instruction for high-speed comparison can be used for applications that require timely response.

ROTC: Rotary table control

◆ Overview

When driving conditions are met, the ROTC instruction aids the movement of a workpiece at a designated position to a specified destination along the optimal path.

ROTC S m1 m2 D			Rotary table control	Applicable model: H3U
S	Counter value storage head address	Head address of counter variables	16-bit instruction (9 steps) ROTC: Continuous execution ROTCP: Pulse execution	
m1	Work station count	Number of work stations of a table (m1 ≥ m2)		
m2	Low-speed work station count	Number of low-speed work stations of a table (m1 ≥ m2)		
D	Status bit element	Head address of eight consecutive elements that store the position detection signal of the rotary table		

◆ Operands

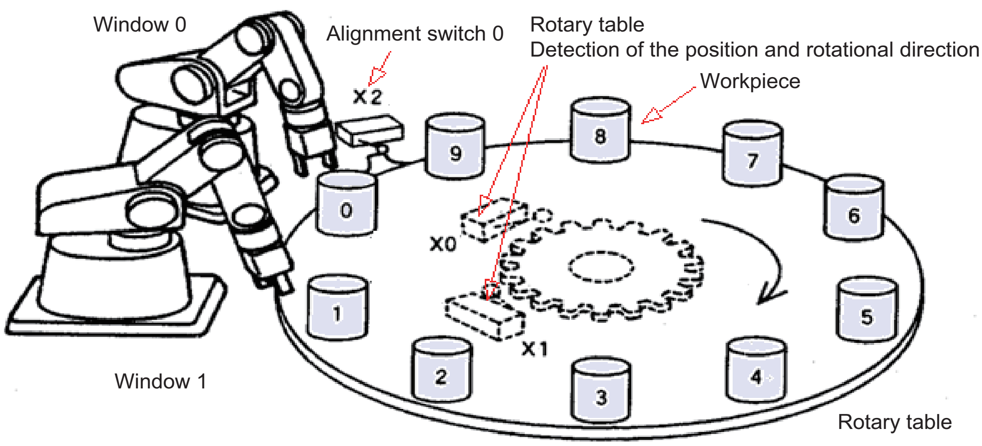
Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	**D**	**R**	T	C	**SD**	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modification**	K	H	E
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	**K**	**H**	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	**K**	**H**	E
D	X	**Y**	**M**	T	C	**S**	**SM**	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modification**	K	H	E

Note: The elements in gray background are supported.

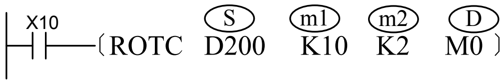
◆ Function

The handling of workpieces on a rotary table is controlled. For normal operation, the position detection signal of the rotary table must be configured using the specified method.

The following figure shows the signal configuration method. X0 and X1 are respectively connected to the A-phase and B-phase output signals of the A/B quadrature encoder. The quadrature phase signal can be acquired using a mechanical switch. X2 is connected to the detection input of work station 0 (the ON state is enabled when the rotary table rotates to this work station). The three signals are used to detect the current rotational speed, direction, and work station of the rotary table.



◆ Application



The variable space occupied by the preceding code is as follows:

Variable	Function	Operation
D200	Used as a counter register	The three units are preset by the user program.
D201	Called window number setting	
D202	Called workpiece number setting	
M0	A-phase signal	The following code is executed before the user program scans this statement: X0 (M0) X1 (M1) X2 (M2)
M1	B-phase signal	
M2	Zero-point detection signal	
M3	High-speed rotation in the forward direction	The results of M3 to M7 are automatically acquired when X10 = ON. M3 to M7 are OFF when X10 = OFF.
M4	Low-speed rotation in the forward direction	
M5	Stop	
M6	Low-speed rotation in the reverse direction	
M7	High-speed rotation in the reverse direction	

M3 to M7 are output by Y ports in the following user program. Only externally executed elements need to be controlled.

D200 is cleared when the flow is active and the zero-point signal M2 = ON. Operation starts only after the clearing operation is completed.

The ROTC instruction can be executed only once in a program.

GRY: Gray code conversion

◆ Overview

The GRY instruction converts binary numbers to gray code equivalents.

GRY S D			Gray code conversion	Applicable model: H3U
S	Data source	BIN data source or data variable unit to be converted, in the range 0 to 32,767 in 16-bit operation or 0 to 2,147,483,647 in 32-bit operation	16-bit instruction (5 steps) GRY: Continuous execution GRYP: Pulse execution	32-bit instruction (9 steps) DGRY: Continuous execution DGRYP: Pulse execution
D	Operation result	Unit that stores the gray code value after conversion		

4

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Binary numbers are converted to gray code equivalents.

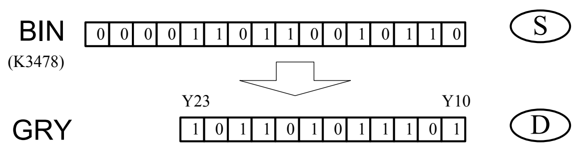
S is the BIN data source or data variable unit to be converted, in the range 0 to 32,767 in 16-bit operation or 0 to 2,147,483,647 in 32-bit operation. When this range is exceeded, M8067 and M8068 are set to ON and the instruction is not executed.

D is the unit that stores the gray code value after conversion.

Algorithm for BIN-to-gray code conversion: XOR operation is performed on every bit starting from the rightmost bit and a left-hand bit to get the gray code value. The leftmost bit remains unchanged (with a value of 0).

Example:

Execution result:



GBIN: Gray code inverse conversion

◆ Overview

The GBIN instruction converts gray code values to binary equivalents.

GBIN S D		**Gray code inverse conversion**	**Applicable model: H3U**	
S	Data source	Gray code value or data variable unit to be converted, in the range 0 to 32,767 in 16-bit operation or 0 to 2,147,483,647 in 32-bit operation	16-bit instruction (5 steps) GBIN: Continuous execution GBINP: Pulse execution	32-bit instruction (9 steps) DGBIN: Continuous execution DGBINP: Pulse execution
D	Operation result	Unit that stores the BIN value after conversion		

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Function

Gray code values are converted to binary equivalents.

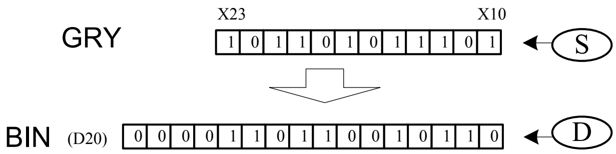
S is the gray code value or data variable unit to be converted, in the range 0 to 32,767 in 16-bit operation or 0 to 2,147,483,647 in 32-bit operation. When this range is exceeded, M8067 and M8068 are set to ON and the instruction is not executed.

D is the unit that stores the BIN value after conversion.

Algorithm for gray code-to-BIN conversion: XOR operation is performed on every bit starting from the second leftmost bit and the decoded value of a left-hand bit. The result is used as the decoded value of the bit. The leftmost bit remains unchanged.

Example:

Execution result:



4.13 Electronic Cam Instructions

Electronic cam instructions	CAMWR	Writing electronic cam data
	CAMRD	Reading electronic cam data
	ECAMWR	Writing electronic cam floating-point data
	ECAMRD	Reading electronic cam floating-point data

For details, see “8.8 Modifying Key Points for Electronic Cams” on Page 576.

4

5 High-speed Input

Chapter 5 High-speed Input

With eight high-speed inputs X00-X07, the H3U standard model supports pulse input frequency up to 200 kHz and provides single-phase unidirectional counting, single-phase bidirectional counting or A/B phase counting, and high-speed interrupt functions.

With three high-speed inputs X, Y, and Z axes, each with two differential inputs, H3U-PM supports pulse input frequency up to 200 kHz and provides pulse+ direction, A/B phase, and CW/CCW high-speed pulse counting and speed measurement functions.

Content	H3U Standard Model	PM Motion Control Model
Number of high-speed inputs	8	3
Type of high-speed input	Collector input	Differential input
Mode of high-speed input	Pulse	Pulse+ direction CW/CCW A/B phase
Maximum high-speed input frequency	200 kHz	200 kHz

5.1 H3U Standard Model

5.1.1 High-speed Counter

As shown in the following table, the built-in high-speed counters of H3U series PLC are assigned to X00 to X07 inputs by number.

Input Assignment	Single-phase Unidirectional Counter Input										
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
X000	U/D								U/D		
X001		U/D							R		
X002			U/D							U/D	
X003				U/D						R	
X004					U/D						U/D
X005						U/D					R
X006							U/D			S	
X007								U/D			S

Input Assignment	Single-phase Bidirectional Counter Input					A/B Phase Counter				
	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X000	U			U		A	A			
X001	D			D		B	B			
X002		U		R		R		A		
X003		D		S				B		
X004			U		U				A	
X005			D		D				B	
X006					R					A
X007					S					B

[U]: up counter input; [D]: down counter input; [R]: reset counter input; [S]: start counter input

[A]: A-phase pulse input; [B]: B-phase pulse input

1) Special M element for single-phase unidirectional up/down counter

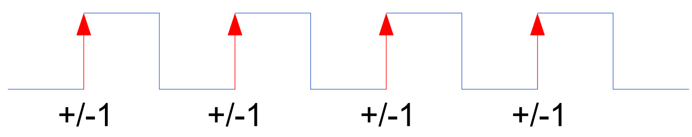
	Single-phase Unidirectional Counter										
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
Up/Down counter control	M8235	M8236	M8237	M8238	M8239	M8240	M8241	M8242	M8243	M8244	M8245

2) Special M element for single-phase bidirectional counting up/down and A/B phase counting up/down

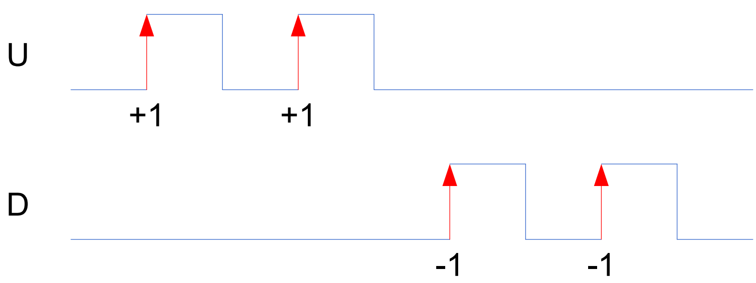
	Single-phase Bidirectional Counter					A/B Phase Counter				
	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
Up/Down counting state	M8246	M8247	M8248	M8249	M8250	M8251	M8252	M8253	M8254	M8255

3) Description of counting modes

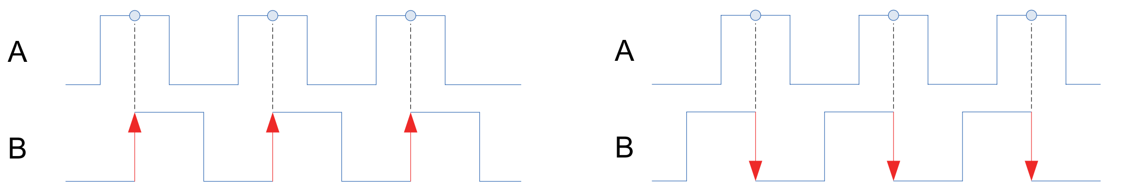
Single-phase unidirectional counter: With only one pulse signal counting input, the corresponding M element for up/down counter control decides on counting up/down. Some counters also have hardware reset and startup signal input ports.



Single-phase bidirectional counter: Two pulse signal counting inputs function as an up/down counter pulse input respectively. Some counters also have hardware reset and startup signal inputs. The up/down counting state of a counter is monitored by reading the special M element for up/down counting state.



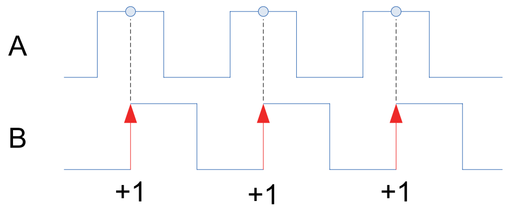
A/B phase counter: A phase and B phase signals with 90° phase difference are used as inputs. The relationship between A phase and B phase determines the counted direction. The up/down counting state of a counter is monitored by reading the special M element for up/down counting state.



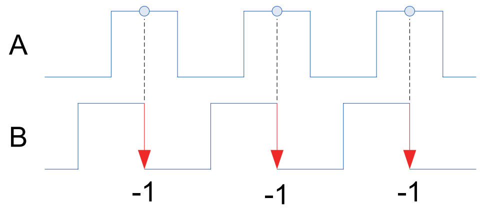
A phase leading B phase, counting up B phase leading A phase, counting down

By enabling quadruplicated frequency with the special M element, A/B phase counters can count at a fundamental or quadruplicated frequency.

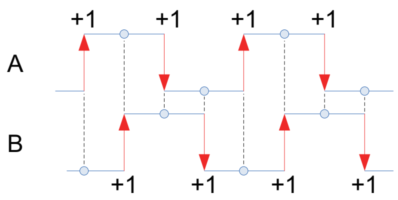
	A/B Phase Counter				
	C251	C252	C253	C254	C255
Enabling quadruplicated frequency	M8195	M8196	M8197	M8198	M8199



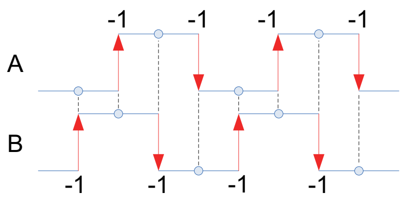
Counting up at a doubled frequency



Counting down at a doubled frequency



Counting up at a quadruplicated frequency



Counting down at a quadruplicated frequency

- The number of a high-speed counter used corresponds to X port. That is, when a high-speed counter Cxxx is specified, the corresponding X input port is specified. Therefore, X port cannot be reused; otherwise, an error will be reported. For example, when C252 uses X0 and X1 input ports, such ports cannot be reused by C235, C236, C243, C246, and C251. The interrupt or pulse capture corresponding to such input ports cannot be used.

4) Instructions for use of counters

- High-speed counters use hardware for counting based on the transition edge of relevant signals, and provide real-time responses, independent of the scan duration of the PLC.
- When the present value of a high-speed counter reaches the set value, for immediate output and processing, execute high-speed pulse comparison instructions, such as HSCS, HSCR, and HSZ. For details, see the description of instructions.
- When the present value of a high-speed counter reaches the set value, for immediate logical processing, execute the high-speed pulse comparison instruction HSCS, and specify the instruction operation as I0 x 0 interrupt (x = interrupt numbers 1–8), provided that subprograms corresponding to interrupt numbers must have been programmed.
- The software filter time of high-speed input signals can be set by setting element to D8021 and the time unit to 250 ns. The default value of D8021 is 1, so the default high-speed filter time is 250 ns. The value range of D8021 is 1 to 100, so the high-speed filter time range is 0.25 to 25 us.

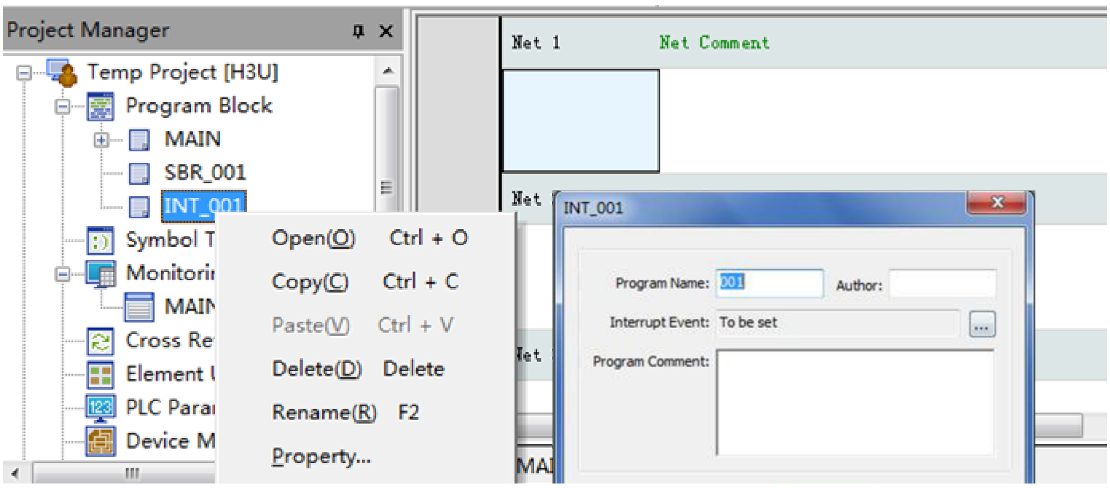
5.1.2 Input Interrupts

Input interrupts include interrupts on the rising/falling edge and counter interrupts. Interrupt numbers (Ixxx) are shown below:

Interrupt on the Rising/Falling Edge			Counter Interrupt
H3U Standard Model			
Port	Rising Edge	Falling Edge	
X00	I001	I000	I010 I020 I030 I040 I050 I060 I070 I080
X01	I101	I100	
X02	I201	I200	
X03	I301	I300	
X04	I401	I400	
X05	I501	I500	
X06	I561	I560	
X07	I571	I570	

1) Use of interrupts:

Interrupts should be used with interrupt subprograms. Choose interrupt events in the attribute of an interrupt subprogram, that is, set the interrupt number. In case of "Enable Interrupts", when the set interrupt events occur, the PLC system suspends normal execution of the main program (remember the current pause point), starts the execution of the interrupt subprogram from the entry address specified by I, returns to the pause point after completion, and continues to execute the main program. As the PLC system gives a high priority of response to interrupt signals, interrupts are independent of the scan duration.



2) Interrupts on the rising/falling edge:

X0 to X7 of the PLC can be separately set to interrupt input ports, each with interrupts on the rising/falling edge indicated by the interrupt number. For example, "I100" indicates the interrupt on the falling edge of X1 port, and "I101" indicates the interrupt on the rising edge of X1 port.

Counter interrupts: Based on the comparison result of the built-in high-speed counter, the PLC system executes the interrupt subprogram (HSCS) and gives priority to control of counting results. High-speed counter interrupt is used when the target output of the HSCS instruction is set to I010–I080.

To use the interrupt function, program corresponding interrupt subprograms and turn on the corresponding "Enable Interrupts" flag before interrupt response. The "Enable Interrupts" flag is shown below:

Settings of Enable/Disable Interrupts			
M8050	Enable/Disable I00x interrupts	X input interrupts: 16 interrupts correspond to interrupts on the rising/ falling edge of X0 to X7 ports. x = 1: interrupt on the rising edge x = 0: interrupt on the falling edge	Each flag bit corresponds to Enable/Disable Interrupt control of one external input: OFF: enable X input interrupts ON: disable X input interrupts
M8051	Enable/Disable I10x interrupts		
M8052	Enable/Disable I20x interrupts		
M8053	Enable/Disable I30x interrupts		
M8054	Enable/Disable I40x interrupts		
M8055	Enable/Disable I50x interrupts		
M8076	Enable/Disable I56x interrupts		
M8077	Enable/Disable I57x interrupts		
M8059	Enable/Disable counter interrupts	Enable/Disable counter interrupts	OFF: Enable counter interrupts ON: Disable counter interrupts

After the "Enable Interrupts" flag corresponding to each interrupt is turned on, the "Enable Global Interrupts" flag must also be turned on. That is, the interrupt function can be enabled only after EI instruction (FNC04) is executed. If the "Disable Global Interrupts" DI instruction (FNC05) is executed, all interrupt responses are disabled. When the "Enable Interrupt" flag corresponding to an input number is turned on and the input signal complies with interrupt settings, the corresponding interrupt subprogram will be executed.

For the detailed instruction for use, see "Chapter 1 Overview" on page 20.

5.1.3 Pulse Capture

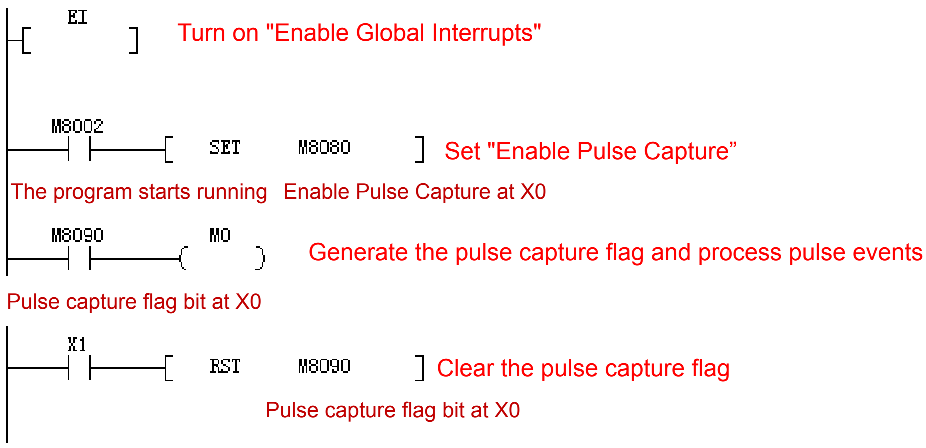
To respond to instant pulse signals at input ports without special requirements on the response time, use the pulse capture function. The PLC will store signals on the rising edge of the input port in M8090–M8097. These signals can be used as the basis for judgment and processing in the main program and manually cleared after the completion of response.

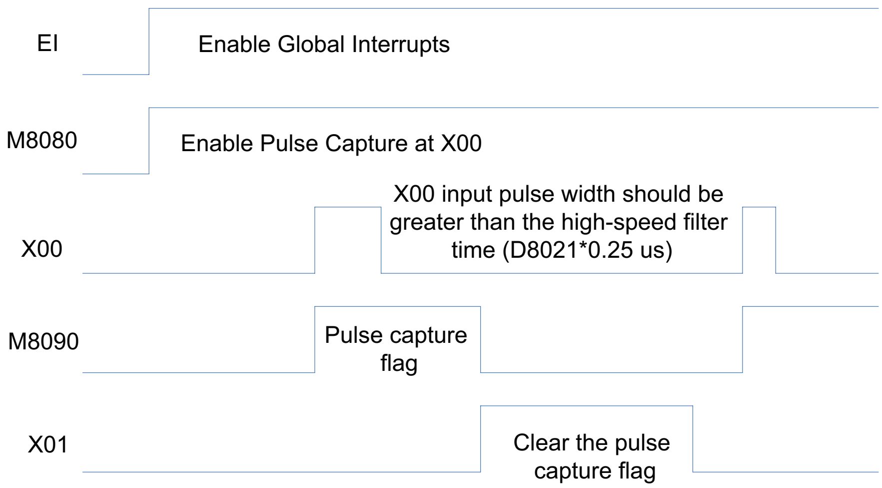
To use the pulse capture function, turn on the "Enable Global Interrupts" EI instruction and the corresponding "Enable Pulse Capture". After the rising edge is triggered on external input signals, turn on the corresponding pulse capture flag. The corresponding "Enable Pulse Capture" and pulse capture flags of each input port are shown below:

Input Port	H3U Standard Model	X00	X01	X02	X03	X04	X05	X06	X07
	Enable Pulse Capture	M8080	M8081	M8082	M8083	M8084	M8085	M8086	M8087
	Pulse Capture Flag	M8090	M8091	M8092	M8093	M8094	M8095	M8096	M8097

When "Enable Pulse Capture" is set to ON, the pulse capture function is enabled. When "Enable Pulse Capture" is set to OFF, the pulse capture function is disabled.

The following example describes the pulse capture function for the X00 input.





In this program, turn on the "Enable Global Interrupts", and set Enable Pulse Capture M8080 to ON at X00. When the external X00 input switches from OFF to ON, set the pulse capture flag M8090 interrupt to ON. Process pulse capture events based on the status of M8090, and then reset the pulse capture flag in the program to facilitate the next pulse capture response.

NOTE

- To successfully use the pulse capture function, the corresponding input signal pulse width must be greater than the high-speed filter time. That is, the pulse width must be greater than D8021*0.25 us.

5.2 High-speed Pulse Comparison Instructions of H3U Standard Model

Main instructions:

Instruction	Function
HSCS	(High-speed counter) Comparison setting
HSCR	(High-speed counter) Comparison reset
HSZ	(High-speed counter) Range comparison
SPD	Pulse density detection

5.2.1 HSCS Comparison Setting

◆ Overview

Compare the present value of the counter with the comparison value. If the values are equal, immediately set the comparison output, independent of the scan duration.

HSCS S1 S2 D			(High-speed Counter) Comparison Setting	Applicable Model: H3U	
S1	Source data	Set comparison value: 32 bits			32-bit instruction (13 steps) DHSCS: continuous execution
S2	Source data	Specified high-speed counters: C235-C255			
D	Result	Storage unit for the comparison result			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	Counter interrupt number					KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions:

When the present value of [S2] counter is equal to the set value of [S1], immediately set [D].

[S2] variables must be high-speed counters C235-C255. As all counters involved are 32-bit counters, the 32-bit instruction DHSCS must be used.

[D] is the storage unit for the comparison result and can also be used to call counter interrupt subprograms: when it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

When [D] is I010 to I080, 0-7 input interrupt subprograms of the high-speed counter are called. The corresponding interrupt subprograms must be programmed and the corresponding "Enable Interrupts" and "Enable Global Interrupts" flags must be turned on before timer interrupts are triggered. When M8059 is set to ON, all high-speed counter interrupts (I010-I080) are disabled.

NOTE

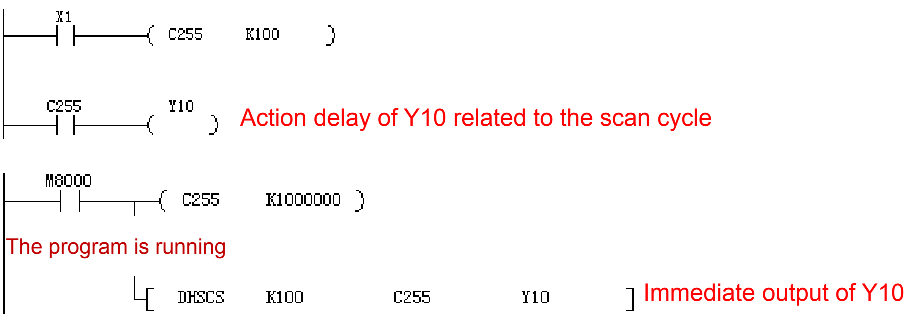
The difference between Y outputs of common instructions and Y outputs of the DHSCS instruction is as follows: (Example 1)

- When the present value of C255 changes from 99 to 100, C255 contact is immediately connected. When the instruction is executed at OUT Y10, Y10 will still be affected by the scan cycle and the value will be output after the program execution and I/O refresh are finished.
- When the present value of C255 changes from 99 to 100 or from 101 to 100, the DHSCS instruction on Y10 is immediately output to the external output in interrupt mode, which is independent of the PLC scan cycle but still affected by output delays of the output module relay (10 ms) or transistor (10 us).

1) Instruction for use:

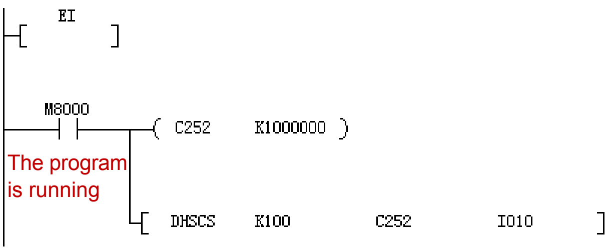
- Before the HSCS instruction is executed, the counter used must have been enabled (see Example 1); otherwise, the value of the counter will not change.
- The counter responds to input signals in interrupt mode and timely compare values. If the compared values are matched, the comparison output is immediately set. In Example 1, when the present value of C255 changes from 99 to 100 or from 101 to 100, Y10 is set immediately and remains in that state. Even if values of C255 and K100 are not equal by comparison, Y10 remains ON, unless there is an additional reset operation.
- The comparison output of the instruction only depends on the comparison result at the pulse input. Without the pulse input, even if the DMOV or DADD instruction is executed to rewrite the content of C235-C255 high-speed counters, the comparison output will not change. Flows driven by instructions cannot simply change the comparison result.
- When the target output of the HSCS instruction is counter interrupts I010-I080, each interrupt number can be used for only once rather than reused. See the previous section for settings and use of counter interrupts.
- Like common instructions, HSCS, HSCR, and HSZ can be executed repeatedly, but there should be less than eight simultaneously active instructions. Only one HSZ instruction in special mode (high-speed table comparison mode or frequency control mode) can be active.

2) Example 1:

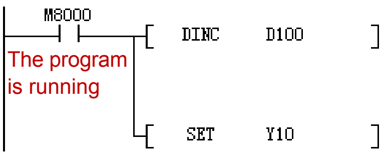


3) Example 2:

- Main program:



- I010 interrupt subprogram



The D operand range of the DHSCS instruction can also be specified to I0x0 (x=1-8). When the counter reaches the set value, interrupt routines are triggered.

If M8059 is set to ON, all high-speed counter interrupts are disabled.

NOTE

Differences of the ON signal on D operand with I010 or Y, M, or S outputs:

- With Y output: when the present value of C252 changes from 99 to 100 or from 101 to 100, Y is set to ON immediately and remains ON. Even if values of C252 and K100 are not equal by comparison, Y remains ON, unless there is an additional reset operation.
- With I010: When the present value of C252 changes from 99 to 100 or from 101 to 100, I010 will trigger only one interrupt.

5.2.2 HSCR Comparison Reset

◆ Overview

Compare the present value of the counter with the comparison value. If the values are equal, immediately reset the comparison output, independent of the scan duration.

HSCR S1 S2 D		(High-speed counter) Comparison reset	Applicable Model: H3U	
S1	Source data	Set comparison value: 32 bits		32-bit instruction (13 steps) DHSCR: Continuous execution
S2	Source data	Specified high-speed counters: C235–C255		
D	Result	Storage unit for the comparison result		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions:

When the present value of S2 counter is equal to the value of S1, immediately reset [D].

S2 variables must be high-speed counters C235–C255. As all counters involved are 32-bit counters, the 32-bit instruction DHSCR must be used.

[D] is the storage unit for the comparison result: when it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; when it is M, S, or SM variable, the result is immediately refreshed.

1) Note:

Except that the HSCR instruction cannot use high-speed counter interrupts as comparison outputs, the operation principle of the HSCR instruction is the same as that of the HSCS instruction. The comparison output action of the HSCR instruction is just the opposite of that of the HSCS instruction. That is, when the present value of the counter is equal to the set value, the specified output is reset. For the instruction for use, see "5.4.2 HSCS Comparison Setting" on page 358.

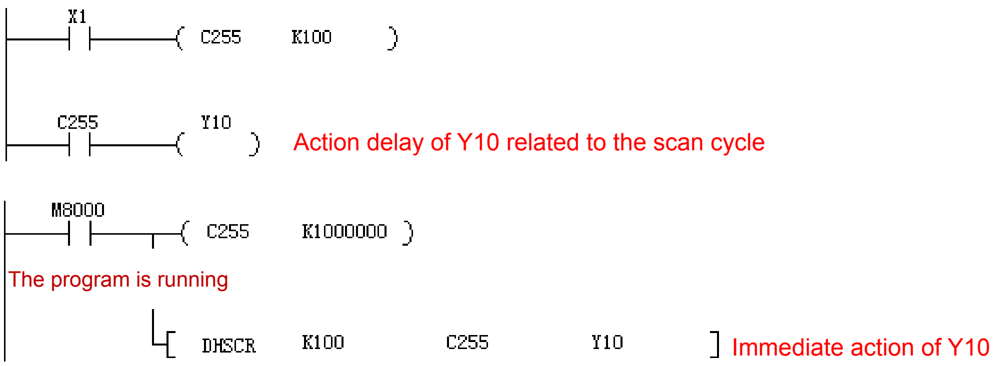
The difference between Y outputs of common instructions and Y outputs of the DHSCR instruction is as follows: (Example 1)

- When the present value of C255 changes from 99 to 100, C255 contact is immediately connected. When the instruction is executed at OUT Y10, Y10 will still be affected by the scan cycle and the value will be output after the program execution and I/O refresh are finished.
- When the present value of C255 changes from 99 to 100 or from 101 to 100, the DHSCS instruction at Y10 is immediately output to the external output in interrupt mode, which is independent of the PLC scan cycle but still affected by output delays of the output module relay (10 ms) or transistor (10 us).

2) Instruction for use:

- Before the HSCR instruction is executed, the counter used must have been enabled (see Example 1); otherwise, the value of the counter remains unchanged.
- The counter responds to input signals in interrupt mode and timely compare values. If the compared values are matched, the comparison output is immediately reset. In Example 1, when the present value of C255 changes from 99 to 100 or from 101 to 100, Y10 is reset immediately and remains in that state. Even if values of C255 and K100 are not equal by comparison, Y10 remains OFF, unless there is an additional set operation.
- The comparison output of the instruction only depends on the comparison result at the pulse input. Without the pulse input, even if the DMOV or DADD instruction is executed to rewrite the content of C235–C255 high-speed counters, the comparison output remains unchanged. Flows driven by instructions cannot simply change the comparison result.
- Like common instructions, HSCS, HSCR, and HSZ can be executed repeatedly, but there should be less than eight simultaneously active instructions. Only one HSZ instruction in special mode (high-speed table comparison mode or frequency control mode) can be active.

3) Example:



5.2.3 HSZ Range Comparison

◆ Overview

Compare the present value of the counter with the comparison value. If the values are equal, immediately reset the comparison output, independent of the scan duration.

HSZ S1 S2 S D			(High-speed counter) Range Comparison	Applicable Model: H3U
S1	Source data	Lower limit		32-bit instruction (17 steps) DHSZ: Continuous execution
S2	Source data	Upper limit		
S	Source data	Specified high-speed counters: C235–C255		
D	Result	Storage unit for the comparison result		

5

◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation						Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions:

Compare the present value of the counter [S] with the set comparison range [S1] to [S2], and immediately output the comparison result to three units starting with [D]. In the instruction:

[S1] is the lower limit of the comparison range (32 bits), and its value must be no greater than the value of [S2], that is, [S1] ≤ [S2].

[S2] is the upper limit of the comparison range (32 bits), and its value must be no smaller than the value of [S1], that is, [S1] ≤ [S2].

[S] variables must be high-speed counters C235–C255. As all counters involved are 32-bit counters, the 32-bit instruction DHSZ must be used.

[D] is the head address of three consecutive storage units for the comparison result: when it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

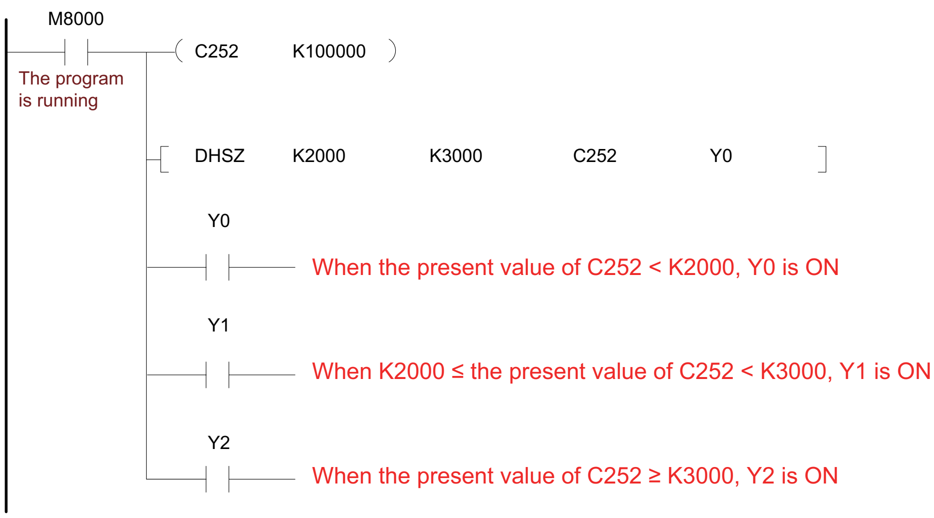
1) Note:

- The action principle of this instruction is similar to that of HSCS and HSCR instructions. The difference lies in that two comparison values are used and the comparison output uses three consecutive address units. See the instruction for use of "5.2.2 HSCR Comparison Reset" on page 344.
- The HSZ instruction operates in interrupt mode. Only when corresponding inputs of the counter have count pulses, comparison is carried out and corresponding outputs are refreshed.

- When [D] is set to special auxiliary relay M8130, the instruction is in high-speed table comparison mode, and variables of the instruction will be resolved in table mode.
- When [D] is set to the special auxiliary relay M8132, the instruction is in frequency control mode. In combination with DPLSY instruction, it can control the output frequency of DPLSY by the present value of high-speed counter.
- Like common instructions, HSCS, HSCR, and HSZ can be executed repeatedly, but there should be less than eight simultaneously active instructions. Only one HSZ instruction in special mode (high-speed table comparison mode or frequency control mode) can be active.

2) Example:

a) Common mode



b) High speed table comparison mode

When the instruction parameter [D] is set to special auxiliary relay M8130, the instruction is in high-speed table comparison mode. Notes to operands:

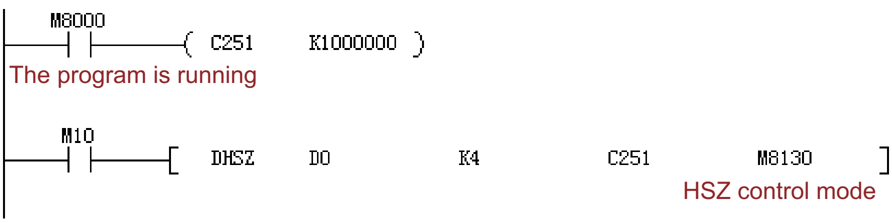
[S1] only corresponds to variables of register D and indicates the head address of the comparison table. Z can be used. After the instruction is enabled, [S1] will no longer be affected by Z.

[S2] can use the constant K or H only to indicate the number of rows of the table. Z can be used. After the instruction is enabled, [S1] will no longer be affected by Z.

[S] variables must be high-speed counters C235–C255.

When [D] is set to M8130, the instruction is in high-speed table comparison mode.

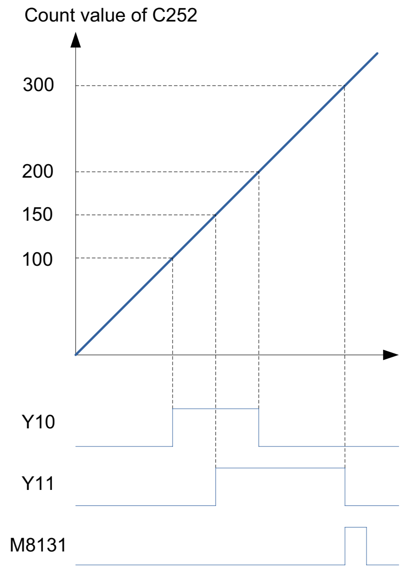
- Example: The following describes instruction programming.



- The following is an equivalent comparison table:

The initial variable of [S1] table is D0	Comparison Value (32 Bits) (High-order, Low-order)	Y Output Number	ON/OFF	Table Counter D8130
The number of rows of [S2] table is K4	(D1, D0)	D2	D3	0
	(D5, D4)	D6	D7	1
	(D9, D8)	D10	D11	2
	(D13, D12)	D14	D15	3
Parameter example	K100	H10	K1	When the instruction is executed, the counter follows the cycle from 0 to 1 to 2 to 3 to 0.
	K150	H11	K1	
	K200	H10	K0	
	K300	H11	K0	
Description	Operate after receipt of the set pulse value.	H10 indicates Y10 port. H11 indicates Y11 port.	K1 indicates ON. K0 indicates OFF.	

- Notes to actions:



When the present value of a high-speed counter C251 specified by [S] is equal to the set value (D1, D0), the Y output specified by D2 copies the state of OFF (D3 = K0) or ON (D3 = K1) and remains in that state. The action of the Y output is processed completely in interrupt mode.

When the present value of C251 is equal to the first group of set values, D8130 = K1. When it is equal to the second group of set values, D8130 = K2. When comparison operations are performed successively till the end of the last comparison action, M8131 = ON. After a scan cycle, D8130 is cleared and compared with the first group of set values again.

When the condition contact M10 of the instruction is turned OFF, execution of the instruction is interrupted, D8130 is cleared, but all output states related to the instruction remain unchanged. When the instruction is scanned for the first time and the user program is executed, settings of the comparison table are defined. Parameter settings of the table should be completed before execution of the instruction.

The table comparison instruction can be used in the user program for only once. The instruction can be used in combination with the HSCS, HSCR, or HSZ instruction for other purposes, but there should be less than eight simultaneously active instructions.

c) Frequency control mode

When the instruction parameter [D] is set to special auxiliary relay M8132, the instruction is in frequency control mode. In combination with the DPLSY instruction, it can control the output frequency of DPLSY by the present value of high-speed counter. Notes to operands:

[S1] only corresponds to variables of register D and indicates the head address of the comparison table. V can be used. After the instruction is enabled, [S1] will no longer be affected by V.

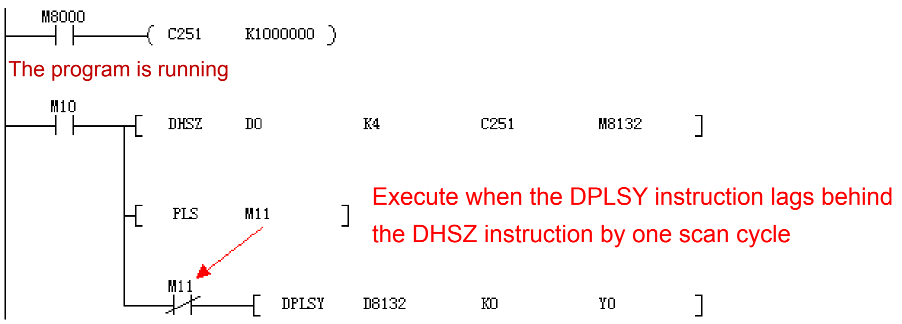
[S2] can use the constant K or H only to indicate the number of rows of the table. 1 ≤ (K or H) ≤ 128. V can be used. After the instruction is enabled, [S1] will no longer be affected by V or Z.

[S] variables must be high-speed counters C235–C255.

[D] is set to M8132 and indicates frequency control mode.

The instruction can be used in the user program for only once. Registers in the table should be preset.

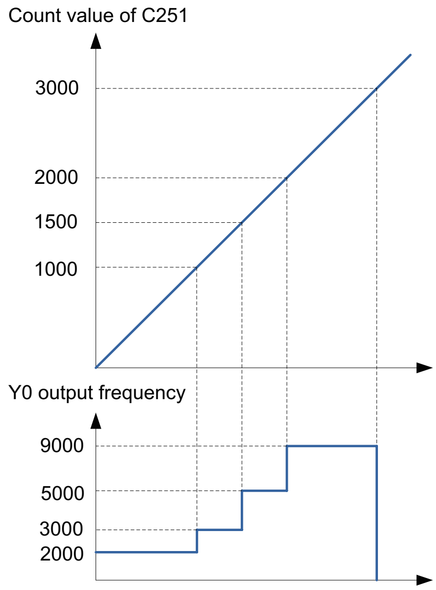
- Example: The following describes instruction programming.



The program controls the operation mode of Y0 output frequency based on the present counting value of C251. The following is the equivalent comparison and output frequency table.

The initial variable of [S1] table is D0	Comparison Value (32 Bits) (High-order, Low-order)	Y0 Output Frequency	Table Counter D8131
Number of rows of [S2] table indicated by K5	(D1, D0)	(D3, D2)	0
	(D5, D4)	(D7, D6)	1
	(D9, D8)	(D11, D10)	2
	(D13, D12)	(D15, D14)	3
	(D17, D16)	(D19, D18)	4
Parameter example	K1000	K2000	When the instruction is executed, the counter follows the cycle from 0 to 1 to 2 to 3 to 4 to 0.
	K1500	K3000	
	K2000	K5000	
	K3000	K9000	
	K0	K0	
Description	After receiving pulses, immediately carry out comparison. When they are matched (for example, C251 equal to 1000), Y0 output frequency will change.	The output frequency of Y0 output is changed to the set value in the corresponding column of the table.	

- Notes to actions:



The preset data is written into data registers constituting the table and the high-speed counter C251 specified by [S] is started by the instruction. Do not modify settings of the table during operation.

When 0 ≤ C251 < (D1, D0), the Y0 output frequency is equal to the value of (D3, D2);

When (D1, D0) ≤ C251 < (D5, D4), the Y0 output frequency is equal to the value of (D7, D6);

When (D5, D4) ≤ C251 < (D9, D8), the Y0 output frequency is equal to the value of (D11, D10);

When (D9, D8) ≤ C251< (D13, D12), the Y0 output frequency is equal to the value of (D15, D14);

When (D13, D12) ≤ C251 < (D17, D16), the Y0 output frequency is equal to the value of (D19, D18);

And so on.

After the operation in the last row, the complete flag M8133 turns ON, and operation in the first row is repeated.

To end operation in the last row, set the frequency of the last row to K0. When the driving coil M10 is set to OFF, the pulse output is turned OFF and the table counter D8131 is reset.

The above table takes effect after end of the first scan of program. Therefore, use [PLS M11] instruction to make the PLSY instruction execute from the second scan cycle after the driving coil M10 is set to ON.

- Note

In frequency control mode, the PLSY instruction and PLSR instruction executed during programming cannot obtain two pulse outputs at the same time.

5.2.4 SPD Pulse Density Detection

◆ Overview

The number of pulses at the specified port is detected within the set time for pulse frequency detection.

SPD S1 S2 D		Pulse density detection	Applicable Model: H3U
S1	Source data	Specified pulse signal input port	16-bit instruction (7 steps) SPD: Continuous execution
S2	Source data	Set pulse detection duration	
D	Result	Count of pulses	

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions:

The detected number of pulses by [S1] port in duration time [S2] is stored in address [D].

[S1] indicates the pulse signal input port, which is specified within the range X00 to X07.

[S2] indicates the pulse detection duration (ms), which is specified within the range 1 to 32,767.

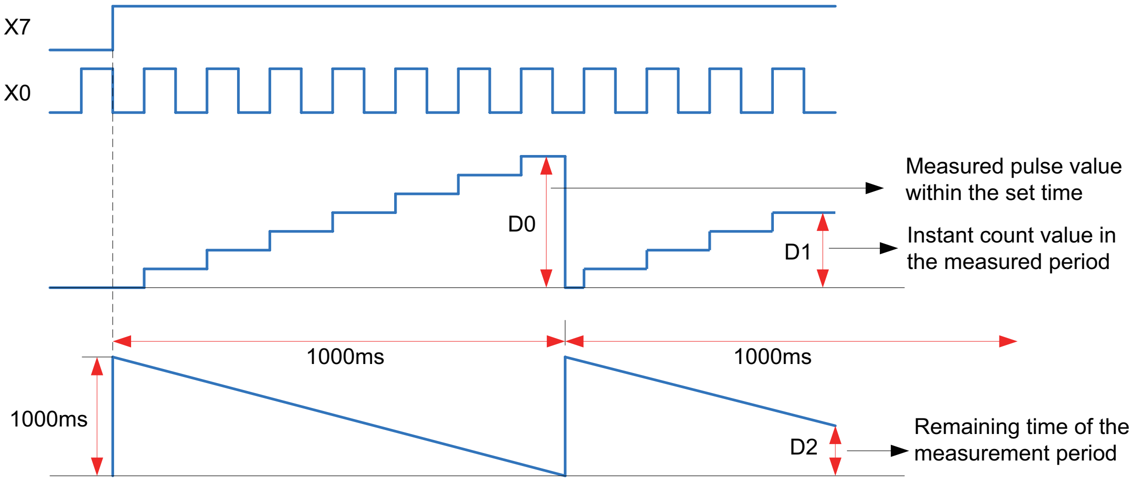
[D] indicates the count of pulses, occupying three consecutive address units starting with [D]. [D + 0] indicates the number of pulses within the set duration [S2]. [D + 1] indicates the real-time number of pulses. [D + 2] indicates remaining time of the sampling period.

1) Note:

X00 to X07 ports used for the SPD instruction can be used for high-speed counters or interrupt inputs.

2) Example:

- Notes to actions:



In the figure above, when X7 is set to ON, D1 counts the X0 action which is from OFF to ON state, and stores the result to D0 after 1000 ms. Then D1 is reset and counts again the X0 action. D2 is used to measure the remaining time.

Therefore, the pulse frequency can be calculated according to the D0 and [S2] setting value. If the pulse signal is measured by rotary encoder, the speed can be calculated.

5.3 H3U-PM Motion Control Model

5.3.1 High-speed Counter

H3U-PM model has three-channel high-speed inputs. Each channel has two differential inputs, which correspond to the PLC inputs Ax+/-, Bx+/- (x: 1, 2, and 3 indicating X, Y, and Z axes respectively). The three high-speed counters integrated within the PLC correspond to three-channel input counters as follows:

High-speed Counter	C252	C253	C254
Input Channel	X axis	Y axis	Z axis

1) Configuration of the Input Mode

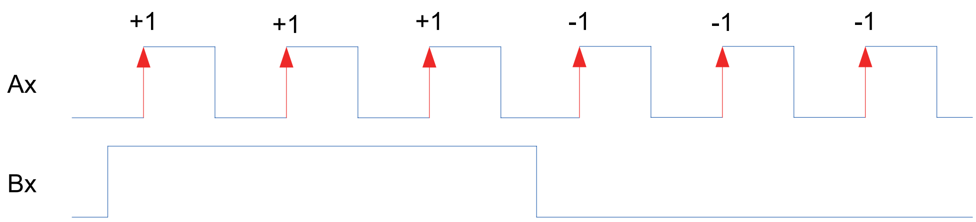
H3U-PM high-speed counters support pulse+ direction, A/B phase, and CW/CCW high-speed pulse counting by configuration of the input mode of input channels with special SD elements.

X axis	SD60	0: Pulse+Direction 1: A/B phase 2: CW/CCW
Y axis	SD160	
Z axis	SD260	

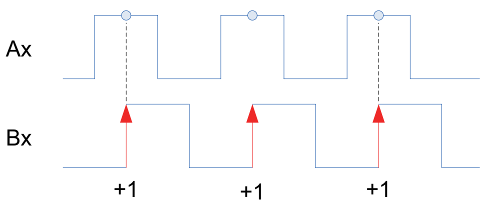
When the input mode is set to A/B phase, by enabling quadruplicated frequency, A/B phase counters can count at a doubled or quadruplicated frequency.

	A/B Phase Counter		
	C252	C253	C254
Enabling quadruplicated frequency	M8196	M8197	M8198

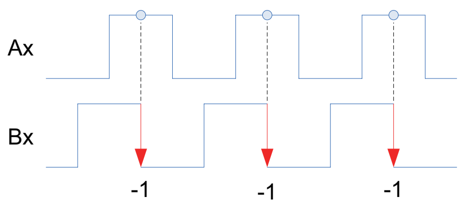
2) Notes to the Input Mode



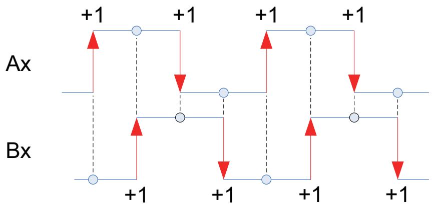
Pulse+ direction counting



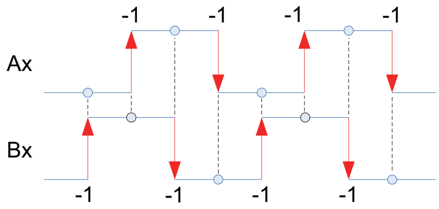
A/B phase counting up at a doubled frequency



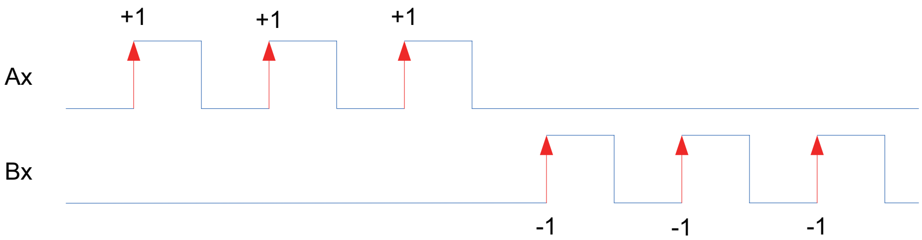
A/B phase counting down at a doubled frequency



A/B phase counting up at a quadruplicated frequency



A/B phase counting down at a quadruplicated frequency



CW/CCW counting

5

3) Use of counters

- High-speed counters use hardware for counting based on the transition edge of relevant signals, and provide real-time responses, independent of the scan duration of the PLC.
- When the present value of a high-speed counter reaches the set value, for immediate output and processing, execute high-speed pulse comparison instructions, such as HSCS, HSCR, and HSZ. For details, see the interpretation of instructions.
- When the present value of a high-speed counter reaches the set value, for immediate logical processing, execute the high-speed pulse comparison instruction HSCS, and specify the instruction operation to I0x0 interrupt (x = interrupt numbers 1–8), provided that subprograms corresponding to interrupt numbers must have been programmed.
- The software filter time of high-speed input signals can be set by setting element to D8021 and the time unit to 250 ns. The default value of D8021 is 1, so the default high-speed filter time is 250 ns. The value range of D8021 is 1 to 100, so the high-speed filter time range is 0.25 to 25 us.

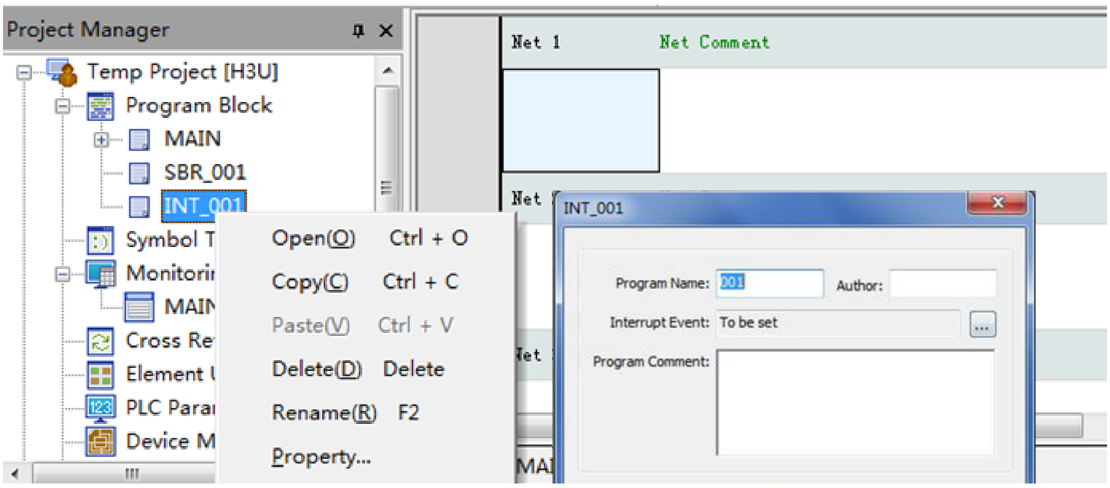
5.3.2 Input Interrupts

Input interrupts include interrupts on the rising/falling edge and counter interrupts. The interrupt numbers (Ixxx) are shown below:

Interrupts on the Rising/Falling Edge			Counter Interrupts
Port	Rising Edge	Falling Edge	I010 I020 I030 I040 I050 I060 I070 I080
X-axis PG0	I001	I000	
Y-axis PG1	I101	I100	
Z-axis PG2	I201	I200	

1) Use of Interrupts:

Interrupts should be used with interrupt subprograms. Choose interrupt events in the attribute of an interrupt subprogram. That is, set the interrupt number. In case of "Enable Interrupts", when the set interrupt events occur, the PLC system suspends normal execution of the main program (remember the current pause point), starts the execution of the interrupt subprogram from the entry address specified by I, returns to the pause point after completion, and continues to execute the main program. As the PLC system gives a high priority of response to interrupt signals, interrupts are independent of the scan duration.



2) Interrupts on the Rising/Falling Edge:

PG0 to PG2 of the PLC can be separately set to interrupt input ports, each with interrupt on the rising/falling edge indicated by the interrupt number. For example, "I100" indicates the interrupt on the falling edge of PG1 port, and "I101" indicates the interrupt on the rising edge of PG1 port.

Counter interrupts: Based on the comparison result of the built-in high-speed counter, the PLC system executes the interrupt subprogram (HSCS) and gives priority to control of counting results. High-speed counter interrupt is used when the target output of the HSCS instruction is set to I010–I080.

To use the interrupt function, program corresponding interrupt subprograms and turn on the corresponding "Enable Interrupts" flag before interrupt response. "Enable Interrupts" flag is shown below:

Settings of Enable/Disable Interrupts			
M8050	Enable/Disable I00x Interrupts	PG input interrupts: Six interrupts correspond to interrupts on the rising/ falling edge of PG0 to PG2 ports. x = 1: interrupt on the rising edge x = 0: interrupt on the falling edge	Each flag bit corresponds to Enable/ Disable Interrupt control of one external input. OFF: enable X input interrupts ON: disable X input interrupts
M8051	Enable/Disable I10x Interrupts		
M8052	Enable/Disable I20x Interrupts		
M8059	Enable/Disable Counter Interrupts	Enable/Disable Counter Interrupts	OFF: Enable Counter Interrupts ON: Disable Counter Interrupts

After the "Enable Interrupts" flag corresponding to each interrupt is turned on, the "Enable Global Interrupts" flag must also be turned on. That is, the interrupt function can be enabled only after EI instruction (FNC04) is executed. If the "Disable Global Interrupts" DI instruction (FNC05) is executed, all interrupt responses are disabled. When the "Enable Interrupt Setting" flag corresponding to an input number is turned on and the input signal complies with interrupt settings, the corresponding interrupt subprogram will be executed.

For the detailed instruction for use, see 11 "Interrupt Subroutine."

5.3.3 Pulse Capture

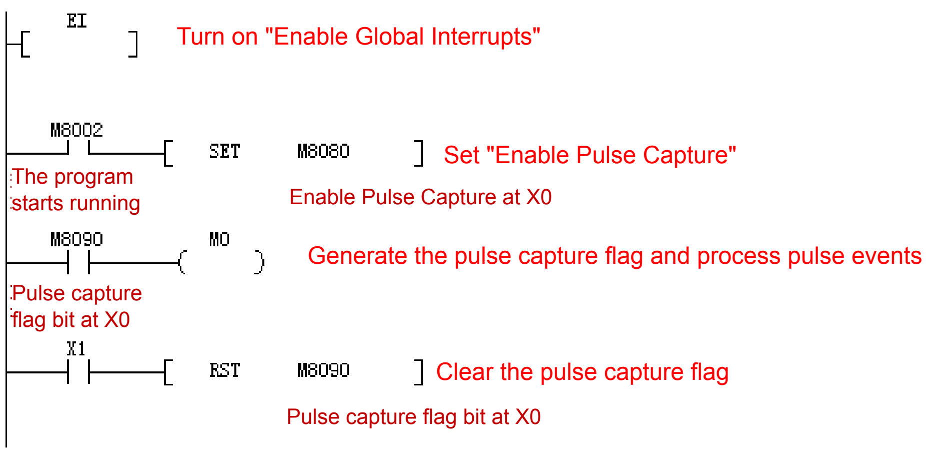
To respond to instant pulse signals at input ports without special requirements on the response time, use the pulse capture function. The PLC will store signals on the rising edge of the input port in M8090-M8092. These signals can be used as the basis for judgment and processing in the main program and manually cleared after the completion of response.

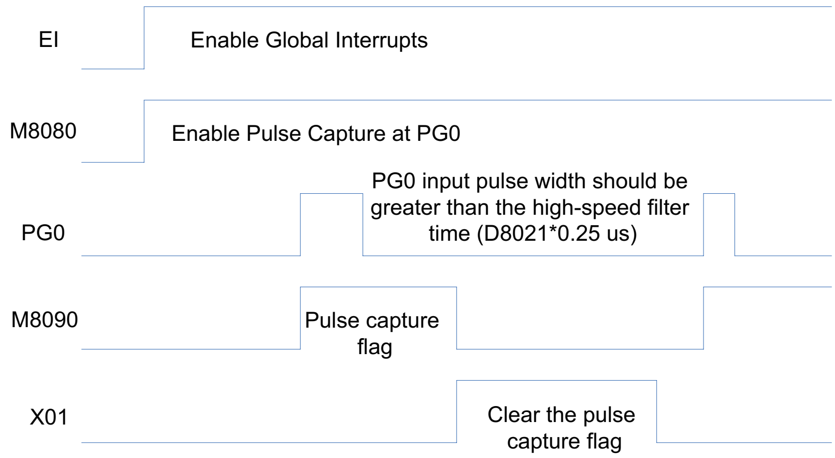
To use the pulse capture function, turn on the "Enable Global Interrupts" EI instruction and the corresponding "Enable Pulse Capture". After the rising edge is triggered on external input signals, turn on the corresponding pulse capture flag. The corresponding "Enable Pulse Capture" and pulse capture flags of each input port are shown below:

Input Port	H3U-PM Motion Control Model	X-axis PG0	Y-axis PG1	Z-axis PG2
	Enable Pulse Capture	M8080	M8081	M8082
	Pulse Capture Flag	M8090	M8091	M8092

When "Enable Pulse Capture" is set to ON, the pulse capture function is enabled. When "Enable Pulse Capture" is set to OFF, the pulse capture function is disabled.

The following example describes the pulse capture function for the PG0 input.





In this program, turn on the "Enable Global Interrupts", and set Enable Pulse Capture M8080 to ON at PG0. When the external PG0 input switches from OFF to ON, set the pulse capture flag M8090 interrupt to ON. Process pulse capture events based on the status of M8090, and then reset the pulse capture flag in the program to facilitate the next pulse capture response.

NOTE

- To successfully use the pulse capture function, the corresponding input signal pulse width must be greater than the high-speed filter time. That is, the pulse width must be greater than D8021*0.25 us.

5.4 High-speed Comparison Instructions for H3U-PM Motion Control Model

Compared with H3U standard model, H3U-PM model has more functions: its HSCS instruction has an additional function of starting electronic cams, and HSOS and HSOR instructions are added to execute the interrupt comparison setting/reset output of the high-speed output value or cam value, execute the counter interrupt subprogram, and start cams.

Main instructions:

Instruction	Function
HSCS	(High-speed counter) Comparison setting
HSCR	(High-speed counter) Comparison reset
HSZ	(High-speed counter) Range comparison
HSOS	High-speed interrupt comparison setting
HSOR	High-speed interrupt comparison reset

5.4.1 Operation Mode of High-speed Comparison Instructions

Instruction and usage mode of H3U-PM motion control model:

Instruction	Comparison Object	Comparison Result Output
HSCS	High-speed Counter	Y, M, and S bit elements
		I010-I080 counter interrupts
		Starting electronic cams
HSCR	High-speed Counter	Y, M, and S bit elements
		C counter
HSZ	High-speed Counter	Y, M, and S bit elements
		Output frequency value
HSOS	High-speed output value	Y, M, and S bit elements
		I010-I080 counter interrupts
		Starting electronic cams
	Electronic cam value	Y, M, and S bit elements
		I010-I080 counter interrupts
		Starting electronic cams
HSOR	High-speed output value	Y, M, and S bit elements
	Electronic cam value	Y, M, and S bit elements

5

5.4.2 HSCS Comparison Setting

◆ Overview

By high-speed counter interrupt comparison setting, the value of the counter is compared with the comparison value. If the values are equal, the PLC system immediately sets bit elements, enables counter interrupts, and starts electronic cams, independent of the scan cycle.

HSCS S1 S2 D		(High-speed counter) Comparison setting	Applicable Model: H3U-PM	
S1	Source data	Set comparison value: 32 bits		32-bit instruction (13 steps) HSCS: continuous execution
S2	Source data	Specified high-speed counters: C252-C254		
D	Result	Storage unit for the comparison result		

◆ Operands

Operand	Bit Soft Component							Word Soft Component															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	High-speed counter interrupt number SM60, SM160, SM260					KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Functions and actions:

[D] is immediately set when the present value of [S2] counter is equal to the set value of [S1].

[S2] variables must be high-speed counters C252–C254. As all counters involved are 32-bit counters, the 32-bit instruction DHSCS must be used.

C252 and other 32-bit high-speed counters:

Set value	Remarks
C252	Corresponding to X axis
C253	Corresponding to Y axis
C254	Corresponding to Z axis

[D] is the storage unit for the comparison result and can also be used to call counter interrupt subprograms and start electronic cams.

When it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

When [D] is I010 to I080, 0–7 input interrupt subprograms of the high-speed counter are called. The corresponding interrupt subprograms must be programmed and the corresponding "Enable Interrupts" and "Enable Global Interrupts" flags must be turned on before timer interrupts are triggered. When M8059 is set to ON, all high-speed counter interrupts (I010–I080) are disabled.

When [D] is SM60, SM160, or SM260, electronic cams are immediately started provided that the corresponding cam configuration (see 8 Electronic Cam) must have been programmed.

Set value	Interrupt output result
Y, M, and S	Bit element output
I interrupts	High-speed counter interrupts
SM60	Starting X-axis electronic cam
SM160	Starting Y-axis electronic cam
SM260	Starting Z-axis electronic cam

For details about starting electronic cams, see “8.4 Starting Electronic Cams or Electronic Gears” on page 568.

1) Note:

- Like common instructions, DHSCS, DHSCR, and DHSZ can be executed repeatedly, but there should be less than eight simultaneously active instructions. Only one DHSZ instruction in special mode (high-speed table comparison mode or frequency control mode) can be active.
- Comparison objects of DHSCS, DHSCR, DHSZ, DHSOS, and DHSOR instructions (see the description of comparison objects below) include high-speed counters, position values of high-speed outputs, and position values of electronic cams. Comparison objects of high-speed interrupt comparison instructions driven by one axis must be consistent; otherwise, errors will be reported. The following describes comparison objects:

Interrupt Comparison Object 1	Interrupt Comparison Object 2	Interrupt Comparison Object 3	Remarks
C252 (DHSCS, DHSCR, and DHSZ instructions are driven)	K0 (DHSOS and DHSOR instructions are driven)	K11 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C253 (DHSCS, DHSCR, and DHSZ instructions are driven)	K1 (DHSOS and DHSOR instructions are driven)	K12 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C254 (DHSCS, DHSCR, and DHSZ instructions are driven)	K2 (DHSOS and DHSOR instructions are driven)	K13 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.

For example, if the DHSOS instruction is used by X axis to compare the output value of X axis (K0), the comparison object of the other simultaneously active instruction must be the output value of X axis (K0) and can be neither the position value of the electronic cam (K11) nor the high-speed counter comparison instruction (for C252).

NOTE

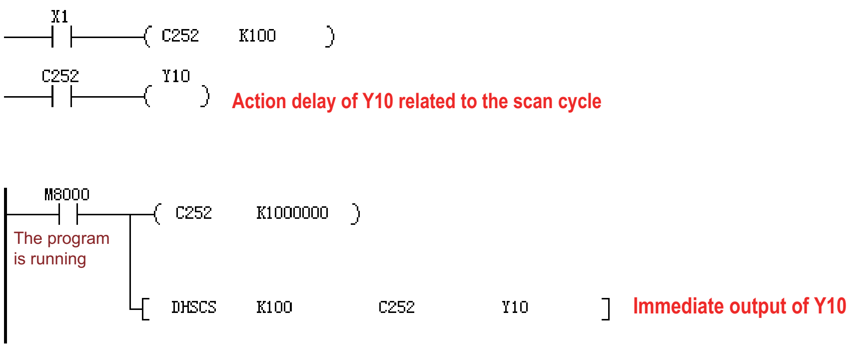
Differences between Y outputs of common instructions and Y outputs of the DHSCS instruction is as follows: (Example 1)

- When the present value of C252 changes from 99 to 100, C252 contact is immediately connected. When the instruction is executed at OUT Y10, Y10 will still be affected by the scan cycle and the value will be output after the program execution and I/O refresh are finished.
- When the present value of C252 changes from 99 to 100 or from 101 to 100, the DHSCS instruction at Y10 is immediately output to the external output in interrupt mode, which is independent of the PLC scan cycle but still affected by output delays of the output module relay (10 ms) or transistor (10 us).

2) Instruction for use:

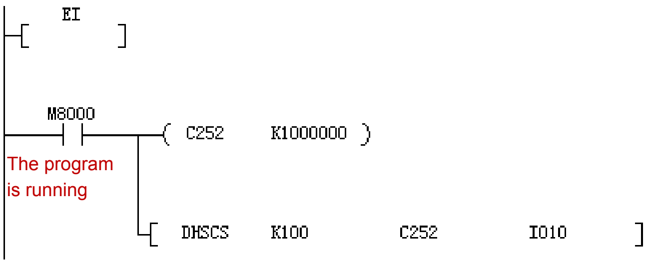
- Before the HSCS instruction is executed, the counter used must have been enabled (see Example 1); otherwise, the value of the counter remains unchanged.
- The counter responds to input signals in interrupt mode and timely compare values. If the compared values are matched, the comparison output is immediately set. In Example 1, when the present value of C252 changes from 99 to 100 or from 101 to 100, Y10 is set immediately and remains in that state. Even if values of C252 and K100 are not equal by comparison, Y10 remains ON, unless there is an additional reset operation.
- The comparison output of the instruction only depends on the comparison result at the pulse input. Without the pulse input, even if the DMOV or DADD instruction is executed to rewrite the content of C252–C254 high-speed counters, the comparison output remains unchanged. Flows driven by instructions cannot simply change the comparison result.
- When the target output of the HSCS instruction is counter interrupts I010–I080, each interrupt number can be used for only once rather than reused. See the previous section for settings and use of counter interrupts.

3) Example 1:

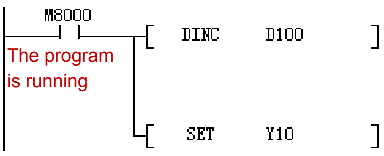


4) Example 2:

- Main program:



- I010 interrupt subprogram



The D operand range of the DHSCS instruction can also be specified as I0x0 (x = 1–8). When the counter reaches the set value, interrupt routines are triggered.

If M8059 is set to ON, all high-speed counter interrupts are disabled.

NOTE

Differences of the ON signal on D operand with I010 or Y, M, or S outputs:

- With Y output: When the present value of C252 changes from 99 to 100 or from 101 to 100, Y is set to ON immediately and remains ON. Even if values of C252 and K100 are not equal by comparison, Y remains ON, unless there is an additional reset operation.
- With I010: When the present value of C252 changes from 99 to 100 or from 101 to 100, I010 will trigger only one interrupt.

5.4.3 HSCR Comparison Reset

◆ Overview

The present value of the counter is compared with the comparison value. If the values are equal, the comparison output is immediately reset, independent of the scan duration.

HSCR S1 S2 D			(High-speed counter) Comparison reset	Applicable Model: H3U-PM
S1	Source data	Set comparison value: 32 bits		32-bit instruction (13 steps) DHSCR: Continuous execution
S2	Source data	Specified high-speed counters: C252–C254		
D	Result	Storage unit for the comparison result		

◆ Operands

5

Operand	Bit Soft Component							Word Soft Component															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions:

When the present value of S2 counter is equal to the value of S1, immediately reset [D].

[S2] variables must be high-speed counters C252-C254. As all counters involved are 32-bit counters, the 32 bit instruction DHSCR must be used.

Set value	Remarks
C252	Corresponding to X axis
C253	Corresponding to Y axis
C254	Corresponding to Z axis

[D] is the storage unit for the comparison result: When it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; when it is M, S, or SM variable, the result is immediately refreshed.

Set value	Interrupt output result
Y, M, and S	Bit element output

1) Note:

Except that the HSCR instruction cannot use high-speed counter interrupts as comparison outputs, the operation principle of the HSCR instruction is the same as that of the HSCS instruction. The comparison output action of the HSCR instruction is just the opposite of that of the HSCS instruction. That is, when the present value of the counter is equal to the set value, the specified output is reset. See the instruction for use of HSCS.

The difference between Y outputs of common instructions and Y outputs of the DHSCR instruction is as follows: (Example 1)

- When the present value of C252 changes from 99 to 100, C252 contact is immediately connected. When the instruction is executed at OUT Y10, Y10 will still be affected by the scan cycle and the value will be output after the program execution and I/O refresh are finished.
- When the present value of C252 changes from 99 to 100 or from 101 to 100, the DHSCR instruction at Y10 is immediately output to the external output in interrupt mode, which is independent of the PLC scan cycle but still affected by output delays of the output module relay (10 ms) or transistor (10 us).

2) Instruction for use:

- Before the HSCR instruction is executed, the counter used must have been enabled (see Example 1); otherwise, the value of the counter remains unchanged.
- The counter responds to input signals in interrupt mode and timely compares values. If the compared values are matched, the comparison output is immediately reset. In Example 1, when the present value of C252 changes from 99 to 100 or from 101 to 100, Y10 is reset immediately and remains in that state. Even if values of C252 and K100 are not equal by comparison, Y10 remains OFF, unless there is an additional set operation.
- The comparison output of the instruction only depends on the comparison result at the pulse input. Without the pulse input, even if the DMOV or DADD instruction is executed to rewrite the content of C252–C254 high-speed counters, the comparison output remains unchanged. Flows driven by instructions cannot simply change the comparison result.

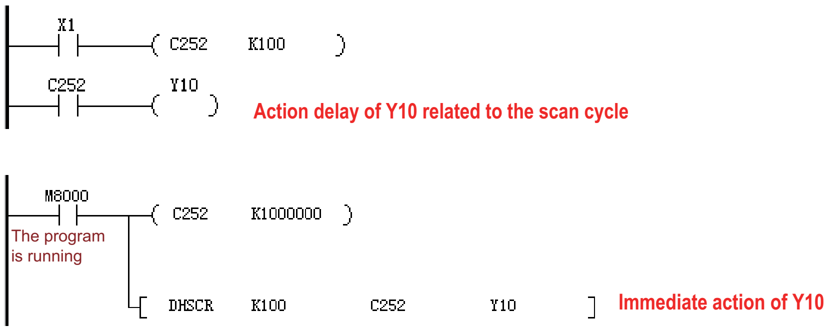
3) Note:

- Like common instructions, DHSCS, DHSCR, and DHSZ can be executed repeatedly, but there should be less than eight simultaneously active instructions. Only one DHSZ instruction in special mode (high-speed table comparison mode or frequency control mode) can be active.
- Comparison objects of DHSCS, DHSCR, DHSZ, DHSOS, and DHSOR instructions (see the description of comparison objects below) include high-speed counters, position values of high-speed outputs, and position values of electronic cams. Comparison objects of high-speed interrupt comparison instructions driven by one axis must be consistent; otherwise, errors will be reported. The following describes comparison objects:

Interrupt Comparison Object 1	Interrupt Comparison Object 2	Interrupt Comparison Object 3	Remarks
C252 (DHSCS, DHSCR, and DHSZ instructions are driven)	K0 (DHSOS and DHSOR instructions are driven)	K11 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C253 (DHSCS, DHSCR, and DHSZ instructions are driven)	K1 (DHSOS and DHSOR instructions are driven)	K12 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C254 (DHSCS, DHSCR, and DHSZ instructions are driven)	K2 (DHSOS and DHSOR instructions are driven)	K13 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.

For example, if the DHSOS instruction is used by X axis to compare the output value of X axis (K0), the comparison object of the other simultaneously active instruction must be the output value of X axis (K0) and can be neither the position value of the electronic cam (K11) nor the high-speed counter comparison instruction (for C252).

4) Example:



5.4.4 HSZ Range Comparison

◆ Overview

The present value of the counter is compared with the comparison value. If the values are equal, the comparison output is immediately reset, independent of the scan duration.

5

HSZ S1 S2 S D			(High-speed counter) Range comparison	Applicable model: H3U-PM
S1	Source data	Lower limit		32-bit instruction (17 steps) DHSZ: Continuous execution
S2	Source data	Upper limit		
S	Source data	Specified high-speed counters: C252-C254		
D	Result	Storage unit for the comparison result		

◆ Operands

Operand	Bit Soft Component							Word Soft Component																
	System·User							System·User					Bit Designation						Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions:

Compare the present value of the counter [S] with the set comparison range [S1] to [S2], and immediately output the comparison result to three units starting with [D].

[S1] is the lower limit of the comparison range (32 bits), and its value must be no greater than the value of [S2], that is, [S1] ≤ [S2].

[S2] is the upper limit of the comparison range (32 bits), and its value must be no smaller than the value of [S1], that is, [S1] ≤ [S2].

[S] variables must be high-speed counters C252–C254. As all counters involved are 32-bit counters, the 32-bit instruction DHSZ must be used.

[D] is the head address of three consecutive storage units for the comparison result: when it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

1) Note:

- Like common instructions, DHSCS, DHSCR, and DHSZ can be executed repeatedly, but there should be less than eight simultaneously active instructions. Only one DHSZ instruction in special mode (high-speed table comparison mode or frequency control mode) can be active.
- Comparison objects of DHSCS, DHSCR, DHSZ, DHSOS, and DHSOR instructions (see the description of comparison objects below) include high-speed counters, position values of high-speed outputs, and position values of electronic cams. Comparison objects of high-speed interrupt comparison instructions driven by one axis must be consistent; otherwise, errors will be reported. The following describes comparison objects:

Interrupt Comparison Object 1	Interrupt Comparison Object 2	Interrupt Comparison Object 3	Remarks
C252 (DHSCS, DHSCR, and DHSZ instructions are driven)	K0 (DHSOS and DHSOR instructions are driven)	K11 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C253 (DHSCS, DHSCR, and DHSZ instructions are driven)	K1 (DHSOS and DHSOR instructions are driven)	K12 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C254 (DHSCS, DHSCR, and DHSZ instructions are driven)	K2 (DHSOS and DHSOR instructions are driven)	K13 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.

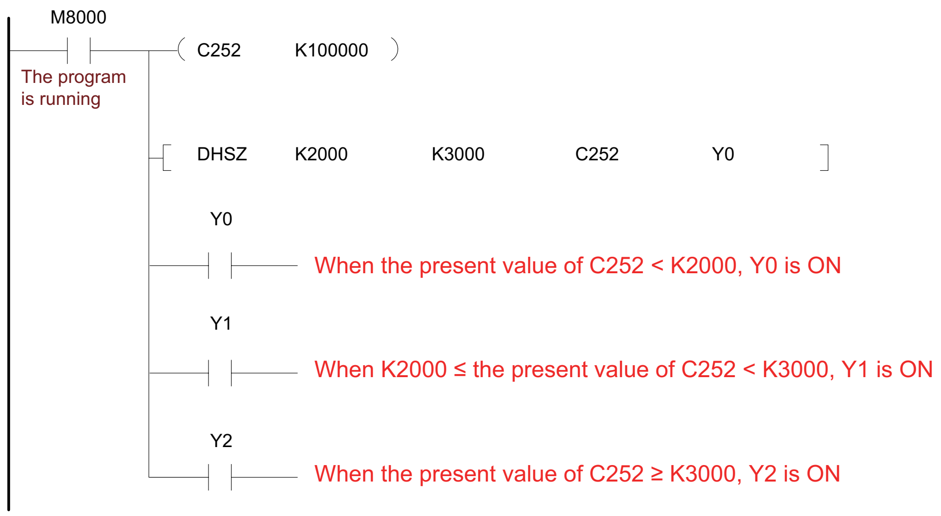
For example, if the DHSOS instruction is used by X axis to compare the output value of X axis (K0), the comparison object of the other simultaneously active instruction must be the output value of X axis (K0) and can be neither the position value of the electronic cam (K11) nor the high-speed counter comparison instruction (for C252).

2) Note:

- The action principle of this instruction is similar to that of HSCS and HSCR instructions. The difference lies in that two comparison values are used and the comparison output uses three consecutive address units. See the instruction for use of HSCR.
- The HSZ instruction operates in interrupt mode. Only when corresponding inputs of the counter have count pulses, comparison is carried out and corresponding outputs are refreshed.
- When [D] is set to special auxiliary relay M8130, the instruction is in high-speed table comparison mode, and variables of the instruction will be resolved in table mode.
- When [D] is set to the special auxiliary relay M8132, the instruction is in frequency control mode. In combination with DPLSY instruction, it can control the output frequency of DPLSY by the present value of high-speed counter.

3) Example:

a) Common mode



b) High speed table comparison mode

5

When the instruction parameter [D] is set to special auxiliary relay M8130, the instruction is in high-speed table comparison mode. Notes to operands:

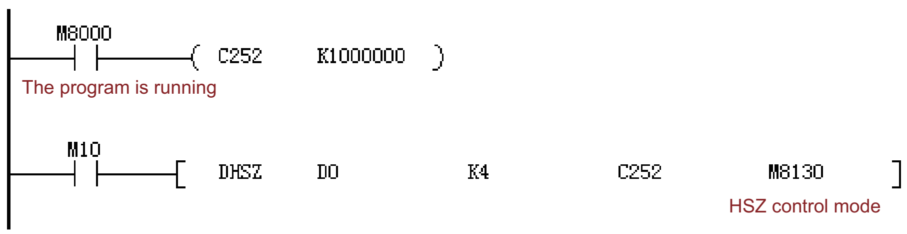
[S1] only corresponds to variables of register D and indicates the head address of the comparison table. Z can be used. After the instruction is enabled, [S1] will no longer be affected by Z.

[S2] can use the constant K or H only to indicate the number of rows of the table. Z can be used. After the instruction is enabled, [S1] will no longer be affected by Z.

[S] variables must be high-speed counters C252–C254.

When [D] is set to M8130, the instruction is in high-speed table comparison mode.

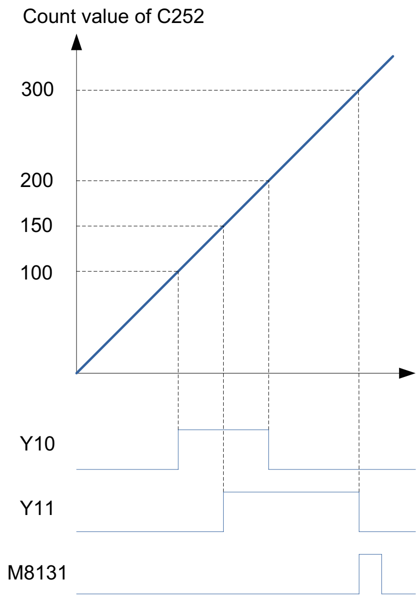
- Example: The following describes instruction programming.



- The following is an equivalent comparison table:

The initial variable of [S1] table is D0	Comparison Value (32 Bits) (High-order, Low-order)	Y Output Number	ON/OFF	Table Counter D8130
The number of rows of [S2] table is K4	(D1, D0)	D2	D3	0
	(D5, D4)	D6	D7	1
	(D9, D8)	D10	D11	2
	(D13, D12)	D14	D15	3
Parameter example	K100	H10	K1	When the instruction is executed, the counter follows the cycle from 0 to 1 to 2 to 3 to 0.
	K150	H11	K1	
	K200	H10	K0	
	K300	H11	K0	
Description	Operate after receipt of the set pulse value.	H10 indicates Y10 port. H11 indicates Y11 port.	K1 indicates ON. K0 indicates OFF.	

- Notes to actions:



When the present value of a high-speed counter C251 specified by [S] is equal to the set value (D1, D0), the Y output specified by D2 copies the state of OFF (D3 = K0) or ON (D3 = K1) and remains in that state. The action of the Y output is processed completely in interrupt mode.

When the present value of C251 is equal to the first group of set values, D8130 = K1. When it is equal to the second group of set values, D8130 = K2. When comparison operations are performed successively till the end of the last comparison action, M8131 = ON. After a scan cycle, D8130 is cleared and compared with the first group of set values again.

When the condition contact M10 of the instruction is turned OFF, execution of the instruction is interrupted, D8130 is cleared, but all output states related to the instruction remain unchanged. When the instruction is scanned for the first time and the user program is executed, settings of the comparison table are defined. Parameter settings of the table should be completed before execution of the instruction.

The table comparison instruction can be used in the user program for only once. The instruction can be used in combination with the HSCS, HSCR, or HSZ instruction for other purposes, but there should be less than eight simultaneously active instructions.

c) Frequency control mode

When the instruction parameter [D] is set to special auxiliary relay M8132, the instruction is in frequency control mode. In combination with the DPLSY instruction, it can control the output frequency of DPLSY by the present value of high-speed counter. Notes to operands:

[S1] only corresponds to variables of register D and indicates the head address of the comparison table. V can be used. After the instruction is enabled, [S1] will no longer be affected by V.

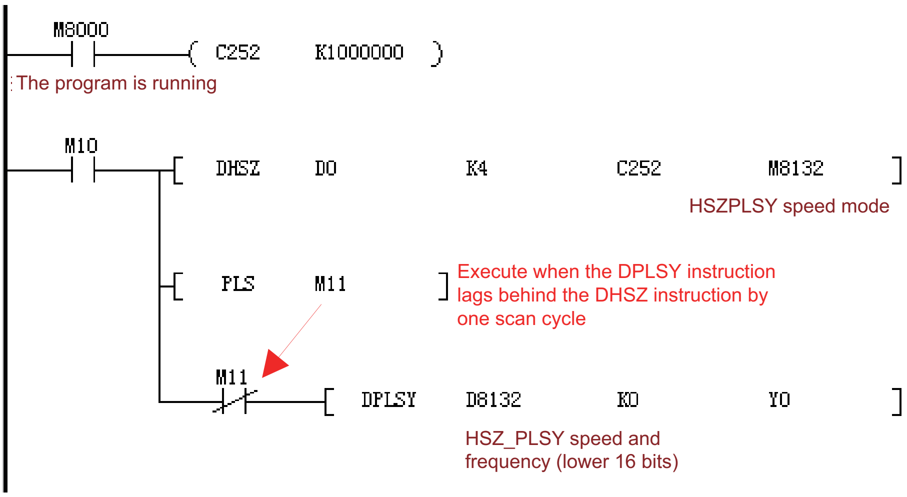
[S2] can use the constant K or H only to indicate the number of rows of the table. 1 ≤ (K or H) ≤ 128. V can be used. After the instruction is enabled, [S1] will no longer be affected by V or Z.

[S] variables must be high-speed counters C252-C254.

[D] is set to M8132 and indicates frequency control mode.

The instruction can be used in the user program for only once. Registers in the table should be preset.

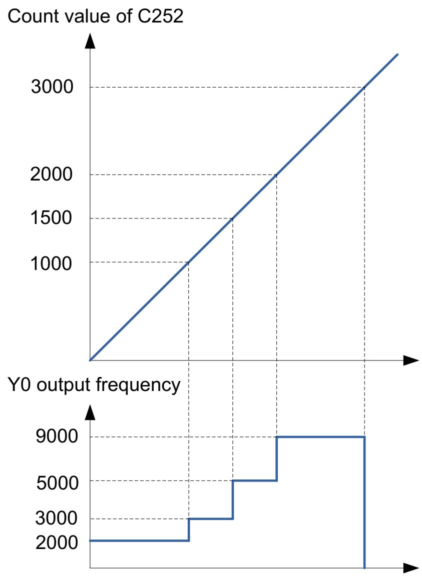
- Example: The following describes instruction programming.



The program controls the operation mode of Y0 output frequency based on the present counting value of C251. The following is the equivalent comparison and output frequency table.

The initial variable of [S1] table is D0	Comparison Value (32 Bits) (High-order, Low-order)	Y0 Output Number	Table Counter D8131
Number of rows of [S2] table indicated by K5	(D1, D0)	(D3, D2)	0
	(D5, D4)	(D7, D6)	1
	(D9, D8)	(D11, D10)	2
	(D13, D12)	(D15, D14)	3
	(D17, D16)	(D19, D18)	4
Parameter example	K1000	K2000	When the instruction is executed, the counter follows the cycle from 0 to 1 to 2 to 3 to 4 to 0.
	K1500	K3000	
	K2000	K5000	
	K3000	K9000	
	K0	K0	
Description	After receiving pulses, immediately carry out comparison. When they are matched (for example, C251 equal to 1000), Y0 output frequency will change.	The output frequency of Y0 output is changed to the set value in the corresponding column of the table.	

- Notes to actions:



The predetermined data is written into data registers constituting the table and the high-speed counter C251 specified by [S] is started by the instruction. Do not modify settings of the table during operation.

When 0 ≤ C251 < (D1, D0), the Y0 output frequency is equal to the value of (D3, D2);

When (D1, D0) ≤ C251 < (D5, D4), the Y0 output frequency is equal to the value of (D7, D6);

When (D5, D4) ≤ C251 < (D9, D8), the Y0 output frequency is equal to the value of (D11, D10);

When (D9, D8) ≤ C251< (D13, D12), the Y0 output frequency is equal to the value of (D15, D14);

When (D13, D12) ≤ C251 < (D17, D16), the Y0 output frequency is equal to the value of (D19, D18);

And so on.

After the operation in the last row, the complete flag M8133 turns ON, and operation in the first row is repeated.

To end operation in the last row, set the frequency of the last row to K0. When the driving coil M10 is set to OFF, the pulse output is turned OFF and the table counter D8131 is reset.

The above table takes effect after end of the first scan of program. Therefore, use [PLS M11] instruction to make the PLSY instruction execute from the second scan cycle after the driving coil M10 is set to ON.

- Note

In frequency control mode, the PLSY instruction and PLSR instruction executed during programming cannot obtain two pulse outputs at the same time.

5

5.4.5 DHSOS High-speed Interrupt Comparison Setting

◆ Overview

By high-speed counter interrupt comparison setting, the PLC system sets bit elements, enables counter interrupts, and starts electronic cams.

DHSOS S1 S2 D		Interrupt comparison setting	Applicable Model: H3U-PM	
S1	Target comparison value	Set interrupt comparison value		32-bit instruction (13 steps) Continuous execution
S2	Comparison object	High-speed output value and position values of electronic cams		
D	Output result	Output result of interrupt comparison		

◆ Operands

Operand	Bit Soft Component							Word Soft Component															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	High-speed counter interrupt number SM60, SM160, SM260					KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

When the present value of [S2] counter is equal to the set value of [S1], the PLC system sets bit elements, enables counter interrupts, and starts electronic cams.

[S2] variables must be the set value in the following table. As all objects involved are 32-bit elements, the 32-bit instruction DHSOS must be used.

Set Value	Interrupt Comparison Object	Remarks
K0	Position value of X axis (SD36, SD37)	Corresponding to X axis
K1	Position value of Y axis (SD136, SD137)	Corresponding to Y axis
K2	Position value of Z axis (SD236, SD237)	Corresponding to Z axis
K11	M-Pos value of X axis cam (SD48, SD49)	Corresponding to X axis
K12	M-Pos value of Y axis cam (SD148, SD149)	Corresponding to Y axis
K13	M-Pos value of Z axis cam (SD248, SD249)	Corresponding to Z axis
Others	Not supported	

[D] is the storage unit for the comparison result and can also be used to call counter interrupt subprograms and start electronic cams.

When it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

When [D] is I010 to I080, 0–7 input interrupt subprograms of the high-speed counter are called. The corresponding interrupt subprograms must be programmed and the corresponding "Enable Interrupts" and "Enable Global Interrupts" flags must be turned on before timer interrupts are triggered. When M8059 is set to ON, all high-speed counter interrupts (I010–I080) are disabled.

When [D] is SM60, SM160, or SM260, electronic cams are immediately started provided that the corresponding cam configuration must have been programmed.

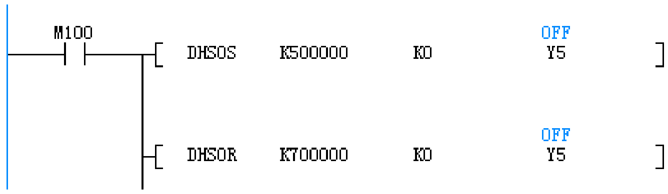
Set Value	Interrupt Output Result
Y, M, and S	Bit element output
I interrupts	High-speed counter interrupts
SM60	Starting X-axis electronic cam
SM160	Starting Y-axis electronic cam
SM260	Starting Z-axis electronic cam

For starting electronic cams, see Section 8.4.

1) Note:

- DHSOS and DHSOR can be used for multiple times, but there should be no more than two instructions driven by one axis (see the description of comparison objects below).

For example:

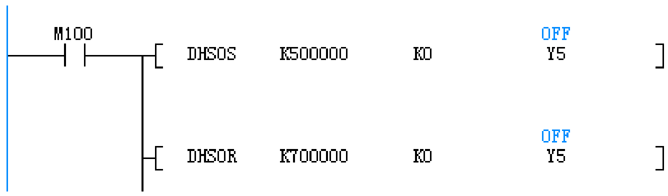


- Comparison objects of DHSCS, DHSCR, DHSZ, DHSOS, and DHSOR instructions (see the description of comparison objects below) include high-speed counters, position values of high-speed outputs, and position values of electronic cams. Comparison objects of high-speed interrupt comparison instructions driven by one axis must be consistent; otherwise, errors will be reported. The following describes comparison objects:

Interrupt Comparison Object 1	Interrupt Comparison Object 2	Interrupt Comparison Object 3	Remarks
C252 (DHSCS, DHSCR, and DHSZ instructions are driven)	K0 (DHSOS and DHSOR instructions are driven)	K11 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C253 (DHSCS, DHSCR, and DHSZ instructions are driven)	K1 (DHSOS and DHSOR instructions are driven)	K12 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C254 (DHSCS, DHSCR, and DHSZ instructions are driven)	K2 (DHSOS and DHSOR instructions are driven)	K13 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.

For example, if the DHSOS instruction is used by X axis to compare the output value of X axis (K0), the comparison object of the other simultaneously active instruction must be the output value of X axis (K0) and can be neither the position value of the electronic cam (K11) nor the high-speed counter comparison instruction (for C252).

2) Example:



As shown above, the comparison object is set to K0, indicating the present high-speed output value. When the present high-speed output value is 500K, set Y5; when the present high-speed output value is 700K, reset Y5.

5

5.4.6 DHSOR High-speed Interrupt Comparison Setting

◆ Overview

Bit elements are reset by high-speed interrupt comparison reset.

DHSOR S1 S2 D		Interrupt comparison reset	Applicable model: H3U-PM	
S1	Target comparison value	Set interrupt comparison value		32-bit instruction (13 steps) Continuous execution
S2	Comparison object	High-speed output value and position values of electronic cams		
D	Output result	Output result of interrupt comparison		

◆ Operands

Operand	Bit Soft Component							Word Soft Component															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

When the present value of [S2] counter is equal to the set value of [S1], bit elements are reset.

[S2] variables must be the set value in the following table. As all objects involved are 32-bit elements, the 32-bit instruction DHSOR must be used.

Set Value	Interrupt Comparison Object	Remarks
K0	Position value of X axis (SD36, SD37)	Corresponding to X axis
K1	Position value of Y axis (SD136, SD137)	Corresponding to Y axis
K2	Position value of Z axis (SD236, SD237)	Corresponding to Z axis
K11	M-Pos value of X axis cam (SD48, SD49)	Corresponding to X axis
K12	M-Pos value of Y axis cam (SD148, SD149)	Corresponding to Y axis
K13	M-Pos value of Z axis cam (SD248, SD249)	Corresponding to Z axis
Others	Not supported	

[D] is the storage unit for the comparison result:

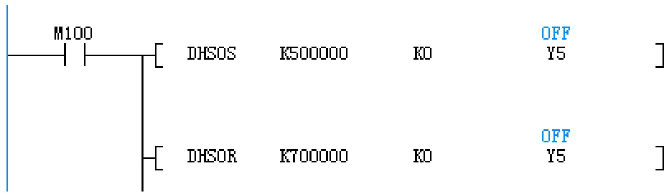
When it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

Set Value	Interrupt Output Result
Y, M, and S	Bit element output

1) Note:

- DHSOS and DHSOR can be used for multiple times, but there should be no more than two instructions driven by one axis (see the description of comparison objects below).

For example:

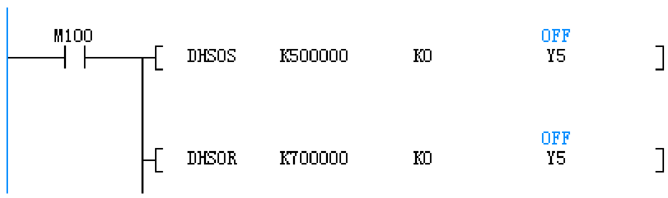


- Comparison objects of DHSCS, DHSCR, DHSZ, DHSOS, and DHSOR instructions (see the description of comparison objects below) include high-speed counters, position values of high-speed outputs, and position values of electronic cams. Comparison objects of high-speed interrupt comparison instructions driven by one axis must be consistent; otherwise, errors will be reported. The following describes comparison objects:

Interrupt Comparison Object 1	Interrupt Comparison Object 2	Interrupt Comparison Object 3	Remarks
C252 (DHSCS, DHSCR, and DHSZ instructions are driven)	K0 (DHSOS and DHSOR instructions are driven)	K11 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C253 (DHSCS, DHSCR, and DHSZ instructions are driven)	K1 (DHSOS and DHSOR instructions are driven)	K12 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.
C254 (DHSCS, DHSCR, and DHSZ instructions are driven)	K2 (DHSOS and DHSOR instructions are driven)	K13 (DHSOS and DHSOR instructions are driven)	The instructions cannot be driven simultaneously but at different time.

For example, if the DHSOS instruction is used by X axis to compare the output value of X axis (K0), the comparison object of the other simultaneously active instruction must be the output value of X axis (K0) and can be neither the position value of the electronic cam (K11) nor the high-speed counter comparison instruction (for C252).

2) Example:



As shown above, the comparison object is set to K0, indicating the present high-speed output value. When the present high-speed output value is 500K, set Y5; when the present high-speed output value is 700K, reset Y5.

6 Positioning and Interpolation

Chapter 6 Positioning and Interpolation

This chapter describes the H3U-supported positioning instructions in detail.

6.1 Overview

Both the positioning instructions and trajectory control are executed through application instructions in the H3U standard models. The positioning instructions have the following characteristics:

- The high-speed output frequency ranges from 1 to 200 kHz.
- Some positioning instructions support S-curve acceleration and deceleration.
- The 2-axis arc and linear interpolation is supported.

6.1.1 Table of Attributes of High-speed Output Instructions

NOTE

- √ indicates that the item has/supports the attribute, and the blank space indicates that the item does not have/support the attribute.

Attributes of high-speed output instructions are listed in the following table.

Instruction	Pulse Direction Output	Trapezoid Acceleration/ Deceleration	S-curve Acceleration/ Deceleration	Separate Acceleration/ Deceleration Setting	Frequency Modification Supported During Running	Pulse Count Modification Supported During Running	Direction Change During Running	Speed or Position Control	H3U Standard Model	H3U-PM Motion Control Model
PWM					√			Speed	√	
PLSY					√	√ (M)		Speed Position Speed+ Position	√	√
PLSV	√				√		√	Speed	√	√
PLSV2	√	√		√ (M)	√		√	Speed	√	√
ZRN		√		√ (M)			√	Speed	√	√
DSZR	√	√		√ (M)			√	Speed	√	
PLSR		√	√ (M)	√ (M)		√ (M)		Position	√	√
DRVA	√	√	√ (M)	√ (M)		√ (M)		Position	√	√
DRVI	√	√	√ (M)	√ (M)		√ (M)		Position	√	√
DVIT	√	√		√ (M)				Speed+ Position	√	
PLSN	√	√		√ (M)				Position	√	√
G90G01	√	√		√ (M)				Position	√	
G91G01	√	√		√ (M)				Position	√	
G90G02	√ (Fixed)	√		√ (M)			√	Position	√	
G91G02	√ (Fixed)	√		√ (M)			√	Position	√	
G90G03	√ (Fixed)	√		√ (M)			√	Position	√	
G91G03	√ (Fixed)	√		√ (M)			√	Position	√	

NOTE

- √ (Fixed) indicates that the pulse direction output port is fixed, and √ (M) indicates that the function can be used only after the special element is set.
- The acceleration/deceleration of H3U high-speed output instructions is determined by the instruction attribute, and is unrelated to the acceleration/deceleration time. For example, the PLSV instruction has no acceleration/deceleration attribute, and it makes no sense to modify its acceleration/deceleration time. The acceleration/deceleration of the positioning instruction ranges from 10 ms to 5000 ms (which is 10 ms to 500 ms for interpolation instructions). The upper/lower limit will be used when the value is outside the range.
- Instructions supported by the H3U-PM motion control model can be used in general main programs or sub- programs. In these instructions, Y0, Y1, and Y2 indicate the output control of the x-axis, y-axis, and z-axis.

6.1.2 Use of Special Elements

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high- speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430; default value: OFF) of special elements is used for setting and differentiation, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

Special registers and relays used in high-speed output instructions are defined as follows:

The special M elements are defined in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status
M8341	M8361	M8381	M8401	M8421	Valid output label for the DSZR/ZRN and other clearing signals
M8342	M8362	M8382	M8402	M8422	Designation of DSZR instruction zero return direction
M8343	M8363	M8383	M8403	M8423	Forward rotation limit
M8344	M8364	M8384	M8404	M8424	Reverse rotation limit
M8345	M8365	M8385	M8405	M8425	Near point signal logical inversion
M8346	M8366	M8386	M8406	M8426	Zero point signal logical inversion
M8347	M8367	M8387	M8407	M8427	S-curve acceleration/deceleration enabling
M8348	M8368	M8388	M8408	M8428	Reserved
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse
M8351	M8371	M8391	M8411	M8431	Port output initialization flag
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
M8353	M8373	M8393	M8413	M8433	Reserved
M8354	M8374	M8394	M8414	M8434	DSZR execution abnormal end flag bit
M8355	M8375	M8395	M8415	M8435	PLSV2 accelerating flag
M8356	M8376	M8396	M8416	M8436	PLSV2 decelerating flag
M8357	M8377	M8397	M8417	M8437	Reserved
M8358	M8378	M8398	M8418	M8438	Reserved
M8359	M8379	M8399	M8419	M8439	Reserved

The special D elements are defined in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8344	D8364	D8384	D8404	D8424	DSZR instruction zero return speed (Hz) (32-bit) [default value: 50,000]
D8345	D8365	D8385	D8405	D8425	
D8346	D8366	D8386	D8406	D8426	Creep speed (Hz) [default value: 2000]
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]
D8350	D8370	D8390	D8410	D8430	Element number clearing
D8351	D8371	D8391	D8411	D8431	Reserved
D8352	D8372	D8392	D8412	D8432	Reserved
D8353	D8373	D8393	D8413	D8433	Reserved

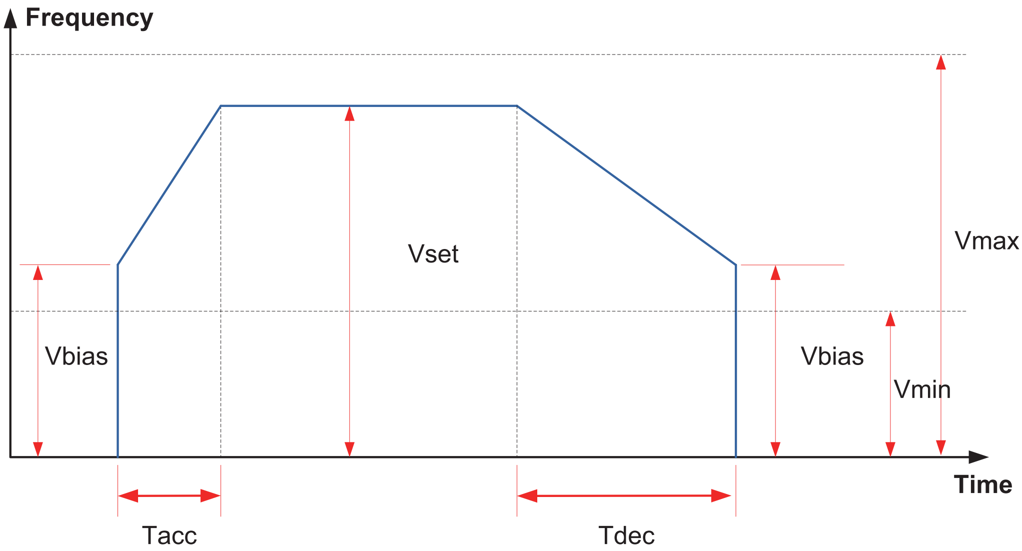
6

Y0	Y1	Y2	Y3	Y4	Attribute
D8354	D8374	D8394	D8414	D8434	Reserved
D8355	D8375	D8395	D8415	D8435	Reserved
D8356	D8376	D8396	D8416	D8436	Reserved
D8357	D8377	D8397	D8417	D8437	Reserved
D8358	D8378	D8398	D8418	D8438	Reserved
D8359	D8379	D8399	D8419	D8439	Reserved

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

6.1.3 Output Frequency-Time Relationship and Acceleration/Deceleration Process

The following figure describes the relationship between the output frequency and the time.



Vset indicates the pulse output frequency set manually using an instruction.

Vbias indicates the base output frequency set manually using a special register.

Vmax indicates the maximum frequency generally set through a special register.

Vmin indicates the calculated minimum frequency.

Tacc indicates the acceleration time.

Tdec indicates the deceleration time. Tdec equals Tacc by default. If the special function flag is set, the acceleration/deceleration time can be set separately for each axis.

Generally, Vmax is equal to or greater than Vset, and Vbias is equal to or greater than Vmin; otherwise, the frequency is adjusted. Vmax and Vmin specify the upper and lower limits of the pulse output frequency respectively.

The actual minimum output frequency Vmin is calculated according to the following formula.

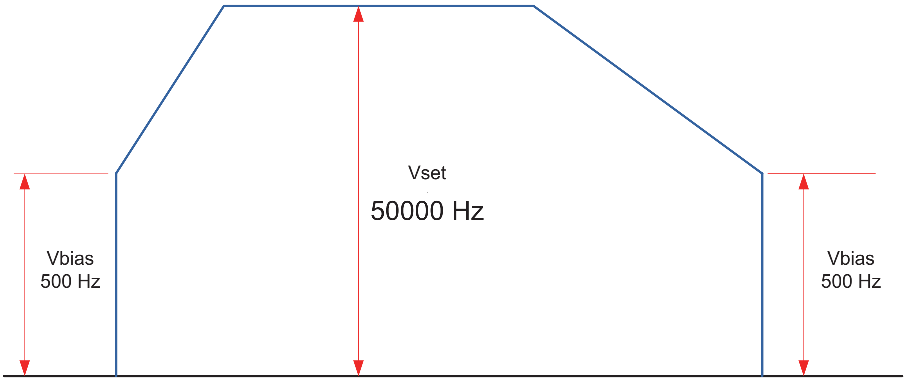
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

That is, even a value lower than the above result is specified, the calculated value is used. The frequency at the beginning and end of acceleration/deceleration shall not be lower than the calculated value.

For example, if the output frequency is 50,000 Hz, and the acceleration/deceleration time is 100 ms,

$$\sqrt{\frac{50000\ (Hz)}{2 \times 100\ (ms) / 1000}}$$

the calculated minimum frequency is 500 Hz. Even the specified base frequency is lower than the calculated value, the calculated value will be used.



6.2 Positioning Instruction

Pulse output	“PLSY: Pulse output”
Pulse positioning	“PLSV: Variable-speed pulse output”
	“PLSV2: Variable-speed pulse output with acceleration/ deceleration”
	“ZRN: Regression through the origin”
	“DSZR: DOG search return to origin”
Pulse output	“PLSR: Pulse output with acceleration/deceleration”
Pulse positioning	“DRVA: Absolute positioning”
	“DRVI: Relative positioning”
	“DVIT: Interrupt positioning”
	“PLSN: Multi-speed pulse output”

PLSY: Pulse output

◆ Overview

A specified number of pulses are output at the specified pulse frequency.

PLSY S1 S2 D		Pulse Output	Applicable model: H3U	
S1	Output frequency	Specified pulse output frequency	16-bit instruction (7 steps) PLSY: Continuous execution	32-bit instruction (13 steps) DPLSY: Continuous execution
S2	Pulse count	Specified number of pulse outputs		
D	Output port	High-speed pulse output port		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

6

◆ Functions and actions

Because the relay is not applicable to high-frequency actions, this instruction is applicable only to the PLC of the transistor output type. This instruction can be used to output S2 pulses at the S1 frequency through the port specified by D. After the pulses are sent, the M8029 flag is set. Wherein:

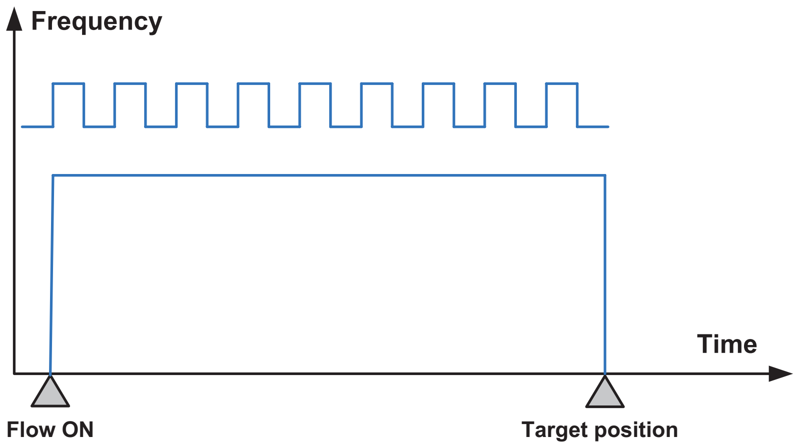
D indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

S1 indicates the designated output pulse frequency. For a 16-bit instruction (PLSY), the frequency ranges from 1 Hz to 32,767 Hz. For a 32-bit instruction (DPLSY), the frequency ranges from 1 Hz to 200,000 Hz (that is, 1 Hz to 200 kHz). The S1 value can be changed during instruction execution.

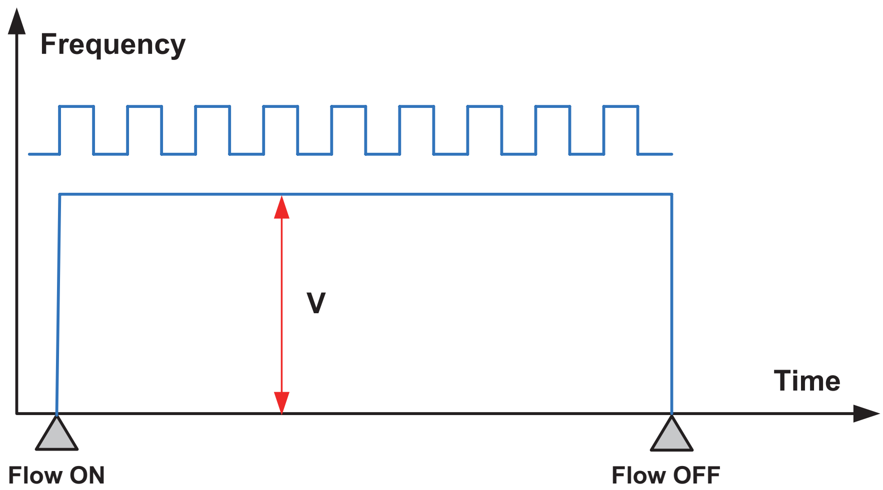
S2 indicates the number of the output pulses. For a 16-bit instruction (PLSY), the number ranges from 1 to 32,767. For a 32-bit instruction (DPLSY), the number ranges from 1 to 2,147,483,647. When S2 is 0, infinite pulses are sent continuously.

When the instruction flow status is OFF, the pulse output stops immediately. When the instruction flow status switches from OFF to ON, the pulse output restarts. When the instruction is executed, the M8029 flag is set to ON.

The following figure shows a pulse output diagram.



The following figure shows an infinite pulse output diagram.



◆ Note:

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

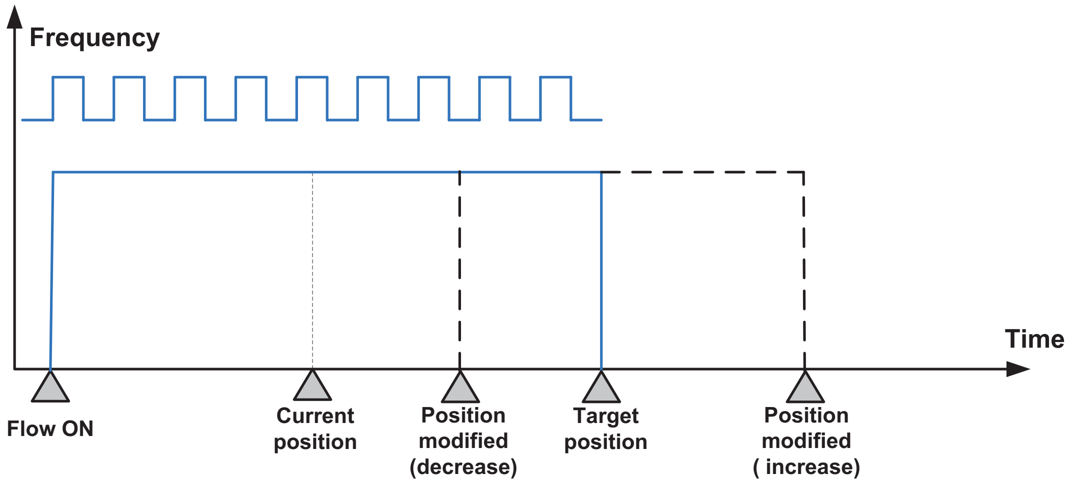
2) Acceleration/deceleration is not supported.

3) The number of output pulses can be increased or decreased during instruction execution.

Before modifying the number of output pulses, you need to set the "pulse modification valid flag bit" (M8350, M8370, M8390, M8410, or M8430; default value: OFF) of special elements. See the following table.

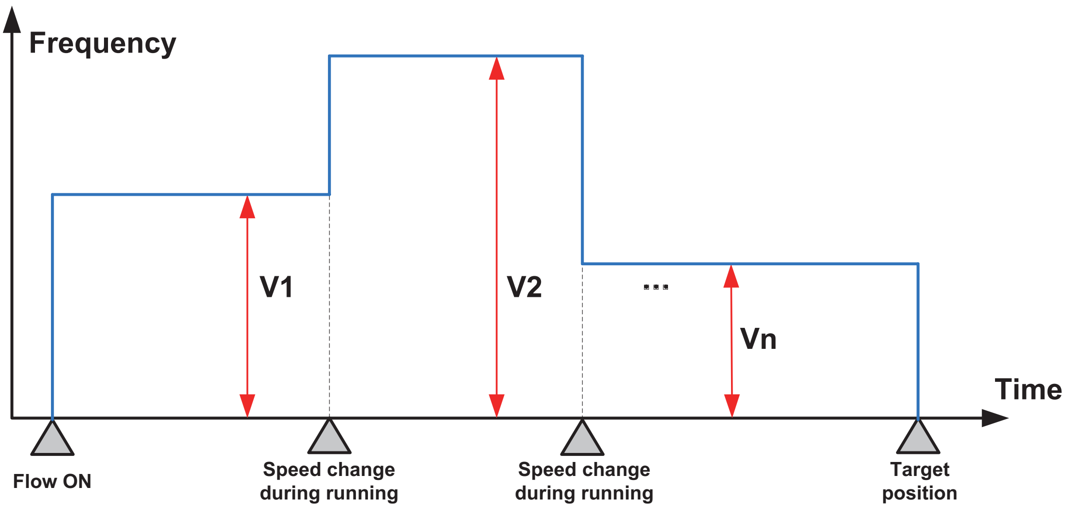
Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

Note that the modified position must be larger than the current pulse position. See the following figure.



4) The output pulse frequency can be increased or decreased during instruction execution.

The special flag does not need to be set. See the following figure.

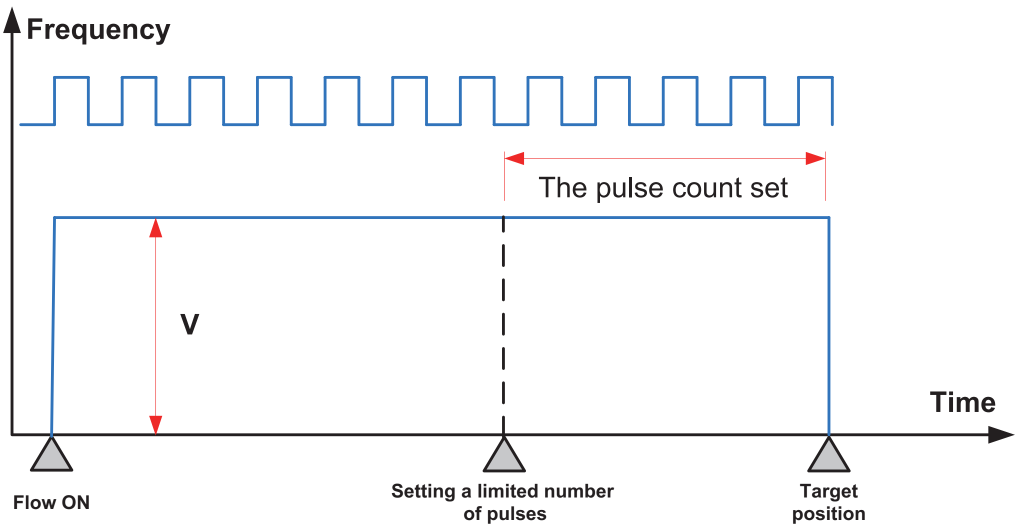


5) The number of output pulses and the output pulse frequency can be modified simultaneously during instruction execution.

6) When the number of output pulses is set to 0, the PLSY instruction is in speed mode, and infinite pulses are sent continuously.

7) When the number of output pulses is set to 0, the PLSY instruction is in speed mode. You can set the number of output pulses to a non-zero value to switch to position mode. After change, a designated number of pulses will be sent. However, the position mode cannot be changed to the speed mode.

6

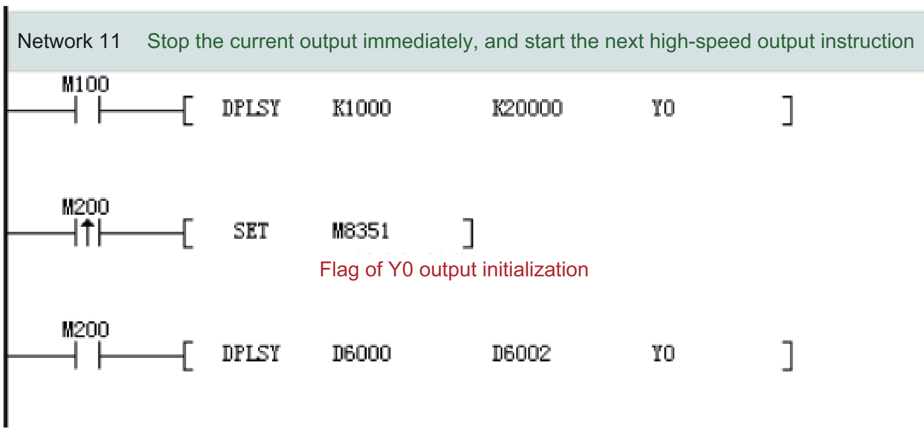


8) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

9) Pulse output complete interrupts

The "interrupt enabling flag bit" of special elements must be set. See the following table.

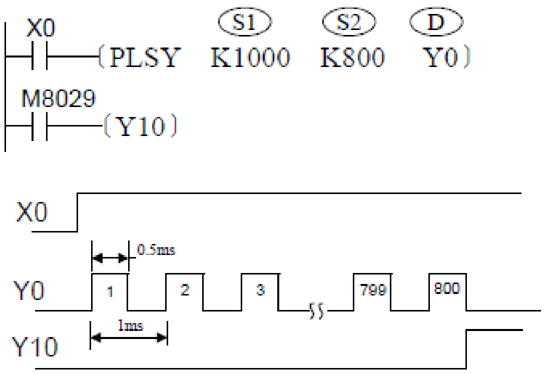
Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

10) Stop the pulse output.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

◆ Program example



When the PLSY instruction is executed, Y starts to output pulses. If the parameter of S2 is modified during execution, the number of currently output pulses is not affected, and the modification takes effect when the instruction is enabled next time. Refer to Notice above when you want to modify the number of pulses to be output during execution.

During pulse output based on the PLSY instruction, if the instruction flow X0 is changed to OFF, the pulse output stops. If X0 is changed to ON, the PLSY instruction starts to output pulses based on the current parameter.

PLSV: Variable-speed pulse output

◆ Overview

Pulses are output through the specified output port at the specified frequency and direction. Acceleration/deceleration is not supported. If the flow is inactive, the pulse output stops immediately.

PLSV S1 D1 D2			Variable Pulse Output	Applicable model: H3U	
S1	Output frequency	Specified pulse output frequency		16-bit instruction (7 steps) PLSV: Continuous execution	32-bit instruction (13 steps) DPLSV: Continuous execution
D1	Output port	High-speed pulse output port			
D2	Output direction	Pulse running direction port or bit variable			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Functions and actions

This instruction is used to output pulses at the specified frequency and running direction through the specified port. Acceleration/deceleration is not supported. When the drive flow is inactive, the pulse output stops immediately. This instruction is applicable only to the PLC of the transistor output type.

S indicates the specified pulse output frequency. For a 16-bit instruction, the range is 1 to 32,767 Hz and -1 to -32,768 Hz. For a 32-bit instruction, the range is 1 to 200,000 Hz and -1 to -200,000 Hz. The "-" indicates an instruction signal running in the reverse direction.

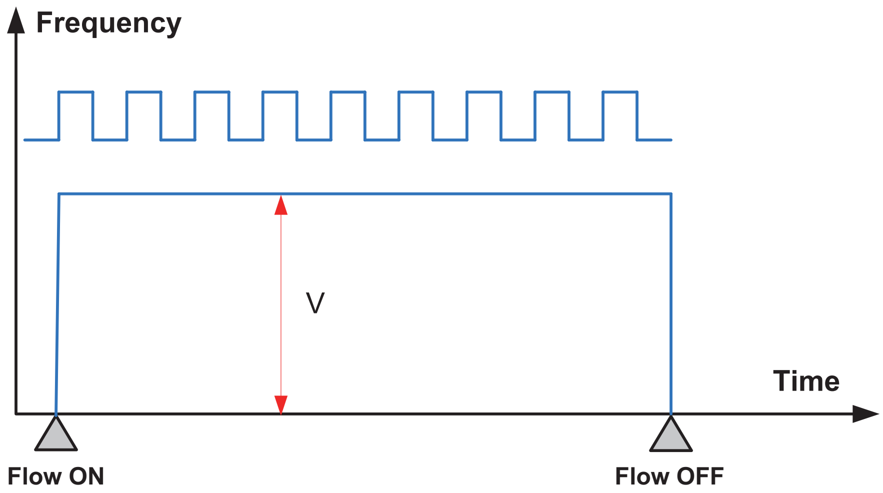
D1 indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the running direction of the output port or the bit variable. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

When the instruction flow is OFF, the output stops immediately. When the flow switches from OFF to ON, the pulse output resumes.

The following figure shows a pulse output diagram.

6



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

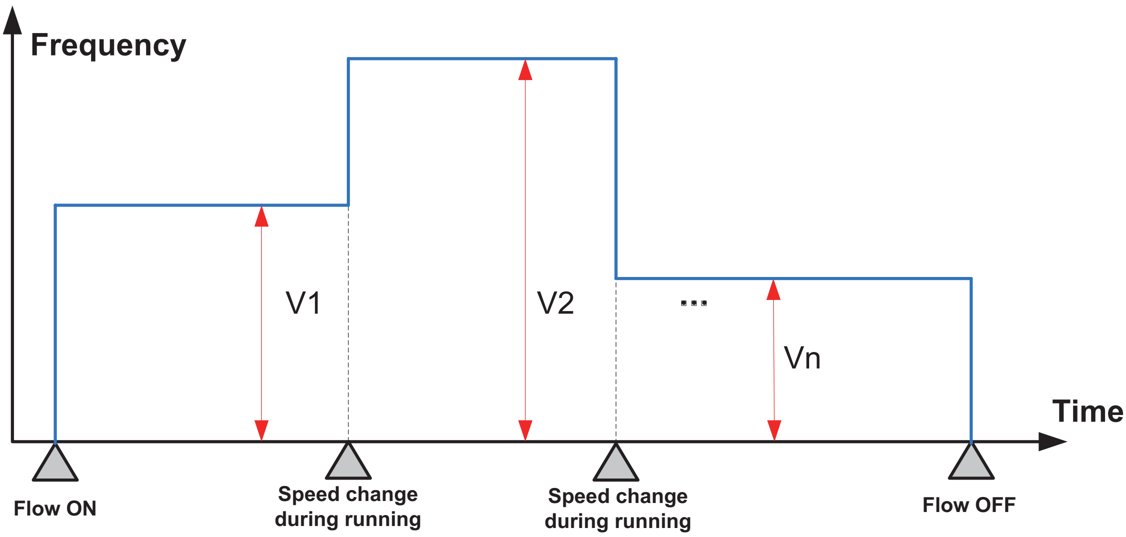
The "pulse output stop flag bit" of special elements can be monitored, and the pulse output status can be viewed. The flag bit is set during pulse output and is automatically reset when output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Acceleration/deceleration is not supported.

3) The output pulse frequency can be increased or decreased during instruction execution.

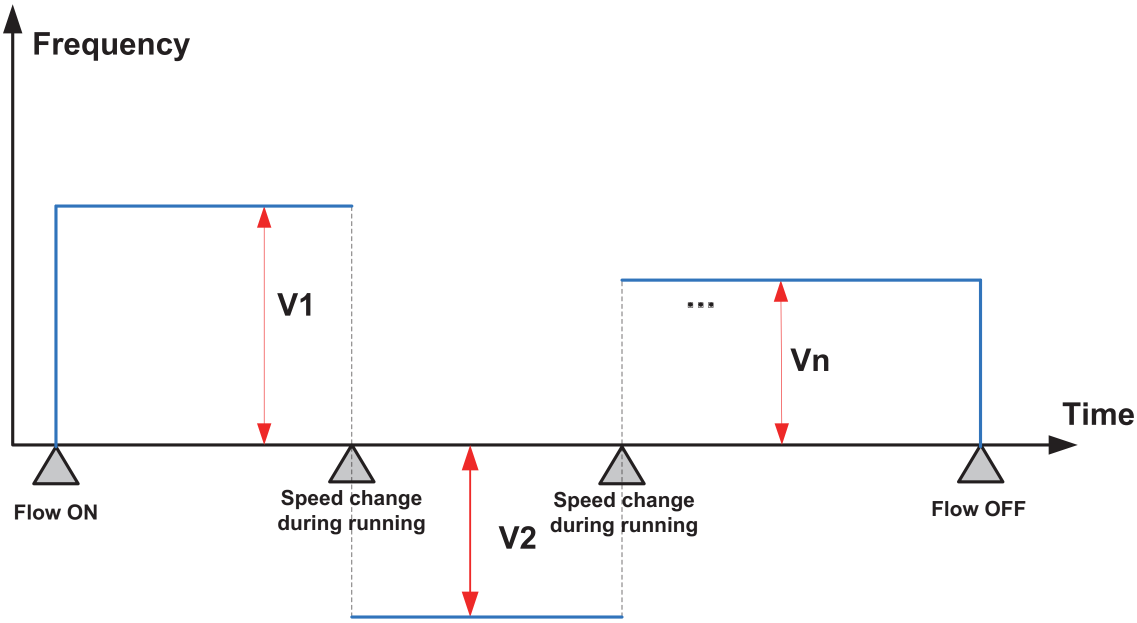
The special flag does not need to be set. See the following figure.



4) The pulse output direction can be modified during instruction execution.

The pulse output direction can be modified by modifying the set output frequency during pulse output.

When the output frequency is a positive value, the output direction is forward; otherwise, the output direction is reverse.

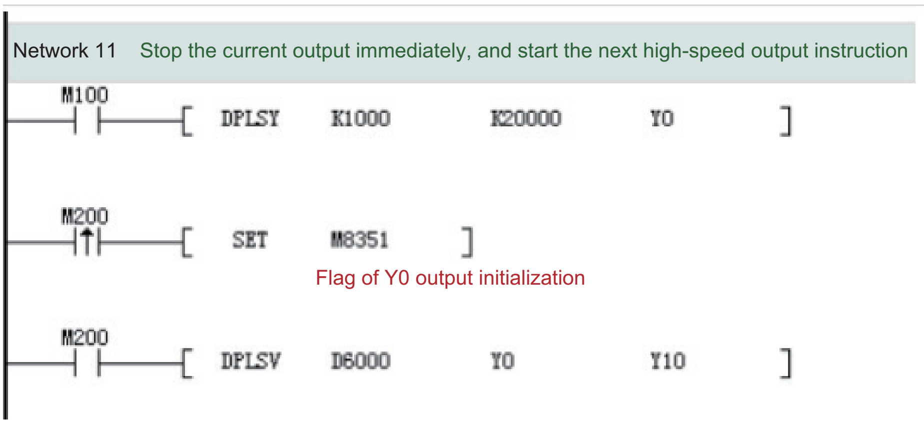


5) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

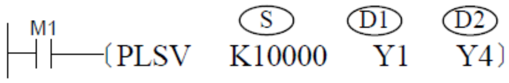
6) The pulse output complete interrupt is not supported in speed control mode.

7) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

◆ Program example



It indicates that when M1 is ON, pulses are output at a frequency of 10 kHz through the port Y1. Y4 is used to control the running direction. If Y4 is ON, the output direction is forward.

PLSV2: Variable-speed pulse output with acceleration/deceleration

◆ Overview

Pulses are output through the specified output port at the specified frequency and direction. Acceleration/ deceleration is supported. When the flow is inactive, the pulse output is decelerated to stop.

PLSV2 S1 D1 D2		Variable-speed pulse output with acceleration/ deceleration	Applicable model: H3U	
S1	Output frequency	Specified pulse output frequency	16-bit instruction (7 steps) PLSV2: Continuous execution	32-bit instruction (13 steps) DPLSV2: Continuous execution
D1	Output port	High-speed pulse output port		
D2	Output direction	Pulse running direction port or bit variable		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Functions and actions

This instruction is used to output pulses at the specified frequency and running direction through the specified port. Acceleration/deceleration is supported. When the drive flow is inactive, the pulse output is decelerated to stop. This instruction is applicable only to the PLC of the transistor output type.

S1 indicates the specified pulse output frequency. For a 16-bit instruction, the range is 50 to 32,767 Hz and -50 to -32,768 Hz. For a 32-bit instruction, the range is 50 to 200,000 Hz and -1 to -200,000 Hz. The "-" indicates an instruction signal running in the reverse direction.

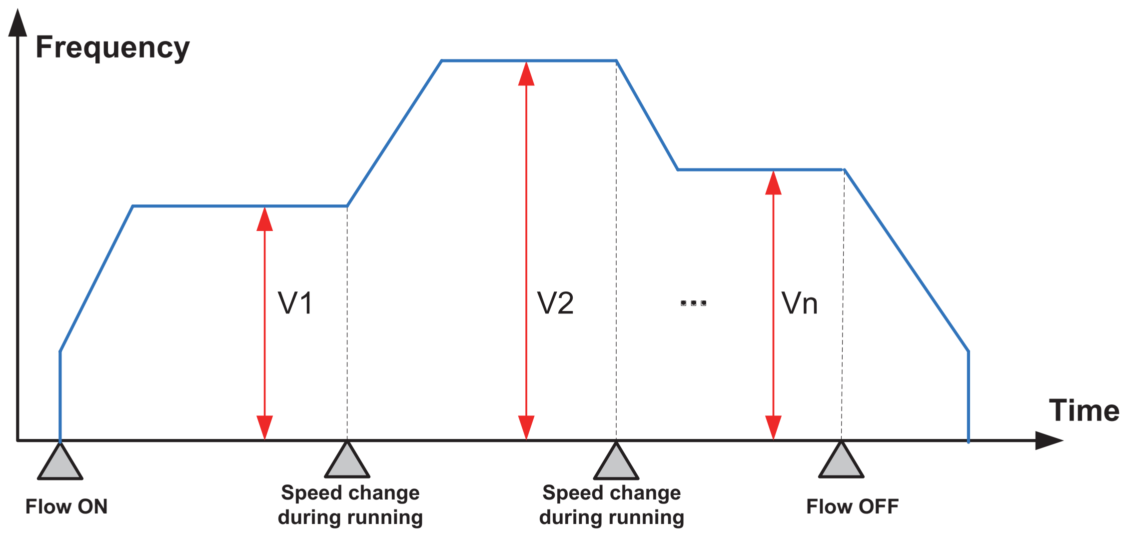
D1 indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the running direction of the output port or the bit variable. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

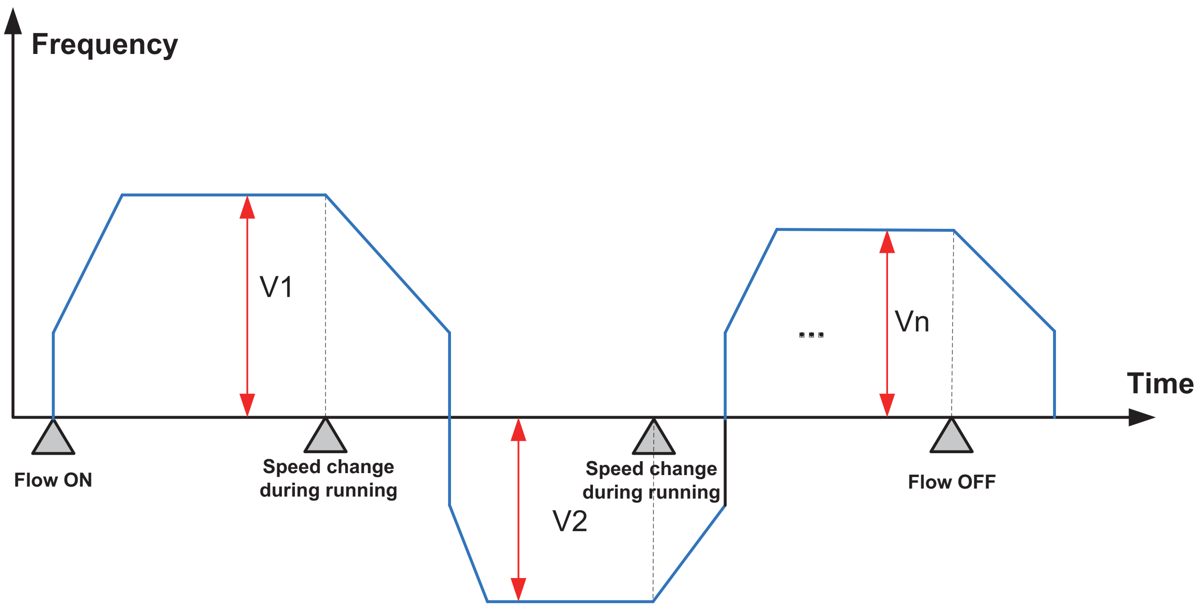
When the instruction flow is OFF, the pulse output is decelerated to stop, and the execution complete flag M8029 remains unchanged. After the instruction driving point switches to OFF and the flag is ON during pulse output, the flag is no longer driven by the instruction. When the flow switches from OFF to ON, the pulse output resumes.

The following figure shows a pulse output diagram.

After the instruction is enabled, the pulse output frequency is accelerated from the base frequency to the set frequency. When the pulse output frequency is changed, the output frequency is automatically accelerated or decelerated to the changed output frequency. When the flow is inactive, the pulse output is decelerated to stop.



The pulse output direction can be modified by modifying the set output frequency during pulse output. When the output frequency is a positive value, the output direction is forward; otherwise, the output direction is reverse. The pulse output acceleration/deceleration time can be set separately, as shown in the following.



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Only trapezoid acceleration/deceleration is supported.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high- speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430; default value: OFF) of special elements is used for setting and differentiation, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

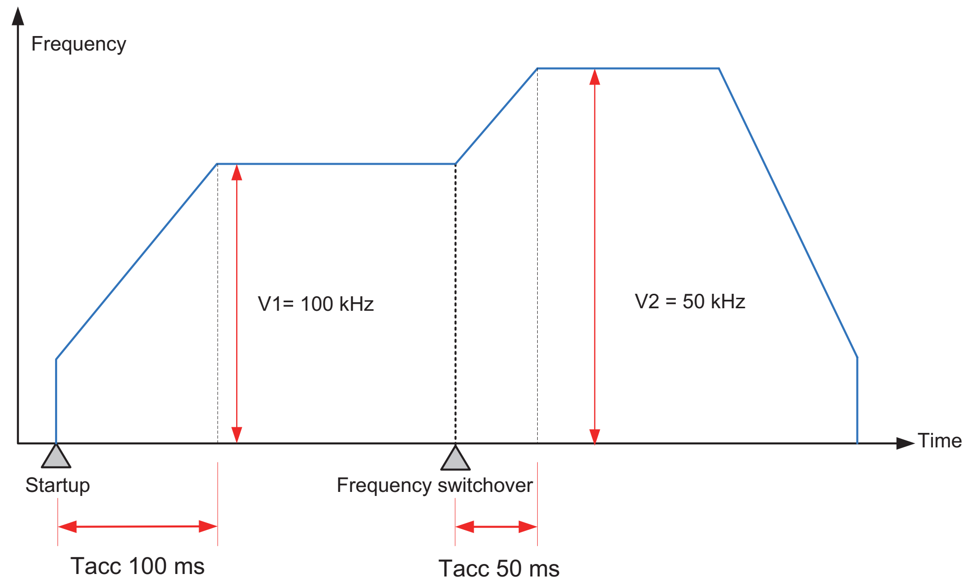
When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) During execution of a multi-segment pulse output instruction, the acceleration/deceleration computation is subject to the first speed. The acceleration/deceleration slope (acceleration/ deceleration speed) remains unchanged during frequency switch.

For example, if output frequency is 100 kHz and the acceleration time is 100 ms at the first speed while the output frequency at the second speed is 150 kHz, it takes about 50 ms to accelerate from the first speed to the second speed. It works similarly in deceleration mode. See the following figure.

6



5) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

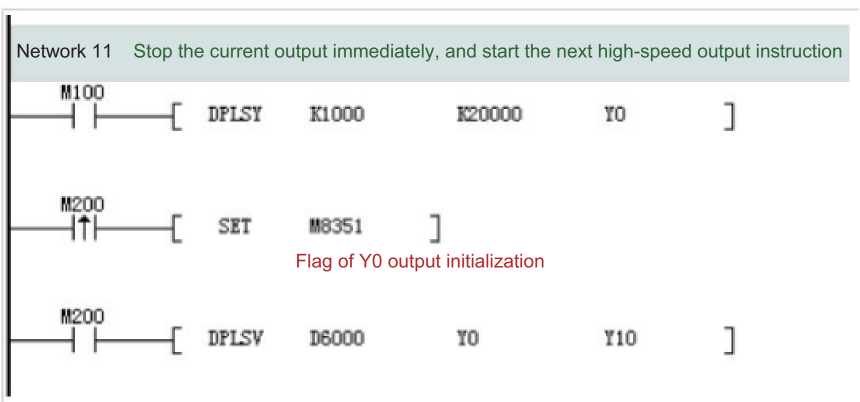
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) The pulse output complete interrupt is not supported in speed control mode.

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

ZRN: Regression through the origin

◆ Overview

After the system is started, it accelerates to the set regression output frequency to drive the actuator to move toward the origin (DOG). After a DOG signal is detected, the system decelerates to the creep speed. When the DOG signal is OFF, pulse output is stopped.

ZRN S1 S2 S3 D		ZRN	Applicable model: H3U	
S1	Regression frequency	ZRN output frequency setting	16-bit instruction (9 steps) ZRN: Continuous execution	32-bit instruction (17 steps) DZRN: Continuous execution
S2	Creep frequency	Creep output frequency setting when the DOG signal is detected		
S3	DOG signal	Specified origin input signal (DOG)		
D	Output port	High-speed pulse output port		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Functions and actions

When the PLC works in combination with the servo drive, this instruction is used to drive the actuator to move toward the action origin (DOG) at the specified pulse speed and pulse output port until the original signal meets relevant conditions.

S1 indicates the speed of the ZRN action. For a 16-bit instruction, the range is 10 to 32,767 Hz. For a 32-bit instruction, the range is 10 to 200,000 Hz.

S2 indicates the creep speed after the original signal switches to ON. The range is 10 to 32,767 Hz.

S3 indicates the original signal (DOG) input. Although any of the X, Y, M, and S signals can be used, the X signal boasts the best timeliness.

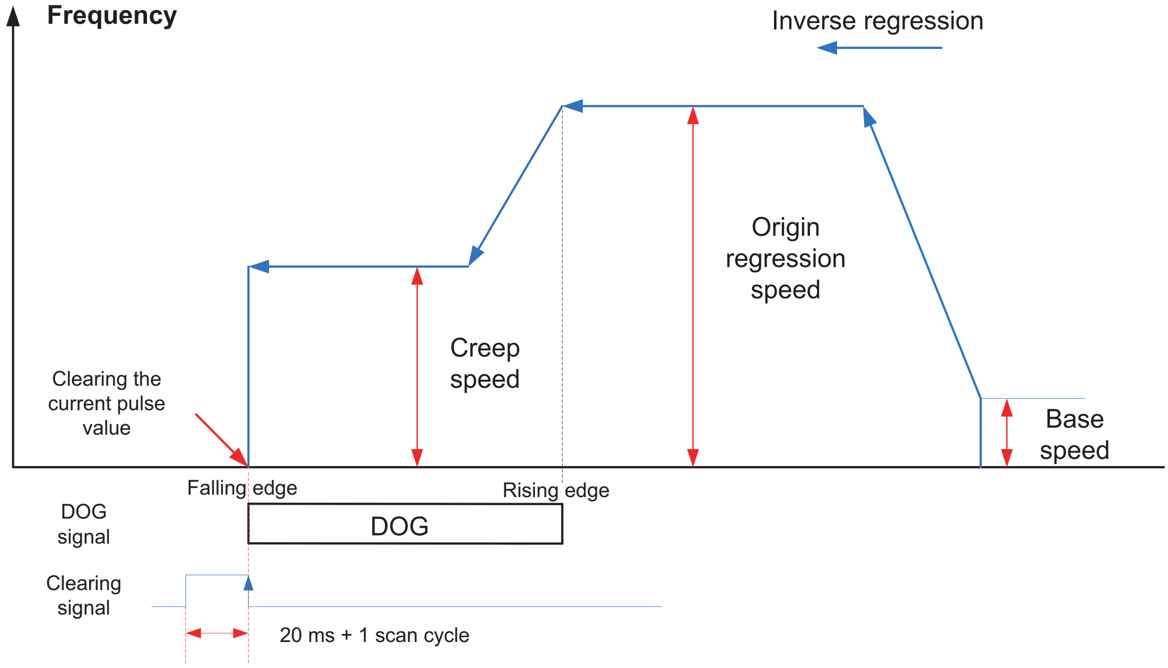
D indicates the head address for pulse output, which can be Y0, Y1, Y2, Y3, or Y4.

When the system is powered on or starts to run, ZRN is executed generally to write the original position data of the mechanical action in advance. If the position information can be retained upon power failure, the instruction does not need to be executed each time the system is powered on. When the instruction is executed, the actuator can only move in a single direction (in the negative direction); therefore, the origin return action must start at the front-end of the DOG signal.

When the instruction flow status is OFF, the pulse output stops immediately. When the instruction flow status switches from OFF to ON, the pulse output restarts. When the instruction is executed, the M8029 flag is set to ON.

The following figure shows a pulse output diagram.

6



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Only trapezoid acceleration/deceleration is supported.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430; default value: OFF) of special elements is used for setting and differentiation, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

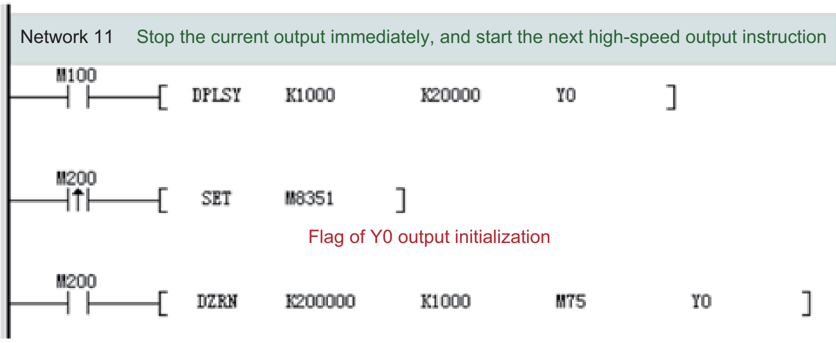
$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

5) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

6) The pulse output complete interrupt is not supported in speed control mode.

7) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

8) Clearing signal output

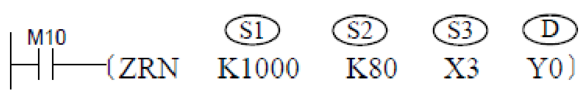
The "clearing signal output active flag bit" of special elements can be set to output a clearing signal. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8341	M8361	M8381	M8401	M8421	Valid output label for the DSZR/ ZRN and other clearing signals

The clearing signal can be specified by special registers and can be output through Y ports. For example, if D8350 = 5, the port that outputs the pulse clearing signal is Y5. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8350	D8370	D8390	D8410	D8430	Number of clearing elements (in decimal)
5 (the default port is Y5)	6 (the default port is Y6)	7 (the default port is Y7)	8 (the default port is Y10)	9 (the default port is Y11)	

◆ Program example



The action of this instruction is as follows: when M10 switches to ON, the PLC outputs pulses at 1000 Hz through the Y0 high-speed output port to drive the stepping motor to return to the origin; when the DOG signal switches to ON (the DOG slider touches the DOG contact), the output frequency decreases to 80 Hz, and the PLC creeps at a low speed. When the DOG signal switches to OFF again, the PLC stops outputting pulses through Y0, and writes 0 into the current value register. Besides, when the clearing signal output function is ON, the clearing signal is output at the same time. Next, the execution complete flag (M8029) is set to ON, and the pulse output monitoring switches to OFF.

DSZR: DOG search return to origin

◆ Overview

After the system is started, it accelerates to the regression output frequency set by the special register to drive the actuator to move toward the origin (DOG) according to the set action sequence. After a DOG signal is detected, the system decelerates to the creep speed. When the DOG signal is OFF, pulse output is stopped.

DSZR S1 S2 D1 D2			DOG search return to origin	Applicable model: H3U	
S1	DOG signal	Specified origin input signal (DOG)		16-bit instruction (9 steps) DSZR: Continuous execution	32-bit instruction (17 steps) DDSZR: Continuous execution
S2	Origin signal	Specified origin input signal			
D1	Output port	High-speed pulse output port			
D2	Output direction	Pulse running direction port or bit variable			

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Functions and actions

When the PLC works in combination with the servo drive, this instruction is used to output pulses at the pulse speed specified by the special register through the pulse output port, so that the actuator moves to the action origin according to the preset action sequence; when the near point signal (DOG) changes from ON to OFF during running, and the clearing signal changes from OFF to ON, the PLC immediately stops outputting the pulse.

The rotational direction signal is output during regression, and the clearing signal is output after regression.

In a system with forward/reverse rotation limit, the DSZR with DOG search mode can be enabled. In a system without forward/reverse rotation limit or not using the forward/reverse rotation limit for original regression, the zero return can be performed by specifying the zero return direction.

S1 indicates the input of the near point signal (DOG). Although any of the X, Y, M, and S signals can be used, the X signal boasts the best timeliness.

S2 indicates the input of the origin signal. To represent the accurate position of the action origin, you can only specify the X signal.

D1 indicates the pulse output port, which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the rotational direction output port. ON indicates rotating in the forward direction (the current value increases with pulse output); OFF indicates rotating in the reverse direction (the current value decreases with pulse output).

1) The special M elements are defined in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status
M8341	M8361	M8381	M8401	M8421	Active output label for the DSZR/ZRN and other clearing signals[1]
M8342	M8362	M8382	M8402	M8422	Designation of DSZR instruction zero return direction[1]
M8343	M8363	M8383	M8403	M8423	Forward rotation limit
M8344	M8364	M8384	M8404	M8424	Reverse rotation limit
M8345	M8365	M8385	M8405	M8425	Near point signal logical inversion (*1)
M8346	M8366	M8386	M8406	M8426	Origin signal logical inversion (*1)
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag (*1)
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/deceleration time and modification to the pulse
M8351	M8371	M8391	M8411	M8431	Port output initialization flag
M8354	M8374	M8394	M8414	M8434	DSZR execution abnormal end flag bit

6

2) The special D elements are defined in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8344	D8364	D8384	D8404	D8424	DSZR instruction zero return speed (Hz) (32-bit) [default value: 50,000]
D8345	D8365	D8385	D8405	D8425	
D8346	D8366	D8386	D8406	D8426	Creep speed (Hz) [default value: 2000]
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]
D8350	D8370	D8390	D8410	D8430	Element number clearing

NOTE

- [1]: When the status changes from RUN to STOP, the signal is cleared.
- For details about the maximum speed, zero return speed, creep speed, and base speed, see "A.2 Special Soft Element Register Range". The following rules must be followed:

  Base speed ≤ zero return speed ≤ maximum speed

  Base speed ≤ creep speed ≤ maximum speed

- Select appropriate parameters based on the set parameter value range. It is suggested that the creep speed be smaller than or equal to the zero return speed.

  The zero return speed ranges from 10 to 200,000 Hz.

  The creep speed ranges from 10 to 32,767 Hz.

  The base speed ranges from 10 to 32,767 Hz.

You can set the zero return direction flag bit to ON or OFF to specify the zero return direction. The base speed is accelerated to the zero return speed and the actuator moves toward the direction specified by the zero return direction flag bit. When the system detects that the near point signal (DOG) specified by S1 is ON, the system decelerates to the creep speed. After the near point signal (DOG) specified by S1 switches from ON to OFF, if the system detects that the origin signal specified by S2 switches from OFF to ON, the system stops outputting the pulse immediately.

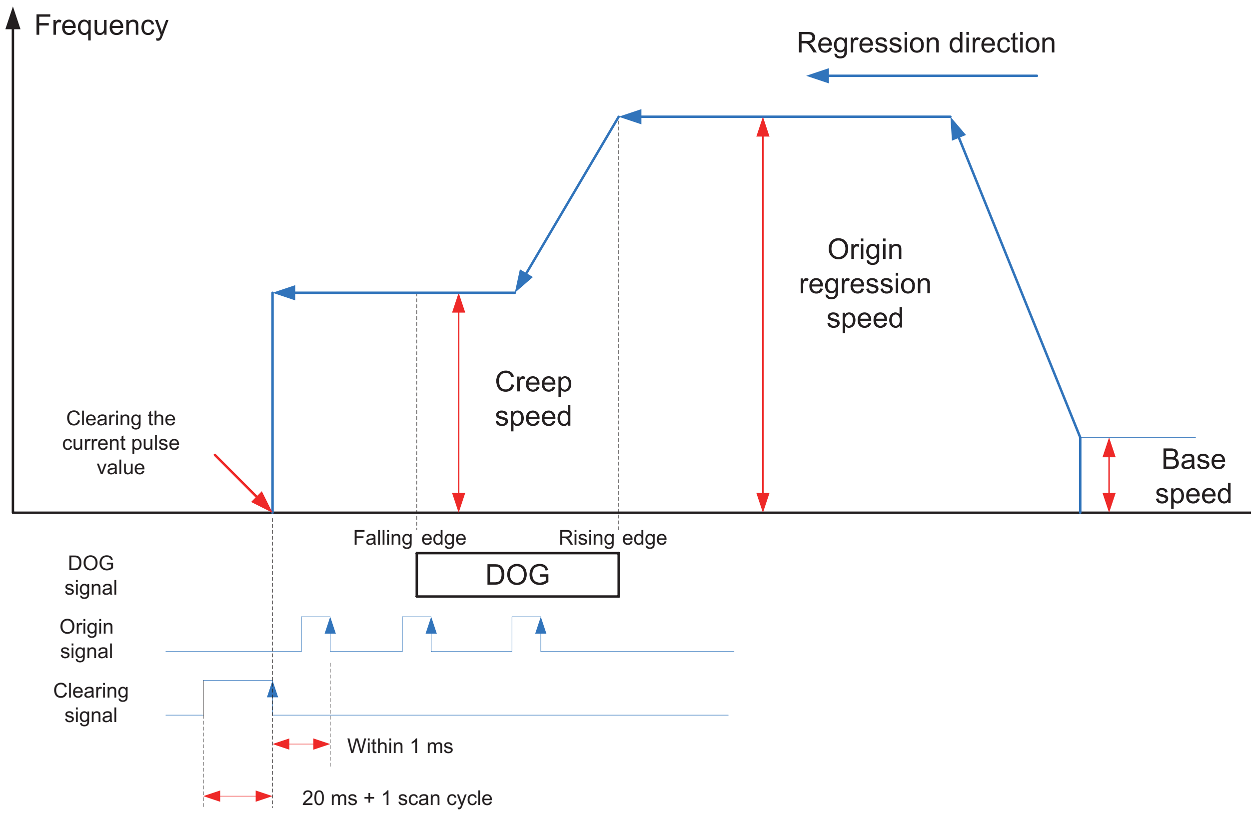
If the same input is specified for the near point signal and the origin signal, similar to the ZRN instruction, this instruction does not use the origin signal. When the near point signal switches from ON to OFF, the system immediately stops outputting the pulse. If the same input is specified for the near point signal and the origin signal and the logical inversion flag bit is set to ON, the logic of the near point signal prevails.

When the clearing signal output function is enabled (ON), after the pulse output stops (within 1 ms), the clearing signal keeps ON within 20 ms plus one calculation cycle. When the instruction execution complete flag bit (M8029) is ON, the zero return action stops.

This is the description about the case where the logical inversion flag bit of the near point signal and the origin signal is set to OFF. If the logical inversion flag bit is set to ON, the ON and OFF states of the corresponding near point signal and origin signal must be changed to each other.

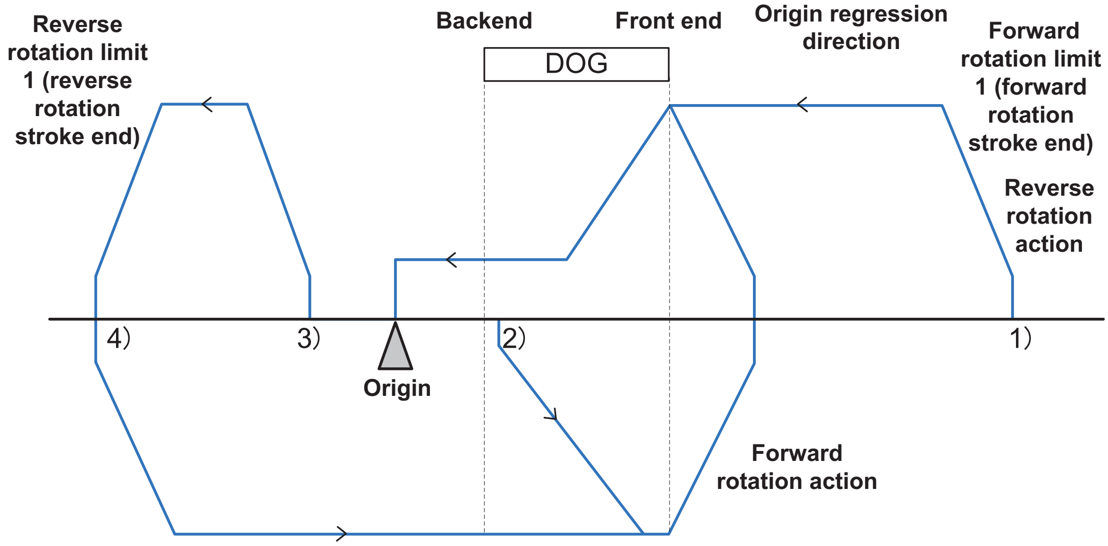
When the instruction flow status is OFF, the pulse output stops immediately. When the instruction flow status switches from OFF to ON, the pulse output restarts. When the instruction is executed, the M8029 flag is set to ON.

The following figure shows a pulse output diagram.



3) DOG search

When DSZR with DOG search is executed when forward and reverse rotation limits are designed. At this time, the zero return action is subject to the corresponding start position of zero return.



a) When the start position is before DOG (including the case when the forward rotation limit 1 is set to ON):

① The system executes the zero return instruction to start the zero return action.

② The system moves to the zero return direction at the zero return speed.

③ The system decelerates to the creep speed upon detecting the DOG frontend.

④ After detecting the DOG backend, the system stops upon detecting the first origin signal.

b) When the start position is within the DOG:

① The system executes the zero return instruction to start the zero return action.

② The system moves to the direction opposite to the zero return direction at the zero return speed.

③ The system decelerates to stop upon detecting the DOG frontend. (Leaving DOG)

④ The system moves to the zero return direction at the zero return speed. (Entering DOG again)

⑤ The system decelerates to the creep speed upon detecting the DOG frontend.

⑥ When detecting the DOG backend, the system stops upon detecting the first origin signal.

c) When the start position is after the DOG (the near point signal is set to OFF):

① The system executes the zero return instruction to start the zero return action.

② The system moves to the zero return direction at the zero return speed.

③ The system decelerates to stop upon detecting the reverse rotation limit 1 (reverse rotation limit).

④ The system moves to the direction opposite to the zero return direction at the zero return speed.

⑤ The system decelerates to stop upon detecting the DOG frontend. (Detecting [leaving] DOG)

⑥ The system moves to the zero return direction at the zero return speed. (Entering DOG again)

⑦ The system decelerates to the creep speed upon detecting the DOG frontend.

⑧ When detecting the DOG backend, the system stops upon detecting the first origin signal.

D) When the limit switch (reverse rotation limit 1) of the zero return direction is set to ON:

① The system executes the zero return instruction to start the zero return action.

② The system moves to the direction opposite to the zero return direction at the zero return speed.

③ The system decelerates to stop upon detecting the DOG frontend. (Detecting [Leaving] DOG)

④ The system moves to the zero return direction at the zero return speed. (Entering DOG again)

⑤ The system decelerates to the creep speed upon detecting the DOG frontend.

⑥ When detecting the DOG backend, the system stops upon detecting the first origin signal.

e) Note: When designing the near point signal (DOG), you need to design a sufficient ON time for the system to decelerate to the creep speed. The creep speed must be as low as possible. If the system stops immediately without deceleration, a great creep speed may result in position offset.

◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Only trapezoid acceleration/deceleration is supported.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

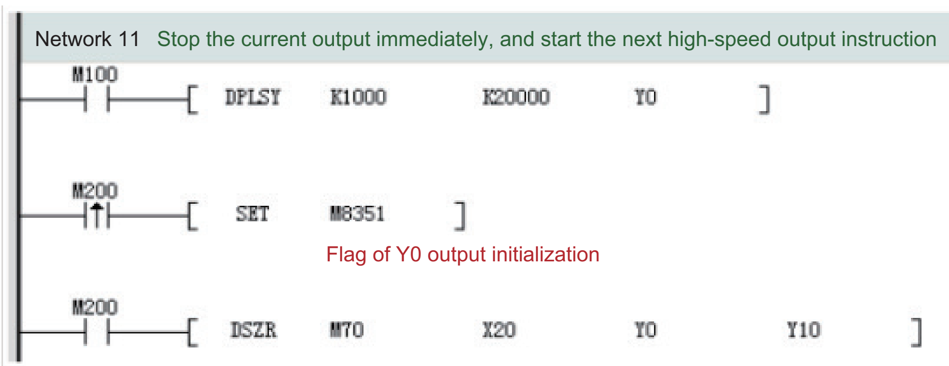
$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

5) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

6) The pulse output complete interrupt is not supported in speed control mode.

7) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

8) Clearing signal is output.

You can set the "clearing signal output valid flag bit" of special elements to output a clearing signal after the origin is regressed. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8341	M8361	M8381	M8401	M8421	Valid output label for the DSZR/ZRN and other clearing signals

The clearing signal is output only through the Y port specified by the clearing elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8350	D8370	D8390	D8410	D8430	Element number clearing
5 (the default port is Y5)	6 (the default port is Y6)	7 (the default port is Y7)	8 (the default port is Y10)	9 (the default port is Y11)	

9) Signal logical inversion is shown in the following table.

OFF: Positive logic (when the input is ON, the near point/clearing signal is ON).

ON: Negative logic (when the input is OFF, the near point/clearing signal is ON).

Y0	Y1	Y2	Y3	Y4	Attribute
M8345	M8365	M8385	M8405	M8425	Near point signal logical inversion (*1)
M8346	M8366	M8386	M8406	M8426	Origin signal logical inversion (*1)

◆ Program example

PLSR: Pulse output with acceleration/deceleration

◆ Overview

A specified number of pulses are output at the specified pulse frequency and the set acceleration/deceleration time.

PLSR S1 S2 S3 D		Pulse output with acceleration/deceleration	Applicable model: H3U	
S1	Output frequency	Specified pulse output frequency	16-bit instruction (9 steps) PLSR: Continuous execution	32-bit instruction (17 steps) DPLSR: Continuous execution
S2	Pulse count	Specified number of pulse outputs		
S3	Acceleration/Deceleration time	Acceleration/Deceleration time setting		
D	Output port	High-speed pulse output port		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

◆ Functions and actions

Because the relay is not applicable to high-frequency actions, this instruction is applicable only to the PLC of the transistor output type. This instruction is used to output pulses of a fixed number at the acceleration/deceleration time. Wherein:

S1 indicates the set output pulse frequency. For a 16-bit instruction, the frequency ranges from 10 Hz to 32,767 Hz. For a 32-bit instruction, the frequency ranges from 10 Hz to 200,000 Hz.

S2 indicates the number of pulses to be output. For a 16-bit instruction, the number ranges from 1 to 32,767. For a 32-bit instruction, the number ranges from 1 to 2,147,483,647.

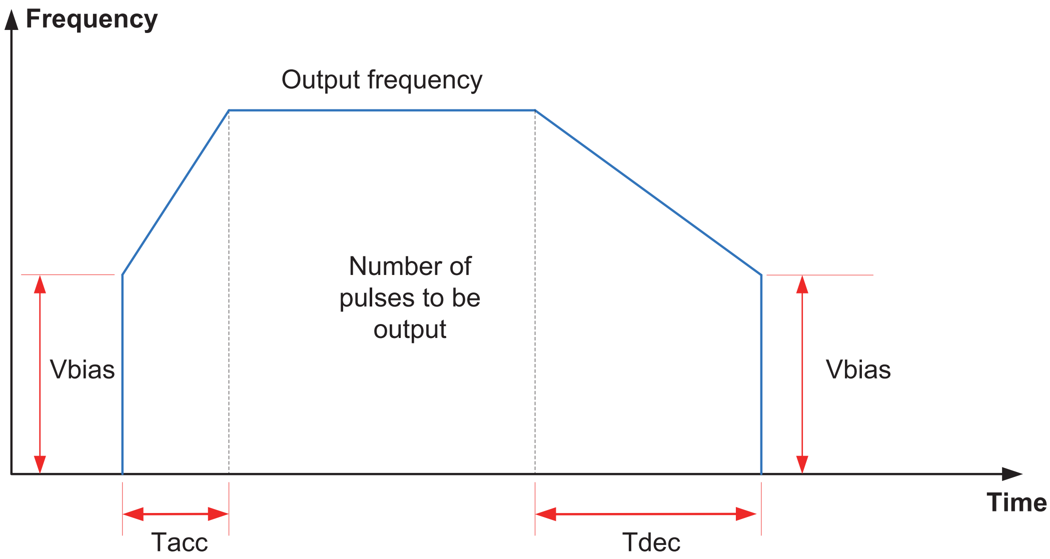
S3 indicates the acceleration/deceleration time ranging from 10 to 5000 ms. Note that the deceleration time is the same as the acceleration time by default. (In the H2U series PLCs, the deceleration time can be set separately. See the following description.)

D indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

Change to the operand during pulse output does not affect the current output operation, but the modified content takes effect when the instruction is executed next time.

When the instruction flow is OFF, the pulse output is decelerated to stop, and the execution complete flag M8029 remains unchanged. After the instruction driving point switches to OFF, when the flag is ON during pulse output, the flag is no longer driven by the instruction. When the flow switches from OFF to ON, the pulse output resumes. When the instruction execution is complete, the M8029 flag is set to ON.

The following figure shows a pulse output diagram.



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The "pulse output stop flag bit" of special elements can be monitored, and the pulse output status can be viewed. The flag bit is set during pulse output and is automatically reset when output is finished. See the following table.

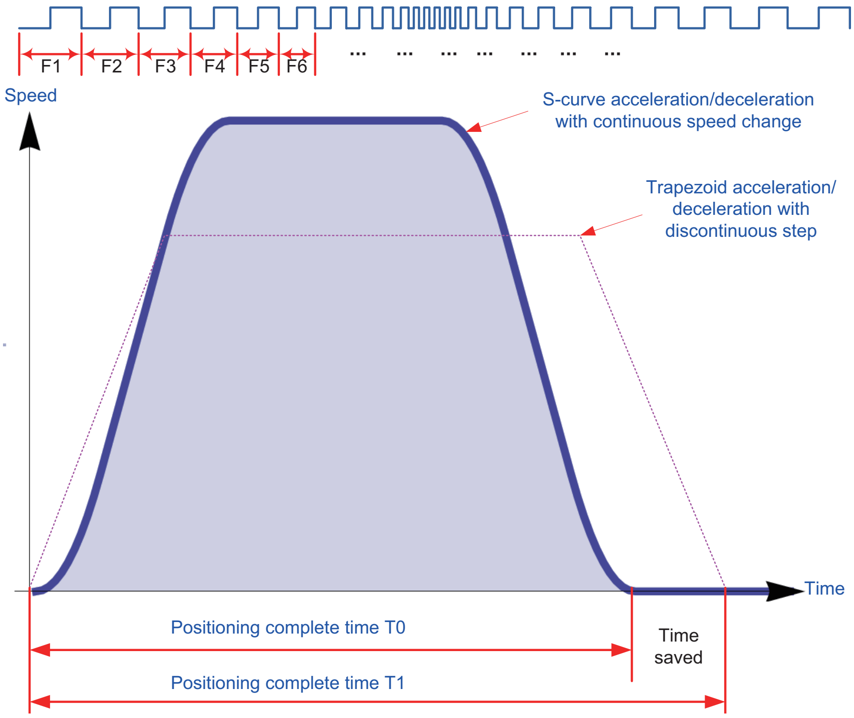
Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Trapezoid acceleration/deceleration and S-curve acceleration/deceleration are supported.

The two acceleration/deceleration modes can be distinguished by setting the "S-curve acceleration/ deceleration enabling flag bit" of special elements. If the flag bit is not set, the trapezoid acceleration/ deceleration mode is used by default. The following table lists details about S-curve acceleration/ deceleration.

Y0	Y1	Y2	Y3	Y4	Attribute
M8347	M8367	M8387	M8407	M8427	S-curve acceleration/deceleration enabling

DRVI, DRVA, and PLSR support S-curve acceleration/deceleration. Therefore, at given mechanical stability, the target speed is increased, the positioning time is shortened, and the processing efficiency is improved.



The advanced pulse-by-pulse modulation algorithm is used for S-curve acceleration/deceleration. Frequency of each pulse is adjusted to ensure more smooth positioning.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms. The time of trapezoid acceleration/deceleration and S-curve acceleration/deceleration can be set separately.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
Setting of the instruction parameters					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
Setting of the instruction parameters					Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

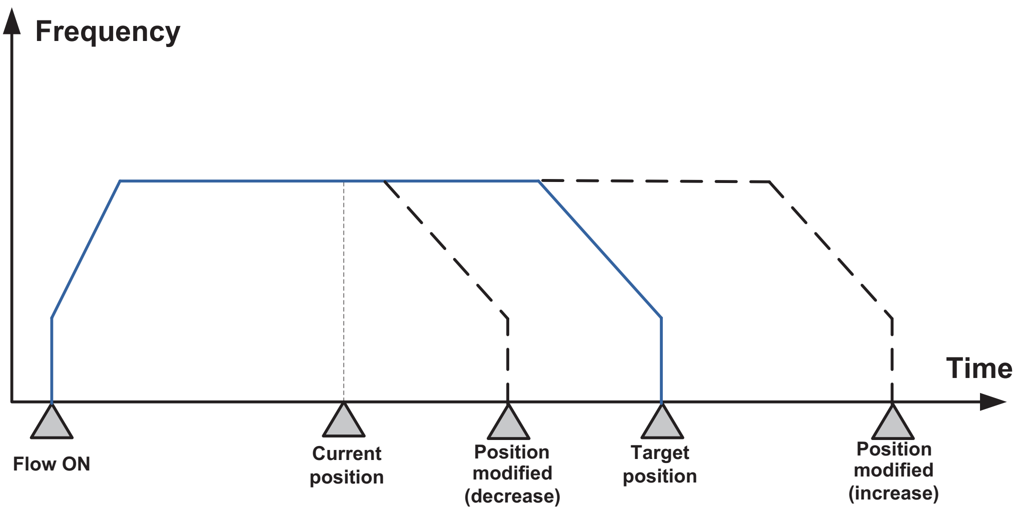
5) The number of output pulses can be increased or decreased during instruction execution.

Before modifying the number of output pulses, you need to set the "pulse modification valid flag bit" (M8350, M8370, M8390, M8410, or M8430; default: OFF) of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

6

Note that the modified position must be larger than the current pulse position. See the following figure.

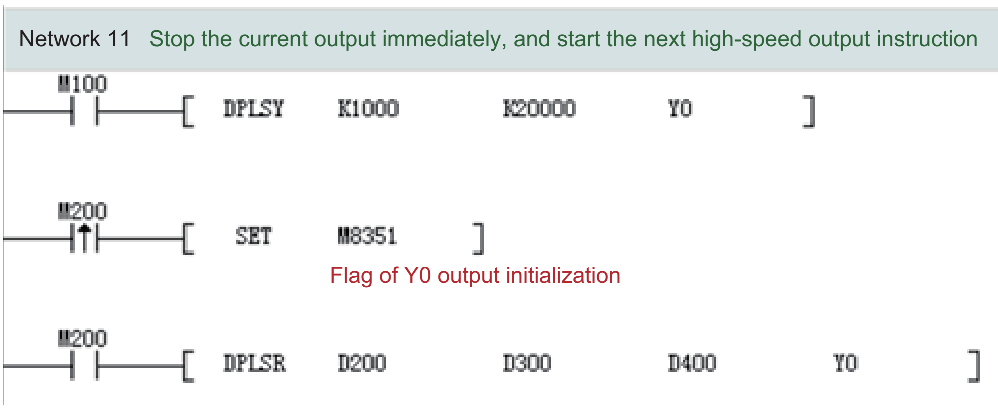


6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Pulse output complete interrupts.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

DRVA: Absolute positioning

◆ Overview

A specified number of pulses are output at the specified pulse frequency in the specified direction through the specified output port. The action is based on the absolute position.

DRVA S1 S2 D1 D2		Absolute positioning	Applicable model: H3U	
S1	Pulse count	Specified number of pulse outputs	16-bit instruction (9 steps) DRVA: Continuous execution	32-bit instruction (17 steps) DDRVA: Continuous execution
S2	Output frequency	Specified pulse output frequency		
D1	Output port	High-speed pulse output port		
D2	Output direction	Pulse running direction port or bit variable		

◆ Operands

Operand	Bit Element							Word Element																	
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output pulses at the specified frequency and in the specified direction through the specified port, and drive the servo actuator to move to the specified destination. This instruction is applicable only to the PLC of the transistor output type.

S1 indicates the specified destination (absolute position). For a 16-bit instruction, the range is –32768 to +32,767. For a 32-bit instruction, the range is –2,147,483,648 to +2,147,483,647. The "–" indicates the reverse direction.

The following table lists the absolute position current values.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

S2 indicates the specified output pulse frequency. For a 16-bit instruction, the range is 10 to 32,767 Hz. For a 32-bit instruction, the range is 10 to 200,000 Hz.

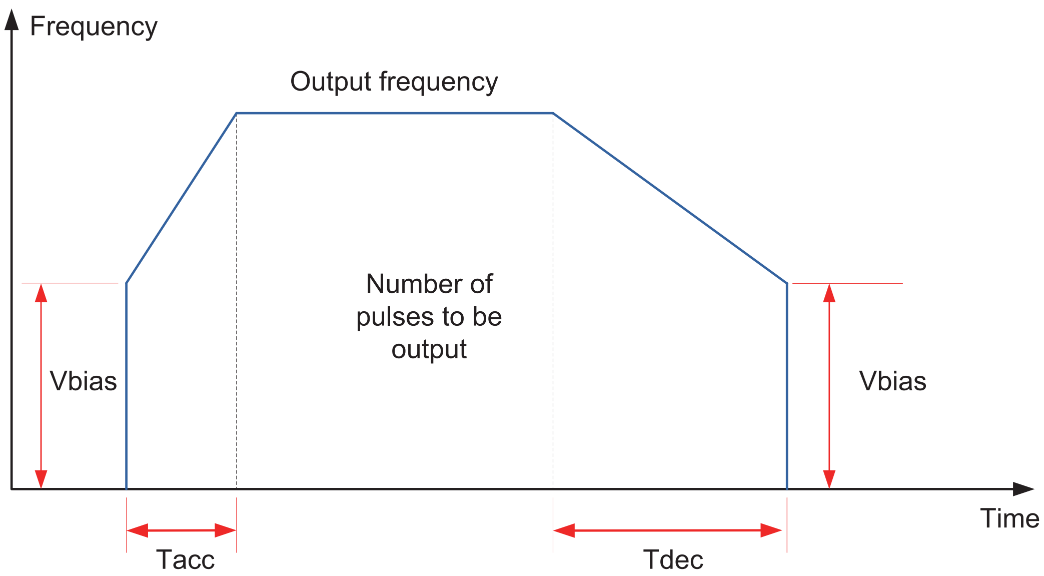
D1 indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the running direction of the output port or the bit variable, which is determined by the difference between S1 and the current position. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

Even if the operand content is modified during instruction execution, the modification takes effect when the instruction is executed next time.

When the instruction flow switches to OFF during instruction execution , the system decelerates to stop. The execution complete flag M8029 remains unchanged. After the instruction flow switches to OFF, when the flag of pulse output is ON, the instruction is no longer driven by the instruction flow.

The following figure shows a pulse output diagram.



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

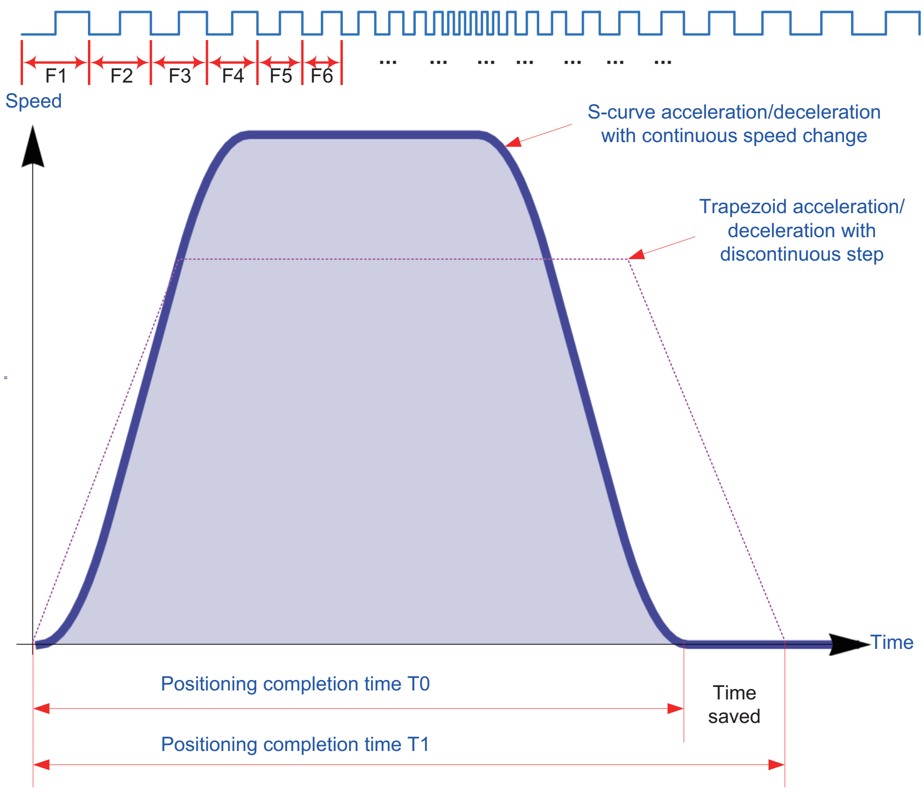
2) Trapezoid acceleration/deceleration and S-curve acceleration/deceleration are supported.

The two acceleration/deceleration modes can be distinguished by setting the "S-curve acceleration/ deceleration enabling flag bit" of special elements. If the flag bit is not set, the trapezoid acceleration/ deceleration mode is used by default. The following table lists details about S-curve acceleration/ deceleration:

6

Y0	Y1	Y2	Y3	Y4	Attribute
M8347	M8367	M8387	M8407	M8427	S-curve acceleration/deceleration enabling

DRVI, DRVA, and PLSR support S-curve acceleration/deceleration. Therefore, at given mechanical stability, the target speed is increased, the positioning time is shortened, and the processing efficiency is improved.



The advanced pulse-by-pulse modulation algorithm is used for S-curve acceleration/deceleration. Frequency of each pulse is adjusted to ensure more smooth positioning.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms. The time of trapezoid acceleration/deceleration and S-curve acceleration/deceleration can be set separately.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

6

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

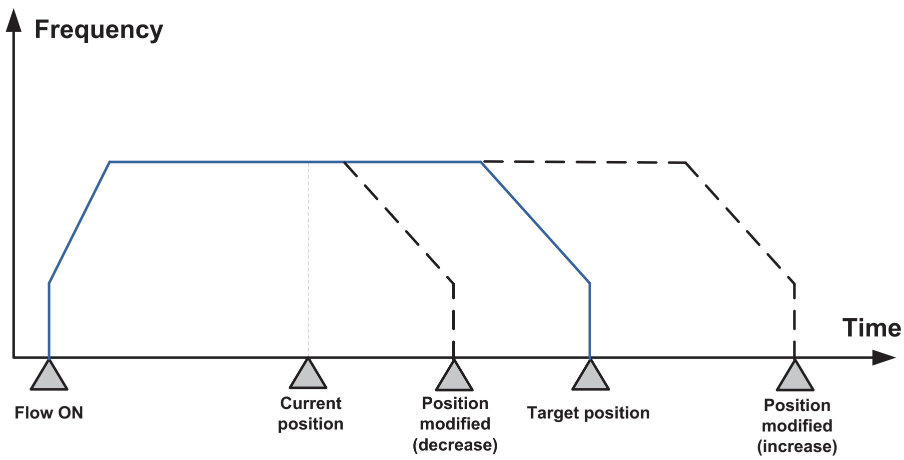
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

5) The number of output pulses can be increased or decreased during instruction execution.

Before modifying the number of output pulses, you need to set the "pulse modification valid flag bit" (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

Note that the modified position must be larger than the current pulse position. See the following figure.

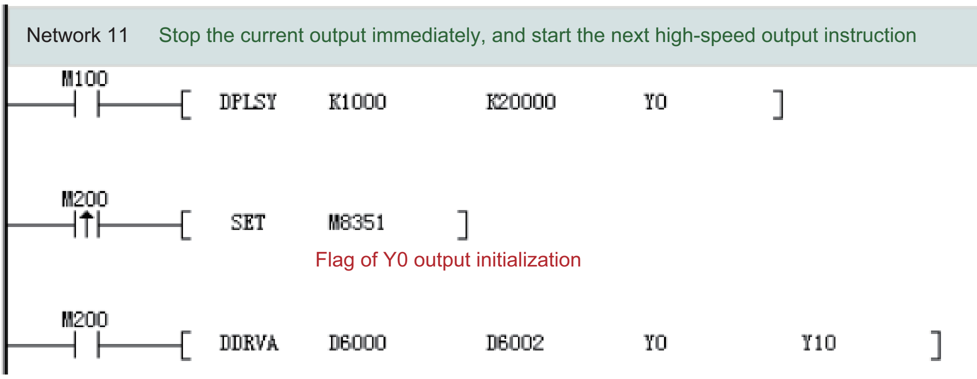


6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Pulse output complete interrupts.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

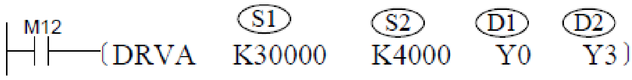
Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

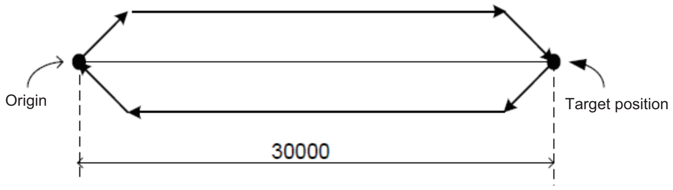
The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

◆ Program example



This instruction is used to control the actuator to move from the specified origin to the destination.



DRVI: Relative positioning

◆ Overview

A specified number of pulses are output at the specified pulse frequency in the specified direction through the specified output port. The action is based on the relative position.

DRVI S1 S2 D1 D2		Relative positioning	Applicable model: H3U	
S1	Pulse count	Specified number of pulse outputs	16-bit instruction (9 steps) DRVI: Continuous execution	32-bit instruction (17 steps) DDRVI: Continuous execution
S2	Output frequency	Specified pulse output frequency		
D1	Output port	High-speed pulse output port		
D2	Output direction	Pulse running direction port or bit variable		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output a specified number of pulses at the specified frequency and in the specified direction through the specified port, and drive the servo actuator to move toward the given offset based on the current position. This instruction is applicable only to the PLC of the transistor output type. Wherein:

6

S1 indicates the specified number of pulses to be output. For a 16-bit instruction, the range is –32768 to +32,767. For a 32-bit instruction, the range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction. The number of pulses to be output is used as the position relative to the current value registers listed in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

S2 indicates the specified output pulse frequency. For a 16-bit instruction, the range is 10 to 32,767 Hz. For a 32-bit instruction, the range is 10 to 200,000 Hz.

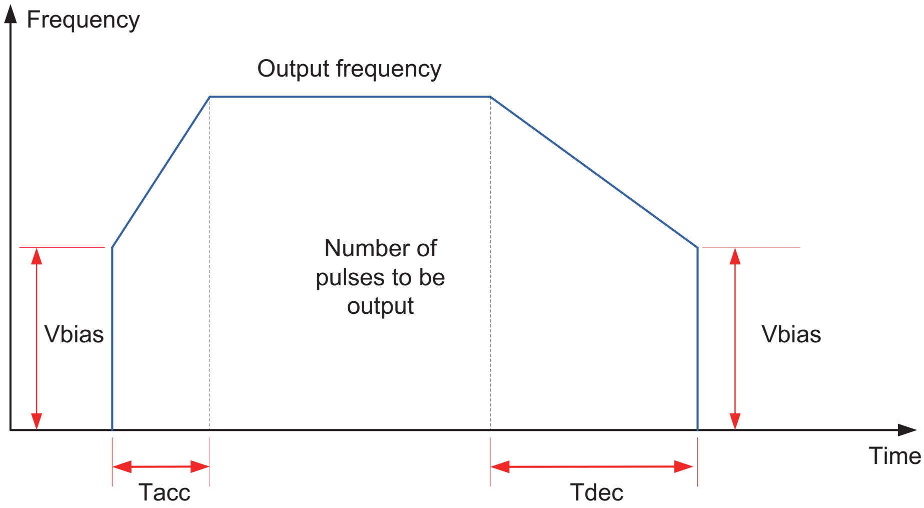
D1 indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the running direction of the output port or the bit variable. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

Even if the operand content is modified during instruction execution, the modification takes effect when the instruction is executed next time.

When the instruction flow switches to OFF during instruction execution , the system decelerates to stop. The execution complete flag M8029 takes no action. After the instruction flow switches to OFF, when the flag of pulse output is ON, the instruction is no longer driven by the instruction flow.

The following figure shows a pulse output diagram.



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This

6

flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

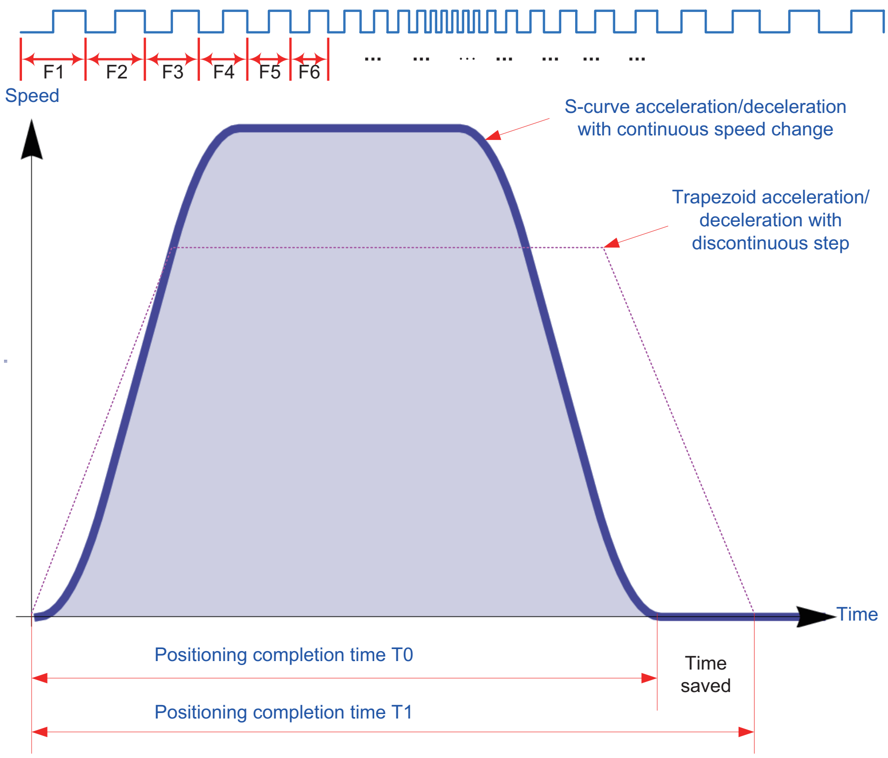
Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Trapezoid acceleration/deceleration and S-curve acceleration/deceleration are supported.

The two acceleration/deceleration modes can be distinguished by setting the "S-curve acceleration/deceleration enabling flag bit" of special elements. If the flag bit is not set, the trapezoid acceleration/deceleration mode is used by default. The following table lists details about S-curve acceleration/deceleration:

Y0	Y1	Y2	Y3	Y4	Attribute
M8347	M8367	M8387	M8407	M8427	S-curve acceleration/deceleration enabling

DRVI, DRVA, and PLSR support S-curve acceleration/deceleration. Therefore, at given mechanical stability, the target speed is increased, the positioning time is shortened, and the processing efficiency is improved.



The advanced pulse-by-pulse modulation algorithm is used for S-curve acceleration/deceleration. Frequency of each pulse is adjusted to ensure more smooth positioning.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms. The time of trapezoid acceleration/deceleration and S-curve acceleration/deceleration can be set separately.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

6

4) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

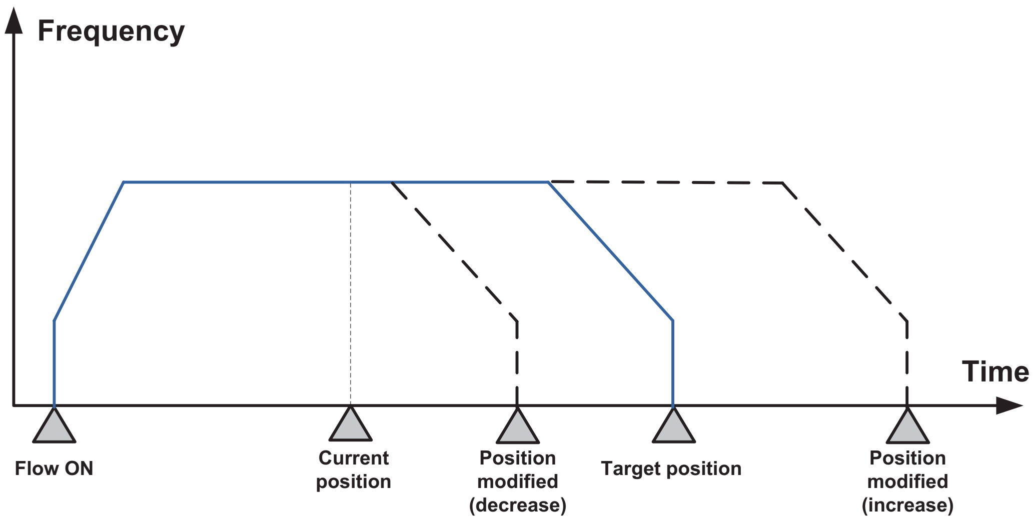
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

5) The number of output pulses can be increased or decreased during instruction execution.

Before modifying the number of output pulses, you need to set the "pulse modification valid flag bit" (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

Note that the modified position must be larger than the current pulse position. See the following figure.

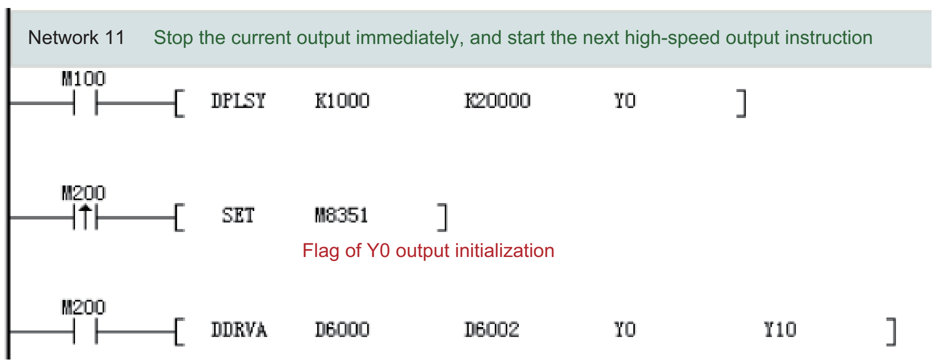


6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Pulse output complete interrupts.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

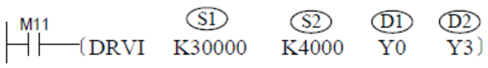
Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

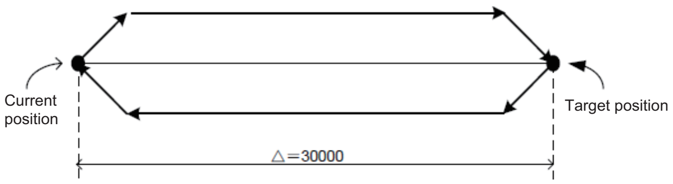
The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

◆ Program example



This instruction is used to output 30,000 pulses at a 4 kHz frequency through the port Y0, and drive the external servo actuator to move in the direction specified by Y3.



DVIT: Interrupt positioning

◆ Overview

The system starts and accelerates to the specified speed-segment output frequency. After detecting an interrupt input signal, the system immediately accelerates or decelerates to the position-segment output frequency, and outputs the specified number of pulses.

<table>
<tr><th colspan="3">DVIT S1 S2 S3 D1 D2 S4</th><th>Interruption positioning</th><th colspan="2">Applicable model: H3U</th></tr>
<tr><td>S1</td><td>Pulse count</td><td colspan="2">Number of pulses to be output at the position segment after the interrupt</td><td rowspan="6">16-bit instruction (13 steps)
DVIT: Continuous execution</td><td rowspan="6">32-bit instruction (25 steps)
DDVIT: continuous execution</td></tr>
<tr><td>S2</td><td>Output frequency 1</td><td colspan="2">Specified speed-segment pulse output frequency</td></tr>
<tr><td>S3</td><td>Output frequency 2</td><td colspan="2">Specified position-segment pulse output frequency after the interrupt</td></tr>
<tr><td>D1</td><td>Output port</td><td colspan="2">High-speed pulse output port</td></tr>
<tr><td>D2</td><td>Output direction</td><td colspan="2">Pulse running direction port or bit variable</td></tr>
<tr><td>S4</td><td>Interrupt input</td><td colspan="2">Interrupt input signal port (X0 to X7)</td></tr>
</table>

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output a specified number of pulses at the specified frequency and in the specified direction through the specified port; continue to output a specified number of pulses after an interrupt signal is detected; and drive the servo actuator to move with the given offset based on the current position. This instruction is applicable only to the PLC of the transistor output type.

S1 indicates the specified number of pulses to be output. For a 16-bit instruction, the range is –32,768 to +32,767. For a 32-bit instruction, the range is –2,147,483,648 to +2,147,483,647. The "–" indicates the reverse direction. The pulse output direction is determined by whether the value is positive or negative.

The number of pulses to be output is used as the position relative to the current value registers listed in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

S2 indicates the pulse output frequency at the speed segment before the interrupt. For a 16-bit instruction, the range is 10 to 32676 Hz. For a 32-bit instruction, the range is 10 to 200,000 Hz.

S3 indicates the pulse output frequency at the position segment after the interrupt. For a 16-bit instruction, the range is 10 to 32676 Hz. For a 32-bit instruction, the range is 10 to 200,000 Hz.

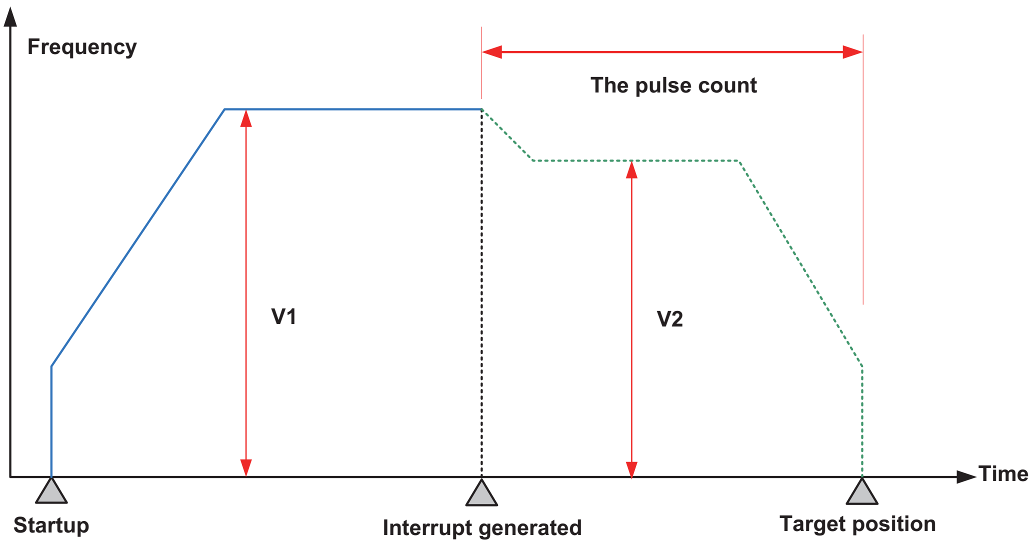
D1 indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the running direction of the output port or the bit variable. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

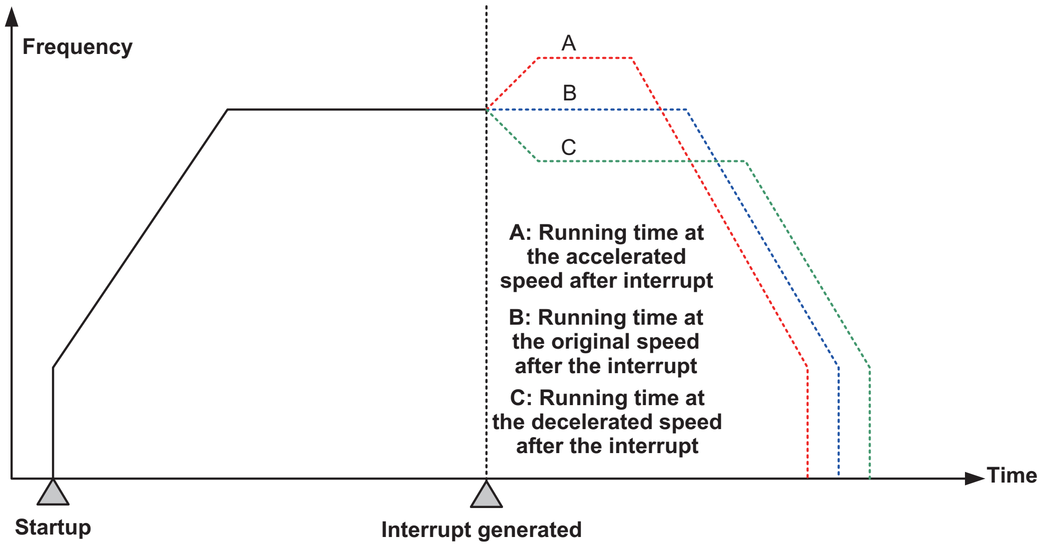
S4 indicates the specified interrupt signal output port, which can be any from X0 to X7.

When the instruction flow is OFF, the pulse output is decelerated to stop, and the execution complete flag M8029 remains unchanged. After the instruction driving point switches to OFF, when the flag is ON during pulse output, the flag is no longer driven by the instruction. When the flow switches from OFF to ON, the pulse output resumes. When the instruction execution is complete, the M8029 flag is set to ON.

The following figure shows a pulse output diagram.



The pulse output frequency at the speed segment before the interrupt may be different from that at the position segment after the interrupt, as shown in the figure below:



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Only trapezoid acceleration/deceleration is supported.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

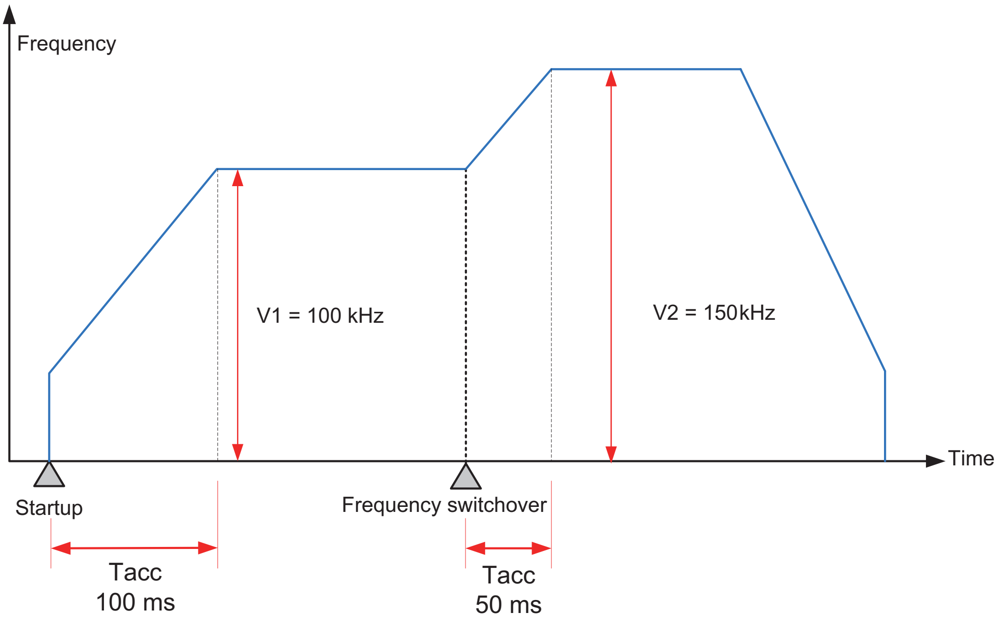
Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) During execution of a multi-segment pulse output instruction, the acceleration/deceleration computation is subject to the first speed. The acceleration/deceleration slope (acceleration/deceleration speed) remains unchanged during frequency switch.

For example, if output frequency is 100 kHz and the acceleration time is 100 ms at the first speed while the output frequency at the second speed is 150 kHz, it takes about 50 ms to accelerate from the first speed to the second speed. It works similarly in deceleration mode. See the following figure.



5) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

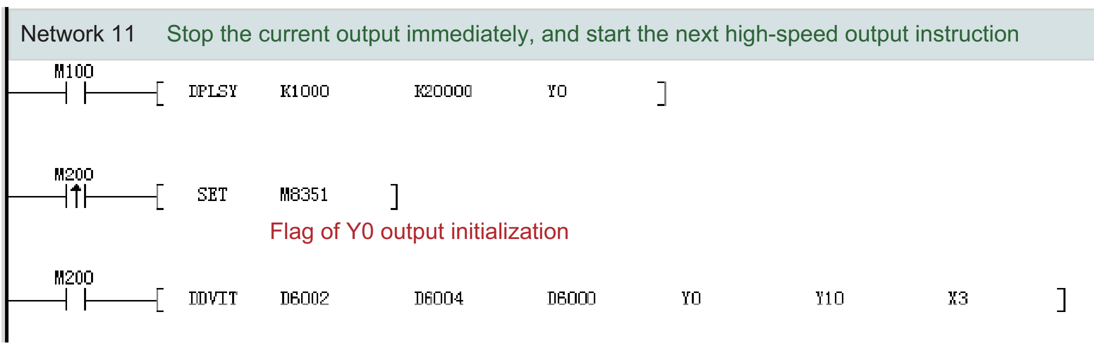
$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Pulse output complete interrupts.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

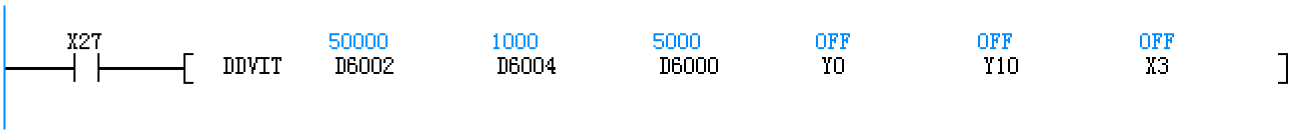
Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

◆ Program example



This instruction is used to drive the system to accelerate from the 1000 Hz output frequency in the forward direction at the speed segment to 5000 Hz after detecting an interrupt at the X3 rising edge, and output 50,000 pulses.

PLSN: Multi-speed pulse output

◆ Overview

This instruction is used to continuously output a specified number of output pulses at the output frequency specified at each segment through the specified output port. The action is based on the relative position. During running, acceleration/deceleration is supported, but the direction cannot be changed.

PLSN S1 S2 D1 D2 S3		Multi-speed pulse output	Applicable model: H3U	
S1	Pulse count	Specified number of pulses to be output at the first segment (which is S1 + 1 x n for a 16-bit instruction or S1 + 2 x n for a 32-bit instruction, indicating the number of pulses to be output at multiple segments)	16-bit instruction (11 steps) PLSN: Continuous execution	32-bit instruction (21 steps) DPLSN: Continuous execution
S2	Output frequency	Specified frequency of pulses to be output at the first segment (which is S2 + 1 x n for a 16-bit instruction or S2 + 2 x n for a 32-bit instruction, indicating the number of pulses to be output at multiple segments)		
D1	Output port	High-speed pulse output port		
D2	Output direction	Pulse running direction port or bit variable		
S3	Number of the multi-segments	Number of the multi-segments (2 to 16)		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output a specified number of pulses at the specified frequency and in the specified direction through the specified port; continue to output a specified number of pulses after an interrupt signal is detected; and drive the servo actuator to move with the given offset based on the current position. This instruction is applicable only to the PLC of the transistor output type. Wherein:

S1 indicates the specified number of pulses to be output at the first segment. The number of pulses to be output at other segments is stored in subsequent consecutive elements. For a 16-bit instruction, the range is -32768 to 32,767. For a 32-bit instruction, the range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction. The pulse output direction is determined by whether the value is positive or negative.

The number of pulses to be output is used as the position relative to the current value registers listed in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

S2 indicates the specified pulse output frequency at the first segment. The pulse output frequencies at other segments are stored in subsequent consecutive elements. For a 16-bit instruction, the range is 10 to 32,767 Hz. For a 32-bit instruction, the range is 10 to 200,000 Hz.

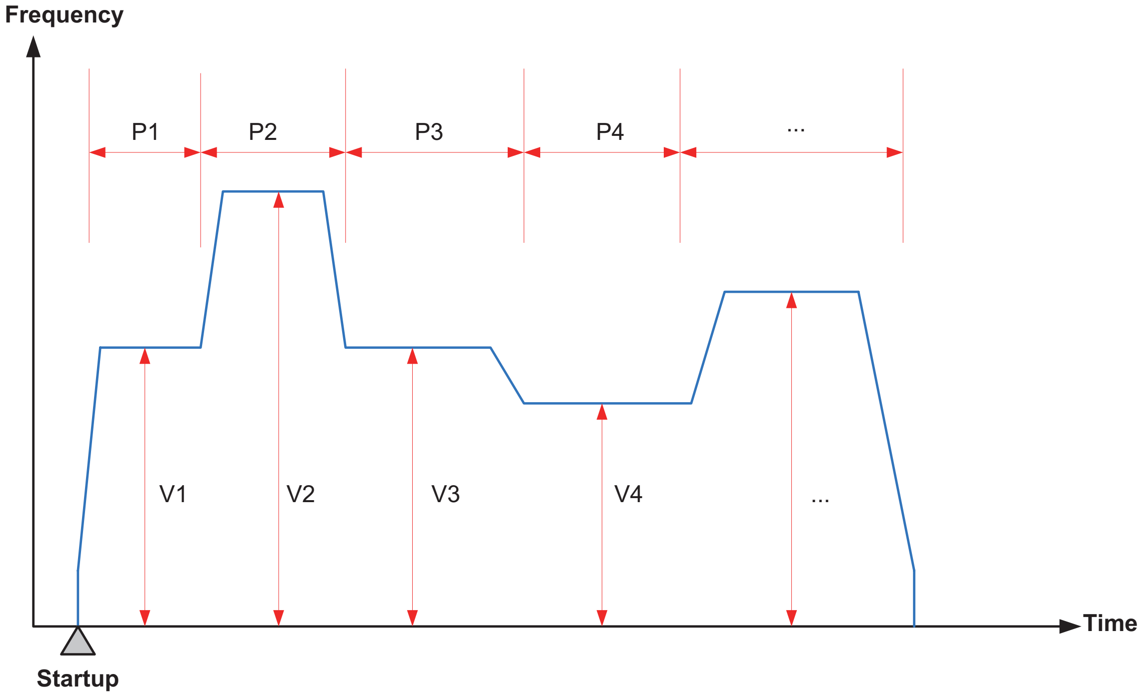
D1 indicates the pulse output port which can be Y0, Y1, Y2, Y3, or Y4.

D2 indicates the running direction of the output port or the bit variable. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

S3 indicates the specified number of the pulse segments, in the range 2 to 16.

When the instruction flow is OFF, the pulse output is decelerated to stop, and the execution complete flag M8029 takes no action. After the instruction driving point switches to OFF, when the flag is ON during pulse output, the flag is not driven again by the instruction. When the flow switches from OFF to ON, the pulse output resumes. When the instruction execution is complete, the M8029 flag is set to ON.

The following figure shows a pulse output diagram.



◆ Note

1) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

2) Only trapezoid acceleration/deceleration is supported.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

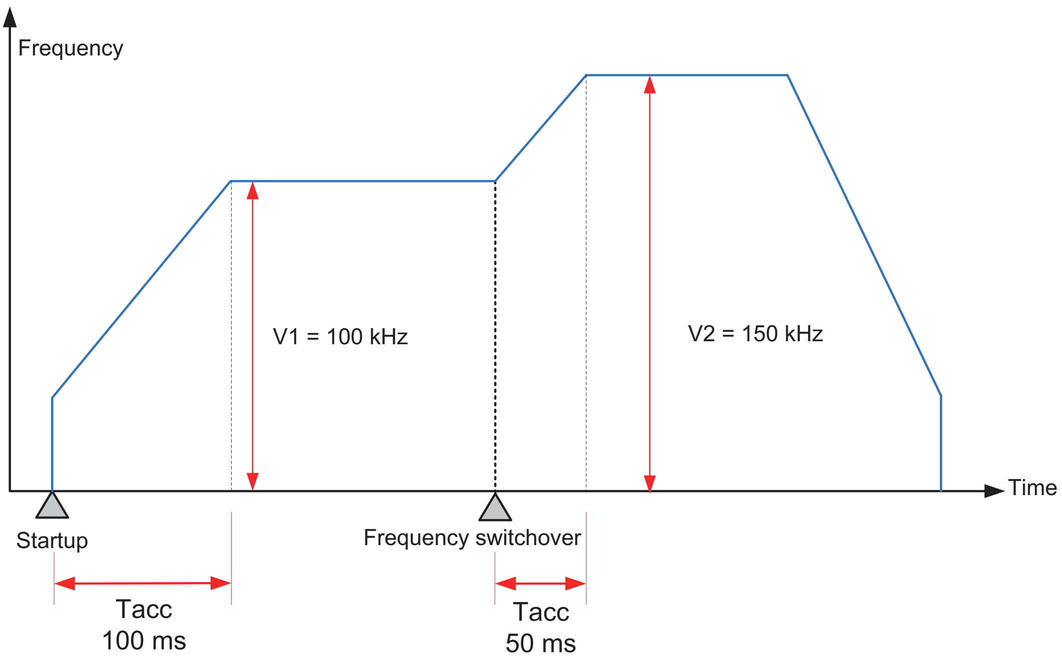
Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

4) During execution of a multi-segment pulse output instruction, the acceleration/deceleration computation is subject to the first speed. The acceleration/deceleration slope (acceleration/deceleration speed) remains unchanged during frequency switch.

For example, if output frequency is 100 kHz and the acceleration time is 100 ms at the first speed while the output frequency at the second speed is 150 kHz, it takes about 50 ms to accelerate from the first speed to the second speed. It works similarly in deceleration mode. See the following figure.



The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

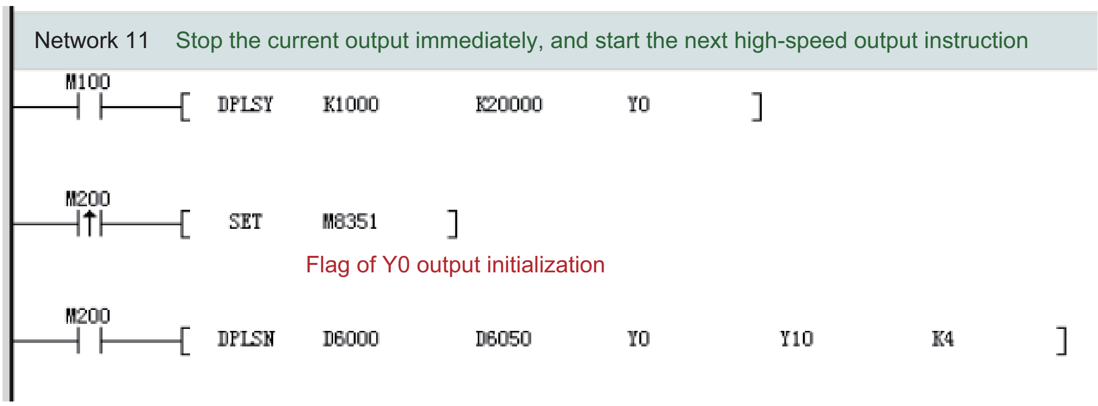
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

5) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

6) Pulse output complete interrupts.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

7) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

6

◆ Program example

It indicates four segments of continuous pulse output in total. In the first segment, 100,000 pulses are output at a frequency of 20 kHz.

In a 32-bit instruction, D6000, D6002, D6004, and D6008 indicate the number of four-segment pulses.

In a 32-bit instruction, D6050, D6052, D6054, and D6058 indicate the corresponding four-segment pulse output frequencies.

6.3 Interpolation Instruction

The system supports interpolation based on the absolute position and interpolation based on the relative position. Based on the interpolation path, the system supports 2-axis linear interpolation, 2-axis clockwise arc interpolation, and 2-axis counterclockwise arc interpolation. The following table lists the relevant instructions.

H3U Model Interpolation	"G90G01: 2-axis linear absolute position interpolation"
	"G91G01: 2-axis linear relative position interpolation"
	"G90G02: 2-axis clockwise absolute position arc interpolation"
	"G91G02: 2-axis clockwise relative position arc interpolation"
	"G90G03: 2-axis counterclockwise absolute position arc interpolation"
	"G91G03: 2-axis counterclockwise relative position arc interpolation"

The preceding instructions support only 32-bit operation. The pulse execution type is not supported.

G90G01: 2-axis linear absolute position interpolation

◆ Overview

This instruction is used to output the set interpolation path at the set combined output frequency. The action is based on the absolute position.

G90G01 S1 S2 S D1 D2			2-axis linear absolute position interpolation	Applicable Model: H3U
S1	X pulse count	Absolute value of the target position on the x-axis (Y0)		32-bit instruction (21 steps) G90G01 continuous execution
S2	Y pulse count	Absolute value of the target position on the y-axis (Y1)		
S	Output frequency	Combined output interpolation frequency		
D1	Output port	High-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.		
D2	Output direction	Port in the pulse running direction		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output pulses at the specified frequency and in the specified direction through the specified port; perform 2-axis linear interpolation; and drive the servo actuator to move to the specified target position according to the linear interpolation. This instruction is applicable only to the PLC of the transistor output type.

- S1 indicates the specified destination (absolute position) of the x-axis. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S2 indicates the specified destination (absolute position) of the y-axis. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

The following table lists the absolute position current values.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

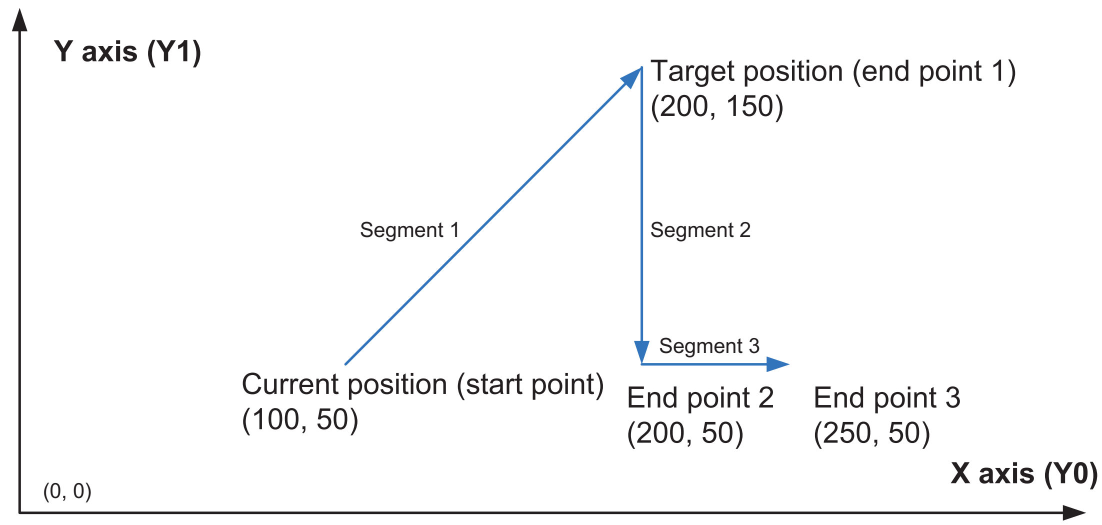
The numerical value of the current value register decreases in the reverse direction.

- S indicates the combined pulse output frequency of the specified x- and y-axes, ranging from 50 to 280,000 Hz. When the combined frequency is allocated to the x- and y-axes, the pulse output frequency of each axis ranges from 50 to 200,000 Hz.
- D1 indicates the high-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.
- D2 indicates the running direction of the output port or the bit variable. Only the Y port can be specified, and two consecutive Y ports are occupied. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

When the instruction flow is OFF, the pulse output is decelerated to stop, and the execution complete flag M8029 takes no action. After the instruction driving point switches to OFF, when the flag is ON during pulse output, the flag is not driven again by the instruction.

The following figure shows a pulse output diagram.

6



In the instruction, S1 and S2 indicate the target absolute positions of x- and y-axes, such as (200,150) in the preceding figure above. When the interpolation instruction (G90G01 or G91G01) is used, the following functions are supported: 2-axis interpolation (the first segment in the preceding figure) and single-axis positioning (the second and third segments in the preceding figure).

◆ Program example

Assume that the current position is (100K, 50K), which indicates linear interpolation from the current position to (200K, 150K) at a combined frequency of 100 kHz. Y0 and Y1 are the pulse output ports, and Y10 and Y11 are the pulse direction output ports.

◆ Note

1) When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G03, or G91G03) is used, parameters of the x-axis (Y0) such as the acceleration/deceleration time prevail.

2) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

5) The actual minimum combined output frequency (that is, the minimum combined base output frequency S) is calculated according to the following formula:

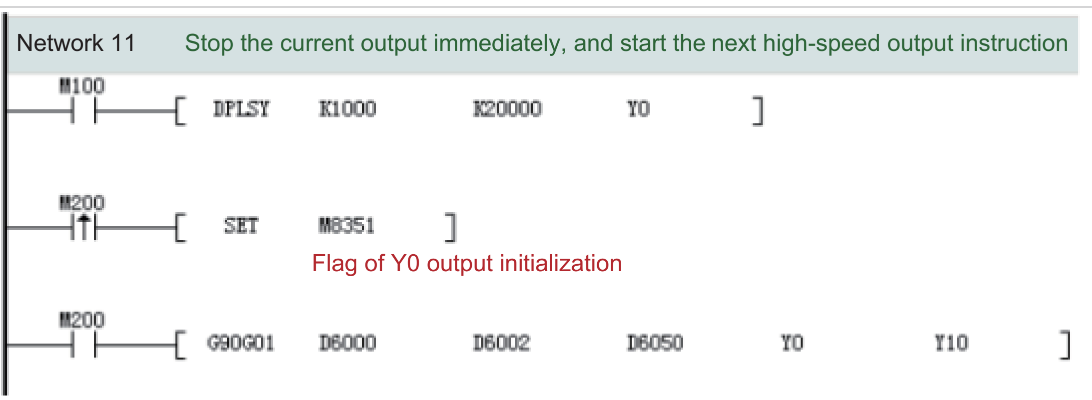
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

1) Interpolation at the x- and y-axes (Y0/Y1) results in output complete interrupt of only one pulse.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

2) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

6

G91G01: 2-axis linear relative position interpolation

◆ Overview

This instruction is used to output the set interpolation path at the set combined output frequency. The action is based on the relative position.

G91G01 S1 S2 S D1 D2		2-axis linear relative position interpolation	Applicable model: H3U	
S1	X pulse count	Difference of the pulse output count of the target position relative to that in the current position at the x-axis (Y0)		32-bit instruction (21 steps) G91G01; Continuous execution
S2	Y pulse count	Difference of the pulse output count in the target position relative to that in the current position at the y-axis (Y1)		
S	Output frequency	Combined output interpolation frequency		
D1	Output port	High-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.		
D2	Output direction	Port in the pulse running direction		

6

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output a specified number of pulses at the specified frequency and in the specified direction through the specified port; perform the 2-axis linear interpolation; and drive the servo actuator to perform 2-axis linear interpolation with the given offset based on the current position. This instruction is applicable only to the PLC of the transistor output type. Wherein:

- S1 indicates the specified number of pulses output at the x-axis (offset). The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S2 indicates the specified number of pulses output at the y-axis (offset). The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

The number of pulses to be output is used as the position relative to the current value registers listed in the following table.

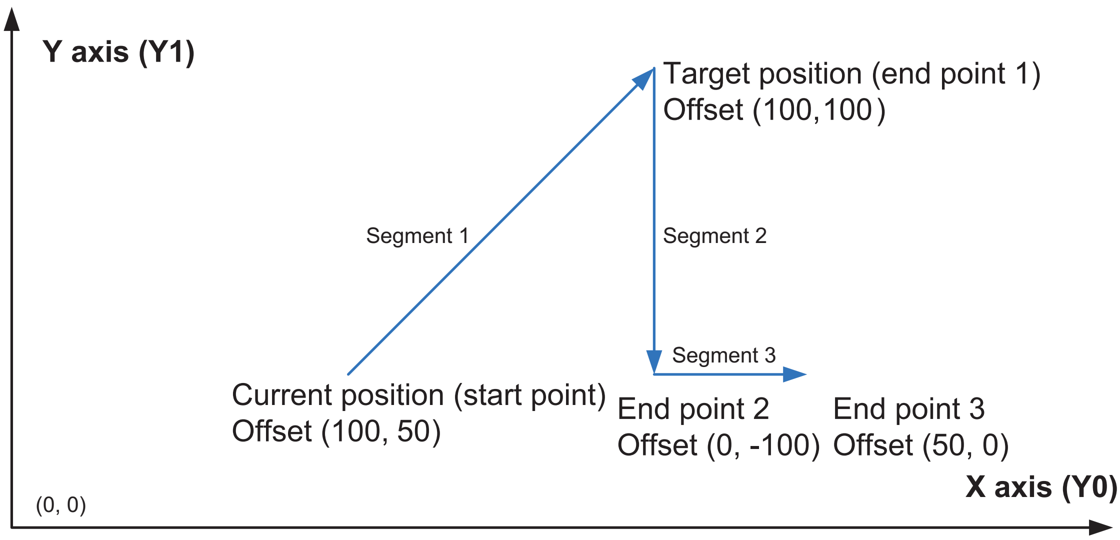
Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

- S indicates the combined pulse output frequency of the specified x- and y-axes, ranging from 0 to 280,000 Hz. When the combined frequency is allocated to the x- and y-axes, the pulse output frequency of each axis ranges from 50 to 200,000 Hz.
- D1 indicates the high-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.
- D2 indicates the running direction of the output port or the bit variable. Only the Y port can be specified, and two consecutive Y ports are occupied. If output is ON, it means running in the forward direction; otherwise, it means running in the reverse direction.

When the instruction flow is OFF, the pulse output is decelerated to stop, and the execution complete flag M8029 takes no action. After the instruction driving point switches to OFF, when the flag is ON during pulse output, the flag is not driven again by the instruction.

The following figure shows a pulse output diagram.



In the instruction, S1 and S2 indicate the target relative positions of x- and y-axes, such as (100,100) in the preceding figure, When the interpolation instruction (G90G01 or G91G01) is used, supported are 2-axis interpolation (the first segment in the preceding figure) and single-axis positioning (the second and third segments in the preceding figure).

◆ Program example

Assume that the current position is (100K, 50K), which indicates linear interpolation from the current position at an offset (100K, 100K), that is, to the position (200K, 150K) at a combined frequency of 100 kHz. Y0/Y1 are the pulse output ports, and Y10/Y11 are the pulse direction output ports.

◆ Note

1) When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters, such as the acceleration/deceleration time, of the X-axis (Y0) prevail.

2) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

5) The actual minimum combined output frequency (that is, the minimum combined base output frequency S) is calculated according to the following formula:

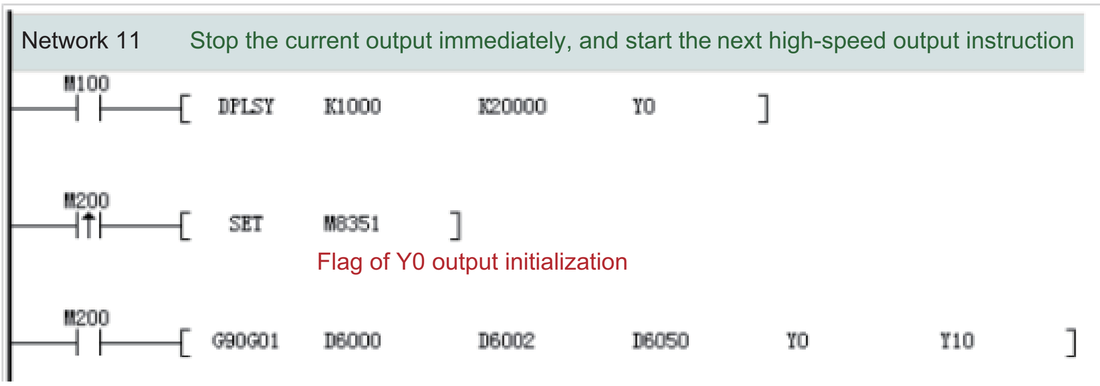
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Interpolation at x- and y-axes (Y0/Y1) results in output complete interrupt of only one pulse.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

G90G02: 2-axis clockwise absolute position arc interpolation

◆ Overview

This instruction is used to output the set clockwise arc interpolation path at the set combined output frequency. The action is based on the absolute position.

G90G02 S1 S2 S3 S4 S D1 D2		Clockwise absolute position arc interpolation	Applicable model: H3U
S1	X pulse count	Absolute value of the target position on the x-axis (Y0)	32-bit instruction (29 steps) G90G02: Continuous execution
S2	Y pulse count	Absolute value of the target position on the y-axis (Y1)	
S3	X center coordinate	Difference of the pulse output count of the center coordinate relative to that in the current position at the x-axis (Y0), or the pulse count of the radius (R)	
S4	Y center coordinate	Difference of the pulse output count of the center coordinate relative to that in the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF.	
S	Output frequency	Combined output interpolation frequency	
D1	Output port	High-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.	
D2	Output direction	Pulse running direction port. Only Y2 can be specified, and Y2/Y3 can be occupied.	

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output pulses at the specified frequency and in the specified direction through the specified port; perform 2-axis clockwise arc interpolation; and drive the servo actuator to move to the specified target position according to the clockwise arc interpolation. This instruction is applicable only to the PLC of the transistor output type. Wherein:

S1 indicates the specified destination position (absolute position) of the X-axis. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

S2 indicates the specified destination (absolute position) of the y-axis. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

The following table lists the absolute position current values.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

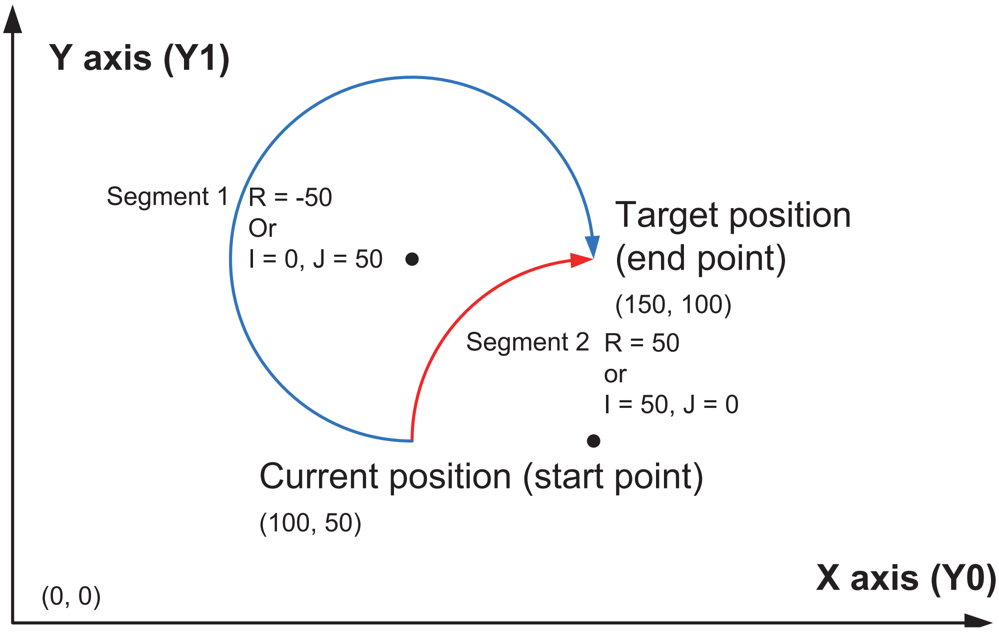
The numerical value of the current value register decreases in the reverse direction.

- S3 indicates the difference of the pulse output count of the specified center coordinate relative to that in the current position at the x-axis (Y0), or the pulse count of the radius (R) The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S4 indicates the difference of the pulse output count of the center coordinate relative to that in the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF. The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S indicates the combined pulse output frequency of the specified x- and y-axes, ranging from 50 to 200,000 Hz.
- D1 indicates the high-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.
- D2 indicates the running direction output port or the bit variable. Only Y2 can be specified, and Y2/Y3 can be occupied.

Note:

- S1/S2 indicates the absolute position of the target position. The user needs to set an appropriate target position so that the correct target circular path can be generated. When the specified target position of the x- and y-axes equals their current position, a complete circle is generated.
- S3/S4 can be set either in IJ (center coordinate) mode or R (radius) mode. If the S4 value is set to 0x7FFF FFFF, it is in R (radius) mode; otherwise, it is in IJ (center coordinate) mode.
- In IJ (center coordinate) mode, no matter it is absolute position interpolation or relative position interpolation, S3/S4 only indicates the difference (offset) of the central coordinate relative to the current position on the x- and y-axes (Y0/Y1).
- In R (radius) mode, When the R value is larger than 0, it indicates an arc less than or equal to 180 degrees. When the R value is smaller than 0, it indicates an arc more than 180 degrees. In R (radius) mode, no complete circle can be generated.
- More than 20 pulses must be output along the arc during arc interpolation; otherwise, an error is returned.
- Up to 8,000,000 pulses can be output along the radius during arc interpolation.
- When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters of Y0 such as the acceleration/deceleration time prevail.
- When the instruction flow switches to OFF during instruction execution , the system decelerates to stop. The execution complete flag M8029 remains unchanged. After the instruction- driven contact switches to OFF, when the flag is ON during pulse output, the contact is no longer driven by the instruction.

The following figure shows a pulse output diagram.



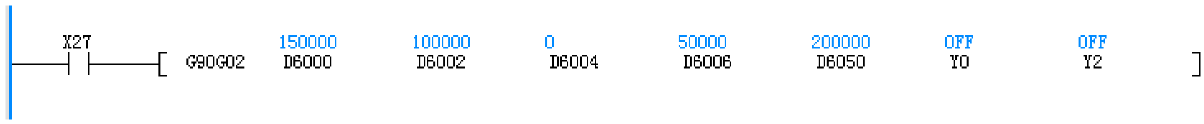
It indicates a clockwise arc interpolation, wherein S1/S2 indicates the target absolute position of the x- and y-axes, such as (150,100) in the preceding figure. When the target position is the same, the example for generation of an arc less than 180 degrees and more than 180 degrees when S3/S4 is in IJ (center coordinate) mode and in R (radius) mode respectively is provided.

NOTE

- In the first segment, (I, J) = Current center coordinate – Starting position coordination [(100, 100) – (100, 50)] = (0, 50).

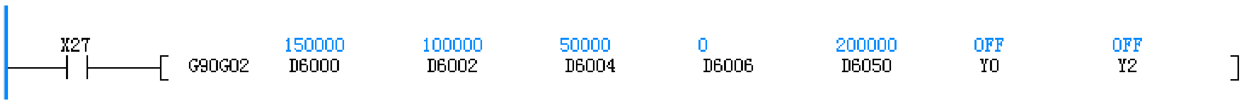
◆ Program example

In the first arc, the IJ (center coordinate) mode is used:



In the first arc, the R (radius) mode is used:

In the second arc, the IJ (center coordinate) mode is used:



In the second arc, the R (radius) mode is used:

◆ Note

1) When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters, such as the acceleration/deceleration time, of the X-axis (Y0) prevail.

2) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

5) The actual minimum combined output frequency (that is, the minimum combined base output frequency S) is calculated according to the following formula:

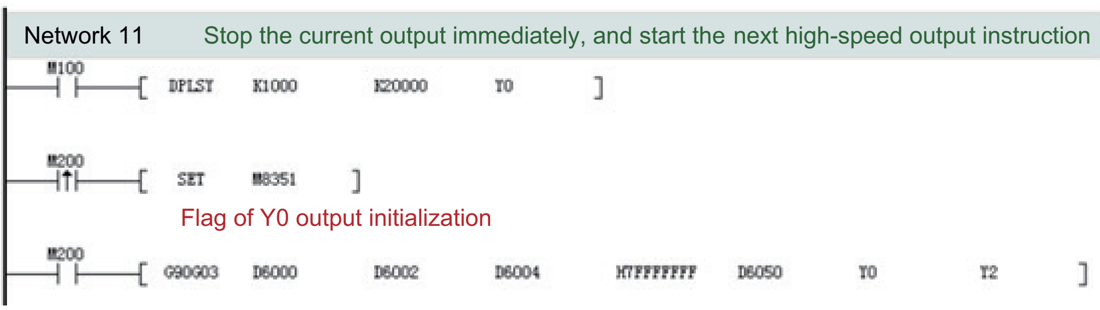
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Interpolation at x- and y-axes (Y0/Y1) results in output complete interrupt of only one pulse.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

G91G02: 2-axis clockwise relative position arc interpolation

◆ Overview

This instruction is used to output the set clockwise arc interpolation path at the set combined output frequency. The action is based on the relative position.

G91G02 S1 S2 S3 S4 S D1 D2		Clockwise relative position arc interpolation	Applicable model: H3U	
S1	X pulse count	Difference of the pulse output count of the target position relative to that of the current position at the X-axis (Y0)		32-bit instruction (29 steps) G91G02: Continuous execution
S2	Y pulse count	Difference of the pulse output count of the target position relative to that of the current position at the y-axis (Y1)		
S3	X center coordinate	Difference of the pulse output count of the center coordinate relative to that of the current position at the X-axis (Y0), or the pulse count of the radius (R)		
S4	Y center coordinate	Difference of the pulse output count of the center coordinate relative to that of the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF.		
S	Output frequency	Combined output interpolation frequency		
D1	Output port	High-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.		
D2	Output direction	Pulse running direction port. Only Y2 can be specified, and Y2/Y3 can be occupied.		

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output a specified number of pulses at the specified frequency and in the specified direction through the specified port; perform the 2-axis arc interpolation; and drive the servo actuator to perform 2-axis arc interpolation with the given offset based on the current position. This instruction is applicable only to the PLC of the transistor output type. Wherein:

S1 indicates the specified number of pulses output at the end point of the x-axis relative to the starting point (offset). The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

S2 indicates the specified number of pulses output at the end point of the y-axis relative to the starting point (offset). The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

The number of pulses to be output is used as the position relative to the current value registers listed in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

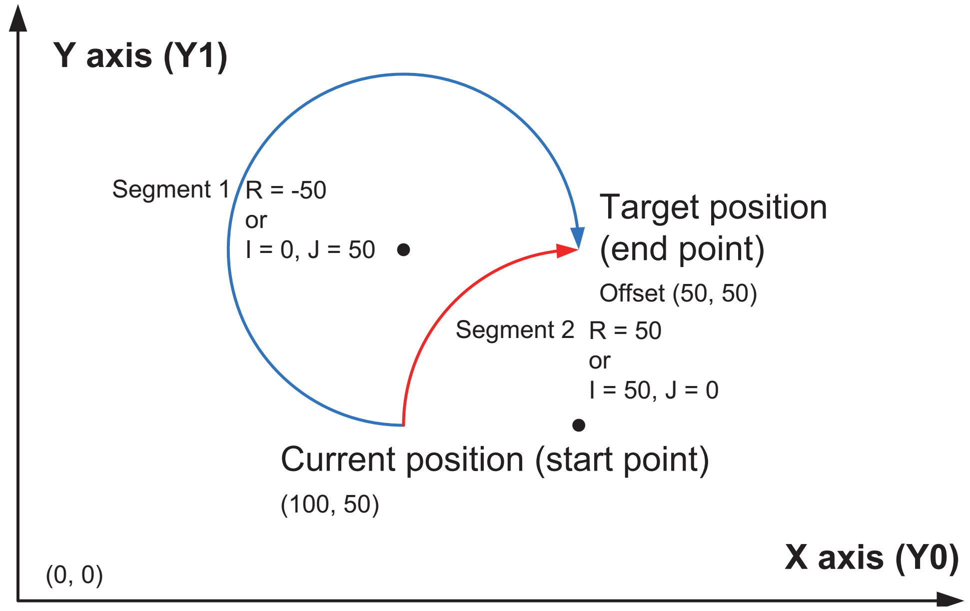
- S3 indicates the difference value of the pulse output count of the specified center coordinate relative to that of the current position at the X-axis (Y0), or the pulse count of the radius R. The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S4 indicates the difference of the pulse output count of the center coordinate relative to that in the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S indicates the combined pulse output frequency of the specified x- and y-axes, ranging from 50 to 200,000 Hz.
- D1 indicates the high-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.
- D2 indicates the running direction output port or the bit variable. Only Y2 can be specified, and Y2/Y3 can be occupied.

6

Note:

- S1/S2 indicates the absolute position of the target position. The user needs to set an appropriate target position so that the correct target circular path can be generated. When S1 = 0 and S2 =0, a complete circle is generated.
- S3/S4 can be set either in IJ (center coordinate) or R (radius) mode. If the S4 value is set to 0x7FFF FFFF, it is in R (radius) mode; otherwise, it is in IJ (center coordinate) mode.
- In the IJ (center coordinate) mode, no matter it is absolute position interpolation or relative position interpolation, S3/S4 only indicates the difference (offset) of the central coordinate relative to the current position on the x- and y-axes (Y0/Y1).
- In the R (radius) mode, When the R value is larger than 0, it indicates an arc less than or equal to 180 degrees. When the R value is smaller than 0, it indicates an arc more than 180 degrees. In the R (radius) mode, no complete circle can be generated.
- More than 20 pulses shall be output along the arc during arc interpolation; otherwise, an error is returned.
- Up to 8,000,000 pulses can be output along the radius during arc interpolation.
- When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters of Y0 such as the acceleration/deceleration time prevail.
- When the instruction flow switches to OFF during instruction execution , the system decelerates to stop. The execution complete flag M8029 remains unchanged. After the instruction- driven contact switches to OFF, when the flag is ON during pulse output, the contact is no longer driven by the instruction.

The following figure shows a pulse output diagram.



It indicates a clockwise arc interpolation, wherein, S1/S2 indicates the target relative position of the x- and y-axes, such as (50,50) in the preceding figure. When the target position is the same, the example for generation of an arc less than 180 degrees and more than 180 degrees when S3/S4 is in IJ (center coordinate) mode and in R (radius) mode respectively is provided.

NOTE

- In the first segment, (I,J) = current center coordinate - Starting position coordination ((100,100) - (100,50)) = (0,50).

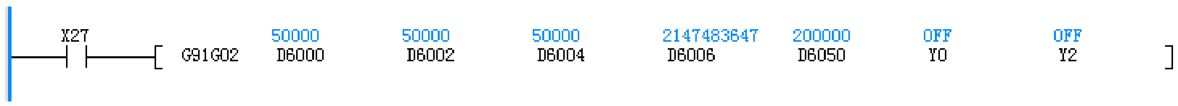
◆ Program example

In the first arc, the IJ (center coordinate) mode is used:

In the first arc, the R (radius) mode is used:

In the second arc, the IJ (center coordinate) mode is used:

In the second arc, the R (radius) mode is used:



◆ Note

1) When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters, such as the acceleration/deceleration time, of the X-axis (Y0) prevail.

2) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

5) The actual minimum combined output frequency (that is, the minimum combined base output frequency S) is calculated according to the following formula:

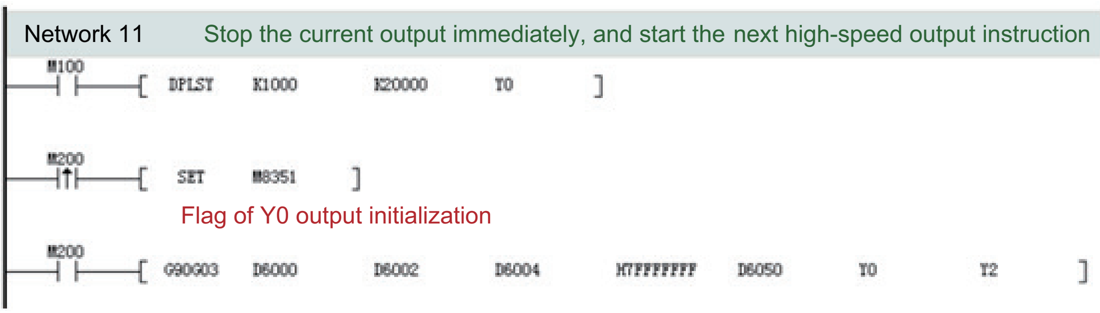
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Interpolation at x- and y-axes (Y0/Y1) results in output complete interrupt of only one pulse.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

6

G90G03: 2-axis counterclockwise absolute position arc interpolation

◆ Overview

This instruction is used to output the set counterclockwise arc interpolation path at the set combined output frequency. The action is based on the absolute position.

G90G03 S1 S2 S3 S4 S D1 D2		Anticlockwise absolute position arc interpolation	Applicable model: H3U	
S1	X pulse count	Absolute value of the target position on the x-axis (Y0)		32-bit instruction (29 steps) G90G03: Continuous execution
S2	Y pulse count	Absolute value of the target position on the y-axis (Y1)		
S3	X center coordinate	Difference of the pulse output count of the center coordinate relative to that in the current position at the x-axis (Y0), or the pulse count of the radius (R)		
S4	Y center coordinate	Difference of the pulse output count of the center coordinate relative to that of the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF.		
S	Output frequency	Combined output interpolation frequency		
D1	Output port	High-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.		
D2	Output direction	Pulse running direction port. Only Y2 can be specified, and Y2/Y3 can be occupied.		

6

◆ Operands

Operand	Bit Element							Word Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

This instruction is used to output pulses at the specified frequency and in the specified direction through the specified port; perform 2-axis counterclockwise arc interpolation; and drive the servo actuator to move to the specified target position according to the counterclockwise arc interpolation. This instruction is applicable only to the PLC of the transistor output type. Wherein:

S1 indicates the specified destination (absolute position) of the x-axis. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

S2 indicates the specified destination (absolute position) of the y-axis. The range is –2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

The following table lists the absolute position current values.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

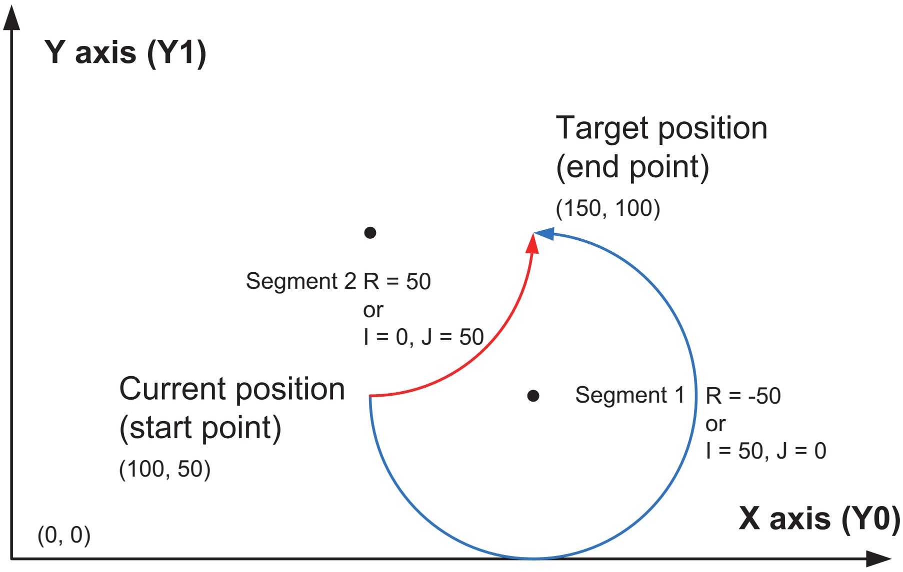
The numerical value of the current value register decreases in the reverse direction.

- S3 indicates the difference value of the pulse output count of the specified center coordinate relative to that of the current position at the X-axis (Y0), or the pulse count of the radius. The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S4 indicates the difference value of the pulse output count of the center coordinate relative to that of the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF. The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S indicates the combined pulse output frequency of the specified x- and y-axes, ranging from 50 to 200,000 Hz.
- D1 indicates the high-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.
- D2 indicates the running direction output port or the bit variable. Only Y2 can be specified, and Y2/Y3 can be occupied.

Note:

- S1/S2 indicates the absolute position of the target position. The user needs to set an appropriate target position so that the correct target circular path can be generated. When the specified target position of axes X and Y equals to their current position, a complete circle is generated.
- S3/S4 can be set either in IJ (center coordinate) or R (radius) mode. If the S4 value is set to 0x7FFF FFFF, it is in R (radius) mode; otherwise, it is in IJ (center coordinate) mode.
- In IJ (center coordinate) mode, no matter it is absolute position interpolation or relative position interpolation, S3/S4 only indicates the difference (offset) of the central coordinate relative to the current position on the x- and y-axes (Y0/Y1).
- In R (radius) mode, When the R value is larger than 0, it indicates an arc less than or equal to 180 degrees. When the R value is smaller than 0, it indicates an arc more than 180 degrees. In R (radius) mode, no complete circle can be generated.
- More than 20 pulses must be output along the arc during arc interpolation; otherwise, an error is returned.
- Up to 8,000,000 pulses can be output along the radius during arc interpolation.
- When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters of Y0 such as the acceleration/deceleration time prevail.
- When the instruction flow switches to OFF during instruction execution , the system decelerates to stop. The execution complete flag M8029 remains unchanged. After the instruction flow switches to OFF, when the flag of pulse output is ON, the instruction is no longer driven by the instruction flow.

The following figure shows a pulse output diagram.



It indicates an counterclockwise arc interpolation, wherein, S1/S2 indicates the target relative position of the x- and y-axes, such as (150,100) in the preceding figure. When the target position is the same, the example for generation of an arc less than 180 degrees and more than 180 degrees when S3/S4 is in IJ (center coordinate) mode and in R (radius) mode respectively is provided.

NOTE

- In the first segment, (I,J) = current center coordinate - starting position coordination ((150,50) - (100,50)) = (50,0).

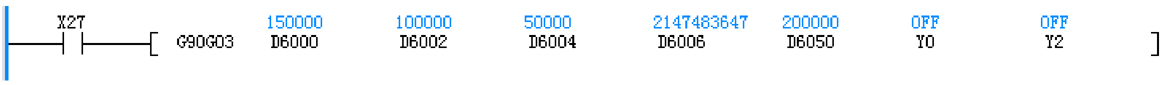
◆ Program example

In the first arc, the IJ (center coordinate) mode is used:

In the first arc, the R (radius) mode is used:

In the second arc, the IJ (center coordinate) mode is used:

In the second arc, the R (radius) mode is used:



◆ Note

1) When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters, such as the acceleration/deceleration time, of the X-axis (Y0) prevail.

2) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

5) The actual minimum combined output frequency (that is, the minimum combined base output frequency S) is calculated according to the following formula:

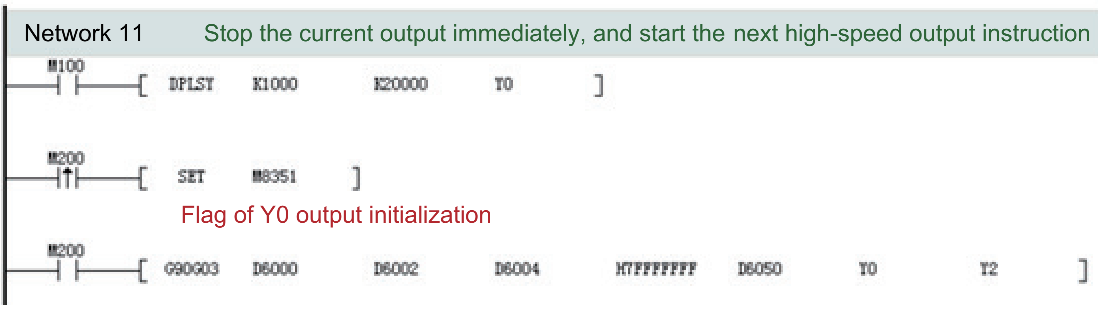
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Interpolation at x- and y-axes (Y0/Y1) results in output complete interrupt of only one pulse.

The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

8) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

G91G03: 2-axis counterclockwise relative position arc interpolation

◆ Overview

This instruction is used to output the set counterclockwise arc interpolation path at the set combined output frequency. The action is based on the relative position.

G91G03 S1 S2 S3 S4 S D1 D2		Anticlockwise relative position arc interpolation	Applicable model: H3U	
S1	X pulse count	Difference of the pulse output count in the target position relative to that in the current position at the x-axis (Y0)		32-bit instruction (29 steps) G91G03: Continuous execution
S2	Y pulse count	Difference of the pulse output count of the target position relative to that of the current position at the y-axis (Y1)		
S3	X center coordinate	Difference of the pulse output count of the center coordinate relative to that of the current position at the X-axis (Y0), or the pulse count of the radius (R)		
S4	Y center coordinate	Difference of the pulse output count of the center coordinate relative to that of the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF.		
S	Output frequency	Combined output interpolation frequency		
D1	Output port	High-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.		
D2	Output direction	Pulse running direction port. Only Y2 can be specified, and Y2/Y3 can be occupied.		

◆ Operands

Operand	Bit Element							Word Element																
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The elements in gray background are supported.

6

◆ Functions and actions

This instruction is used to output a specified number of pulses at the specified frequency and in the specified direction through the specified port; perform the 2-axis counterclockwise arc interpolation; and drive the servo actuator to perform 2-axis counterclockwise arc interpolation with the given offset based on the current position. This instruction is applicable only to the PLC of the transistor output type. Wherein:

- S1 indicates the specified number of pulses output at the end point of the x-axis relative to the starting point (offset). The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S2 indicates the specified number of pulses output at the end point of the y-axis relative to the starting point (offset). The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.

The number of pulses to be output is used as the position relative to the current value registers listed in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

The numerical value of the current value register decreases in the reverse direction.

- S3 indicates the difference value of the pulse output count of the specified center coordinate relative to that of the current position at the X-axis (Y0), or the pulse count of the radius R. The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S4 indicates the difference value of the pulse output count of the center coordinate relative to that of the current position at the y-axis (Y1). If S3 indicates R, S4 must be 0x7FFF FFFF. The range is -2,147,483,648 to +2,147,483,647. The "-" indicates the reverse direction.
- S indicates the combined pulse output frequency of the specified x- and y-axes, ranging from 50 to 200,000 Hz.
- D1 indicates the high-speed pulse output port. Only Y0 can be specified, and Y0/Y1 is occupied.
- D2 indicates the running direction output port or the bit variable. Only Y2 can be specified, and Y2/Y3 can be occupied.

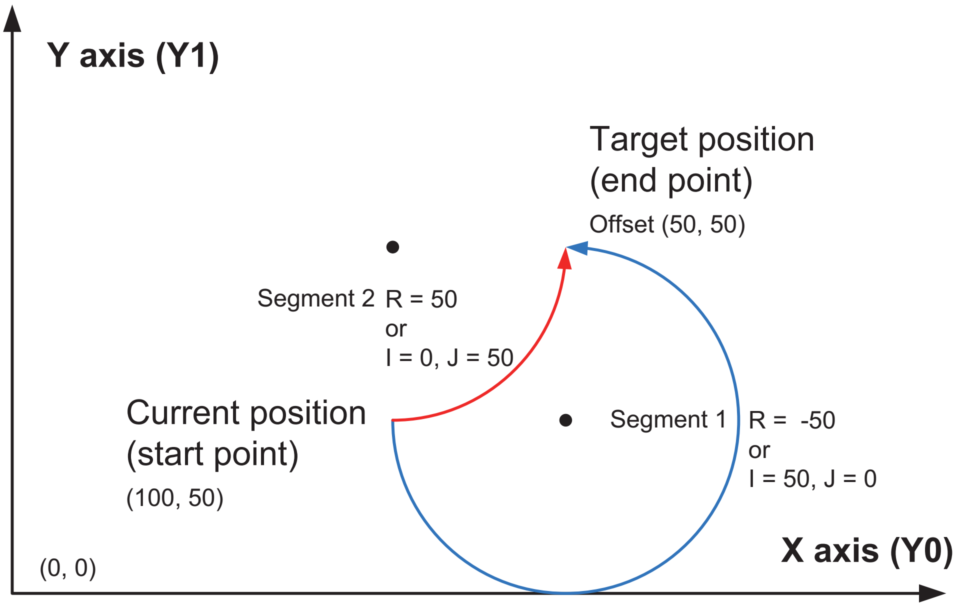
Note:

- S1/S2 indicates the absolute position of the target position. The user needs to set an appropriate target position so that the correct target circular path can be generated. When S1 = 0 and S2 =0, a complete circle is generated.
- S3/S4 can be set either in IJ (center coordinate) mode or in R (radius) mode. If the S4 value is set to 0x7FFF FFFF, it is in R (radius) mode; otherwise, it is in IJ (center coordinate) mode.
- In the IJ (center coordinate) mode, no matter it is absolute position interpolation or relative position interpolation, S3/S4 only indicates the difference (offset) of the central coordinate relative to the current position on the x- and y-axes (Y0/Y1).
- In the R (radius) mode, When the R value is larger than 0, it indicates an arc less than or equal to 180 degrees. When the R value is smaller than 0, it indicates an arc more than 180 degrees. In the R (radius) mode, no complete circle can be generated.
- More than 20 pulses shall be output along the arc during arc interpolation; otherwise, an error is returned.
- Up to 8,000,000 pulses can be output along the radius during arc interpolation.
- When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is

used, parameters of Y0 such as the acceleration/deceleration time prevail.

- When the instruction flow switches to OFF during instruction execution , the system decelerates to stop. The execution complete flag M8029 remains unchanged. After the instruction flow switches to OFF, when the flag of pulse output is ON, the instruction is no longer driven by the instruction flow.

The following figure shows a pulse output diagram.



It indicates an counterclockwise arc interpolation, wherein, S1/S2 indicates the target relative position of the x- and y-axes, such as (50,50) in the preceding figure. When the target position is the same, the example for generation of an arc less than 180 degrees and more than 180 degrees when S3/S4 is in IJ (center coordinate) mode and in R (radius) mode respectively is provided.

NOTE

- In the first segment, (I,J) = current center coordinate - starting position coordination ((150,50) - (100,50)) = (50,0).

◆ Program example

In the first arc, the IJ (center coordinate) mode is used:

In the first arc, the R (radius) mode is used:

In the second arc, the IJ (center coordinate) mode is used:

In the second arc, the R (radius) mode is used:

◆ Note

1) When an interpolation instruction (G90G01, G91G01, G90G02, G91G02, G90G02, or G91G03) is used, parameters, such as the acceleration/deceleration time, of the X-axis (Y0) prevail.

2) The user may monitor the corresponding special register for checking current pulse position, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
D8340	D8360	D8380	D8400	D8420	Current value register (PLS) (32-bit)
D8341	D8361	D8381	D8401	D8421	

You can monitor the "pulse output stop flag bit" of special elements, and view the pulse output status. This flag bit will be set during pulse output and will be automatically reset when pulse output is finished. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8340	M8360	M8380	M8400	M8420	Pulse output status

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be set uniformly or separately for each axis. The separate setting flag bit (M8350, M8370, M8390, M8410, or M8430. The default value is OFF) of special elements is used for setting and distinguishing, as shown in the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8350	M8370	M8390	M8410	M8430	[Positioning instruction] Enable the separate setting of the acceleration/ deceleration time and modification to the pulse

When the separate setting flag bit is set to OFF, the following registers are used for parameters of corresponding axes:

Y0	Y1	Y2	Y3	Y4	Attribute
D8500/D8501					Maximum speed (Hz) (32-bit)
D8502					Base speed (Hz) (16-bit)
D8503					Acceleration/deceleration time (ms) (16-bit)

When the separate setting flag bit is set to ON for an axis, the following registers are used for the parameter of the axis. For other axes of which the separate setting flag bit is not set to ON, their original registers are used.

Y0	Y1	Y2	Y3	Y4	Attribute
D8342	D8362	D8382	D8402	D8422	Maximum speed (Hz) (32-bit) [default value: 200,000]
D8343	D8363	D8383	D8403	D8423	
D8347	D8367	D8387	D8407	D8427	Base speed (Hz) [The default value is 500]
D8348	D8368	D8388	D8408	D8428	Acceleration time (ms) [The default value is 100]
D8349	D8369	D8389	D8409	D8429	Deceleration time (ms) [The default value is 100]

5) The actual minimum combined output frequency (that is, the minimum combined base output frequency S) is calculated according to the following formula:

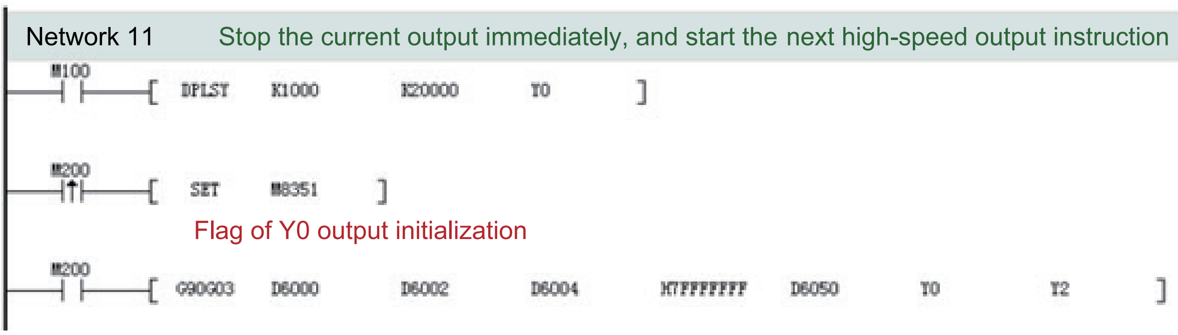
$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

6) You can enable the flag bit to release the high-speed output port resources, so that the next pulse output instruction is started immediately without disabling the previous instruction flow.

The "port output initialization flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8351	M8371	M8391	M8411	M8431	Port output initialization flag

If the flow of the pulse output instruction A is active, the instruction occupies the high-speed output port no matter whether the pulse output is finished. No pulse will be output no matter whether the flow of the pulse output instruction B using the output port is active. Because the resources of this high-speed output port have been occupied by the instruction A, an error indicating port duplication or conflict is returned. In this case, you can enable the output initialization flag bit of this port to release the port resources, and then pulses are output when the flow of the pulse output instruction B using the output port is active.



As shown in the preceding figure, M100 is active, and drives Y0 to output 20,000 pulses at a frequency of 1000 Hz. If the output is driven by M100 but the user wants to set M200 to ON to immediately start the output (SET M8351 in the preceding figure), the high-speed output driven by M100 stops immediately, the instruction driven by M200 occupies the high-speed output port Y0, and the set high-speed output starts immediately.

7) Interpolation at x- and y-axes (Y0/Y1) results in output complete interrupt of only one pulse.

8) The "interrupt enabling flag bit" of special elements must be set. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8352	M8372	M8392	M8412	M8432	Output complete interrupt enabling
Y0 output complete interrupt	Y1 output complete interrupt	Y2 output complete interrupt	Y3 output complete interrupt	Y4 output complete interrupt	Corresponding interrupt

9) The pulse output is stopped.

The pulse output can be stopped by setting the "pulse output stop flag bit" of special elements. See the following table.

Y0	Y1	Y2	Y3	Y4	Attribute
M8349	M8369	M8389	M8409	M8429	Pulse output stop flag

6.4 High-speed Processing Instruction

6

PWM: Pulse width modulation

◆ Overview

This instruction is used to output modulated square waves at the specified pulse width and during the specified pulse period.

PWM S1 S2 D			PWM	Applicable model: H3U	
S1	Output pulse width	Specified output pulse width		16-bit instruction (7 steps) PWM: Continuous execution	
S2	Pulse period	Specified pulse period			
D	Output port	High-speed pulse output port			

◆ Operands

Operand	Bit Element							Word Element														
	System·User							System·User					Bit Designation					Indexed Address		Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modification	K	H	E

Note: The elements in gray background are supported.

◆ Functions and actions

Because the relay is not applicable to high-frequency actions, this instruction is applicable only to the PLC of the transistor output type. This instruction is used to continuously output pulses at the pulse width specified by S1 through the port specified by D within the pulse period specified by S2.

S1 indicates the pulse output width. S1 must be less than or equal to S2. The range is 0 to 32,767 ms.

S2 indicates the pulse output period. S1 must be less than or equal to S2. The range is 1 to 32,767 ms.

D indicates the pulse output port which cannot be occupied by other high-speed instructions.

This instruction is executed by interruption. When the instruction flow is OFF, the output stops immediately.

S1 and S2 can be modified when the PWM instruction is executed.

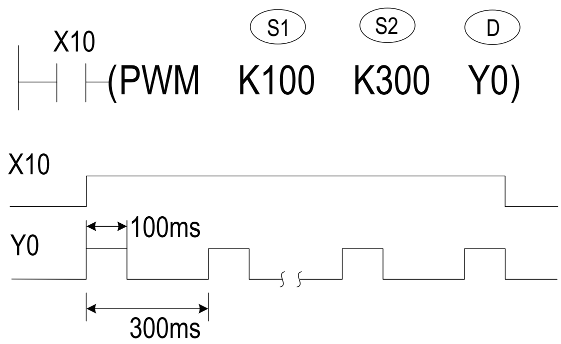
◆ Note

The current pulse position cannot be modified when the pulses are output through the PWM instruction.

S1 and S2 can be modified when the PWM instruction is executed.

The PWM instruction is not supported by the H3U-PM model.

◆ Program example



6

Memo NO. ____________

Date / /

7 Motion Control

Chapter 7 Motion Control

This chapter describes the motion control instructions, G-code, electronic cam, and other functions supported by the PLC of the H3U-PM motion control model.

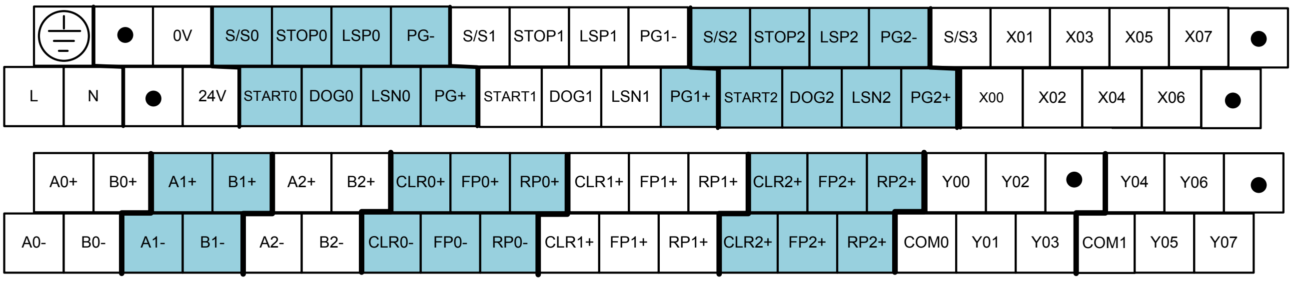
7.1 Difference Between H3U Standard Model and H3U-PM Motion Control Model

The following table lists the main differences between PLCs of the standard and motion control models.

Item	H3U Standard Model	PM Motion Control Model
Number of high-speed inputs	8	3
High-speed input type	Collector input	Differential input
High-speed input mode	Pulse	Pulse+Direction CW/CCW Phase A/B
Maximum high-speed input frequency	200 kHz	200 kHz
Number of high-speed outputs	5	3
High-speed output type	Collector output	Differential output
High-speed output mode	Pulse+Direction	Pulse+Direction CW/CCW Phase A/B
Maximum high-speed output frequency	200 kHz	500 kHz
Type of subprogram supported	General subprogram Interrupt subprogram Encrypted subprogram Parameter-carrying subprogram Encrypted parameter-carrying subprogram	General subprogram Interrupt subprogram Encrypted subprogram Parameter-carrying subprogram Encrypted parameter-carrying subprogram Motion control subprogram (MC subprogram and G-code subprogram)
2-axis linear and arc interpolation	Supported	Supported
3-axis linear interpolation	Not supported	Supported
3-axis helix interpolation	Not supported	Supported
Arc interpolation plane	XY plane	XY plane YZ plane ZX plane
Motion control instruction	Not supported	Supported
G-code	Not supported	Supported
Dedicated input and output signal	None	Yes
Electronic cam	Not supported	Three channels supported

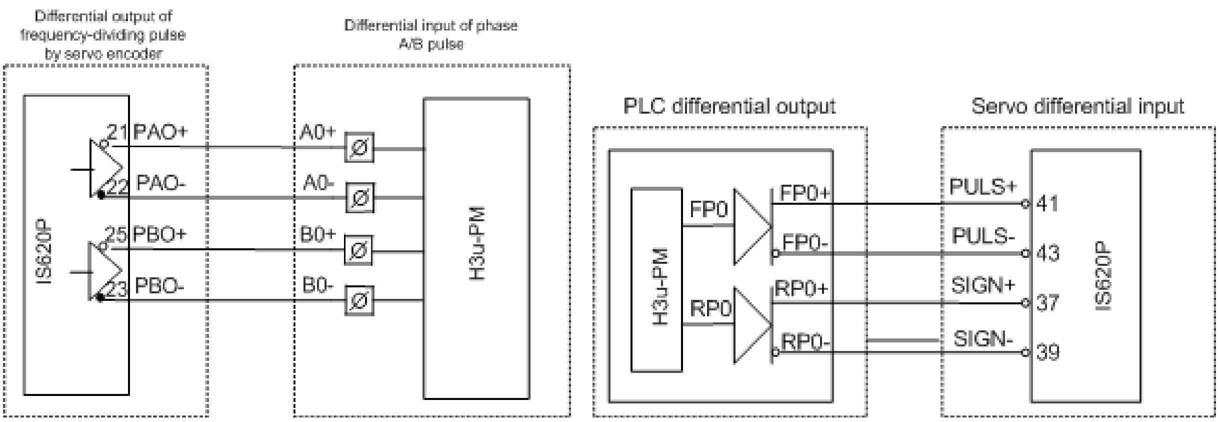
7

7.2 H3U-PM Terminal Description



Terminal	Description
START	Input signal start
STOP	Input signal stop
LSP/LSN	Right limit/left limit
DOG	Near point signal for zero return running
PG+/PG-	Origin signal/external input interrupt, pulse capture
X0 to X7	General input point
A+/A-	A-phase pulse input (differential signal)
B+/B-	B-phase pulse input (differential signal)
FP+/FP-	CW/CCW mode: CW pulse output (differential signal) Pulse/direction: pulse output port A/B-phase mode: A-phase output
RP+/RP-	CW/CCW mode: CCW pulse output (differential signal) Pulse/direction: Direction output port A/B-phase mode: B-phase output
CLR+/CLR-	Clearing signal (clearing signal for the internal deviation counter of the servo driver) or cam synchronous output signal
Y0 to Y3	Ordinary transistor output
Y5 to Y7	Relay output

- H3U-PM and IS620P wiring example:

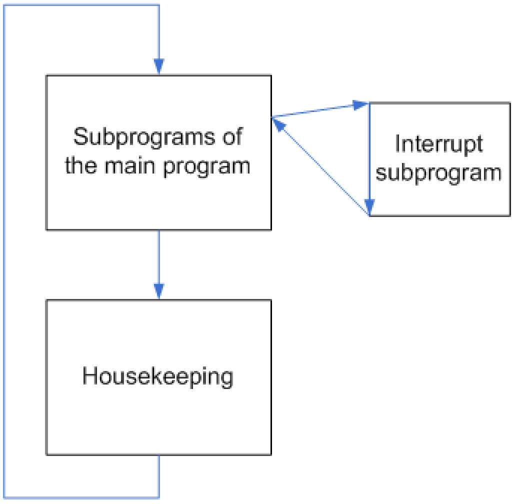


7.3 Execution and Call of Motion Control Subprogram

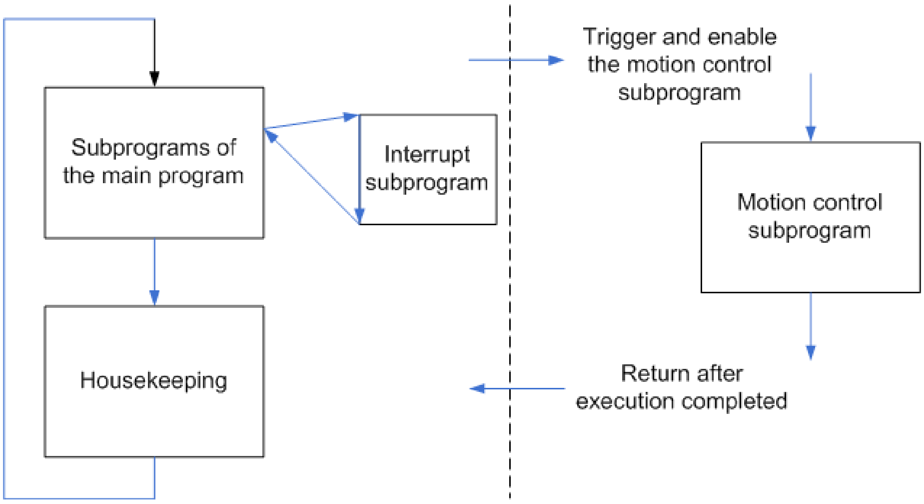
1) Execution of motion control subprograms (MC subprograms and G-code subprograms)

The motion control subprograms supported by the H3U-PM Series PLC are classified into two types: MC and G-code. G-code subprograms are a kind of MC subprograms.

The following figure shows the execution logic and cyclic scan mode of the main program and subprograms.



The motion control main program and subprograms are executed by single trigger in parallel, as shown in the following figure.



- In this case, a "main process" and a "motion process" are generated in the PLC. The main process starts the motion process, and the two processes run in parallel and exchange data through value assignment to elements.
- Both the main program and its subprograms are in the main process, and instructions, including the positioning instructions, can be used. Programs in the motion process can be started, but only one program can be started at a time; otherwise, an error is returned.
- The motion control subprograms (MC subprograms and G-code subprograms) are in the motion process. Subprograms can be called through M98 in the G-code in the motion process.

2) Calling a motion control subprogram (an MC subprogram or a G-code subprogram)

General subprograms, encrypted subprograms, and parameter-carrying subprograms in H3U-PM are the same, but have different attributes. Like the main program, motion control subprograms can be called. The following table shows the call relationship between subprograms.

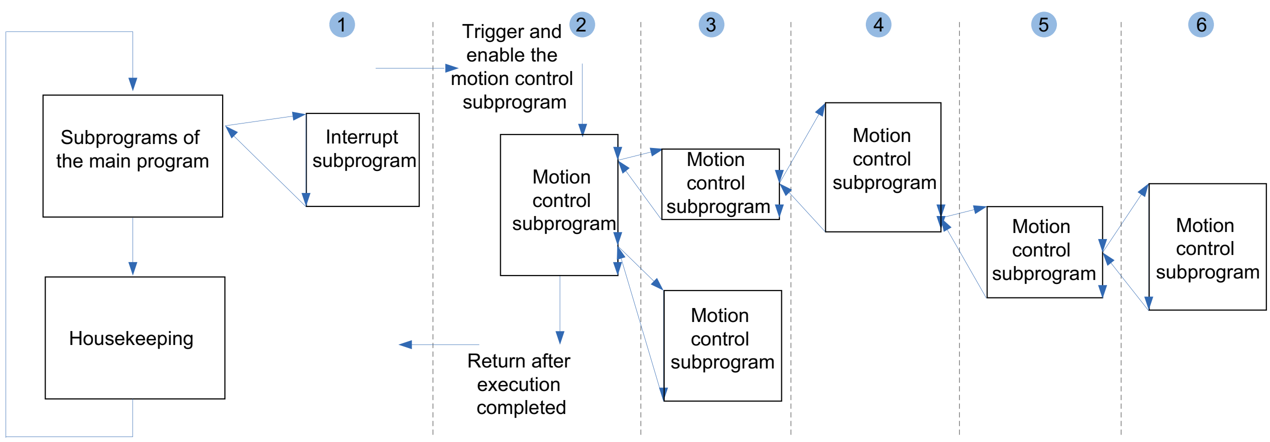
Callee / Caller	General subprogram Encrypted subprogram Parameter-carrying subprogram	MC subprogram	G-code subprogram
Main program (MAIN)	Called by the CALL instruction	(Set by SD/SM)	(Set by SD/SM)
General subprogram Encrypted subprogram Parameter-carrying subprogram	Called by the CALL instruction	(Set by SD/SM)	(Set by SD/SM)
MC subprogram	×	(Called by the MCALL instruction)	(Called by the MCALL instruction)
G-code subprogram	×	×	(Called by M98)

Note: In the example where MAIN calls an MC subprogram, MAIN is the caller, and the MC subprogram is the callee.

Up to 64 MC subprograms are supported and numbered from MC00 to MC63. Only one G-code subprogram file numbered CNC00 (corresponding to MC10000) is supported. The G-code subprogram file may have multiple Oxxxxs numbered from O0000 to O9999, which can be called as subprograms.

a) Nested layers of motion control subprograms

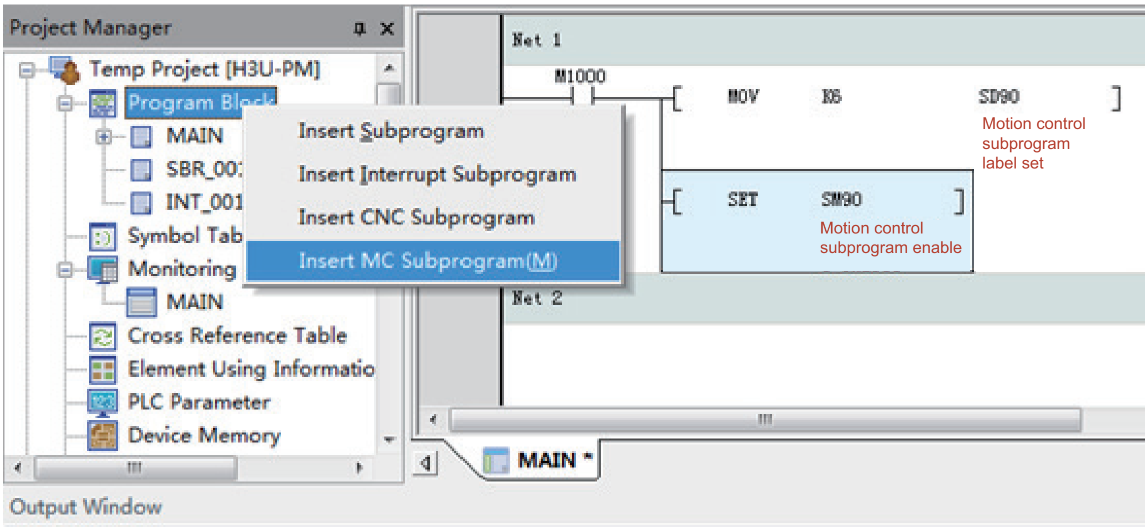
Up to six nested layers of motion control subprograms are supported. The first layer is the motion control subprogram called by a main program or a subprogram. Each time the motion control subprogram is called, the number of nested layers is increased by 1. If the call of the nested layer is returned, the number of nested layers is not increased, as shown in the following figure.



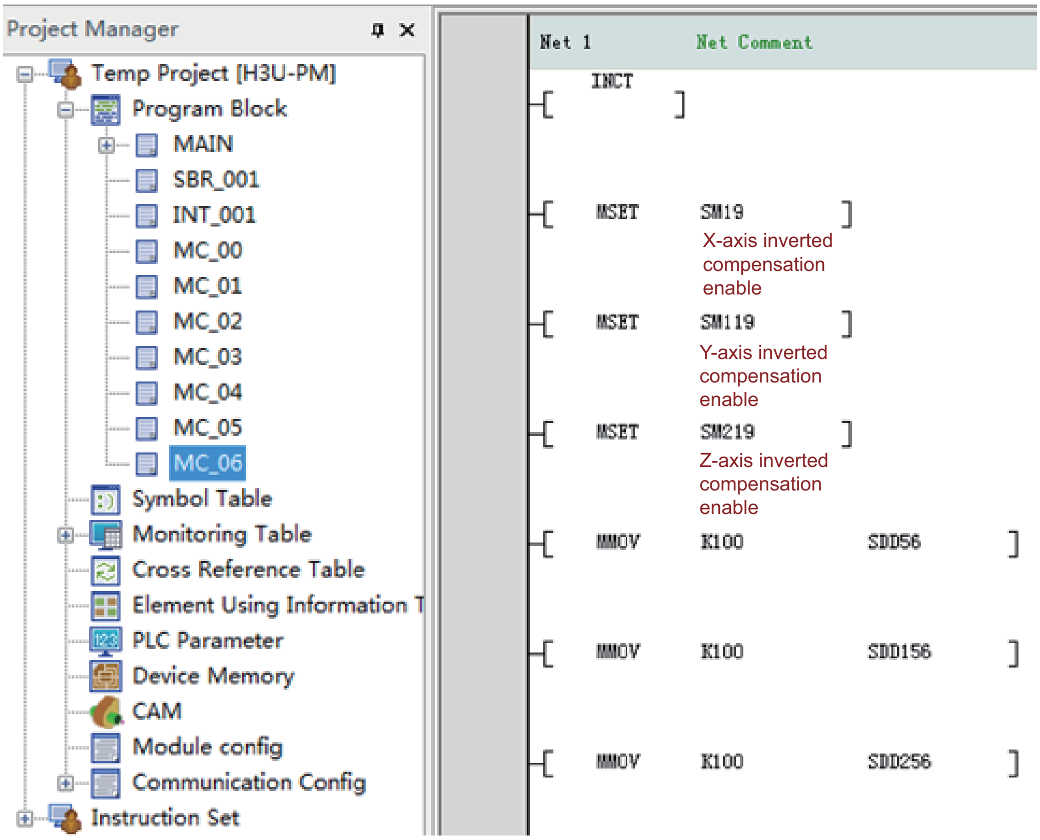
7

b) Creating and calling an MC subprogram

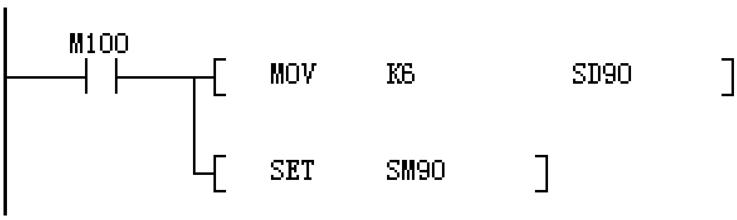
Creating an MC subprogram: Right-click Program Block and choose Insert MC Subprogram (M).



After an MC subprogram is created, run an MC instruction to program the MC subprogram. The MC instruction connects the bus.

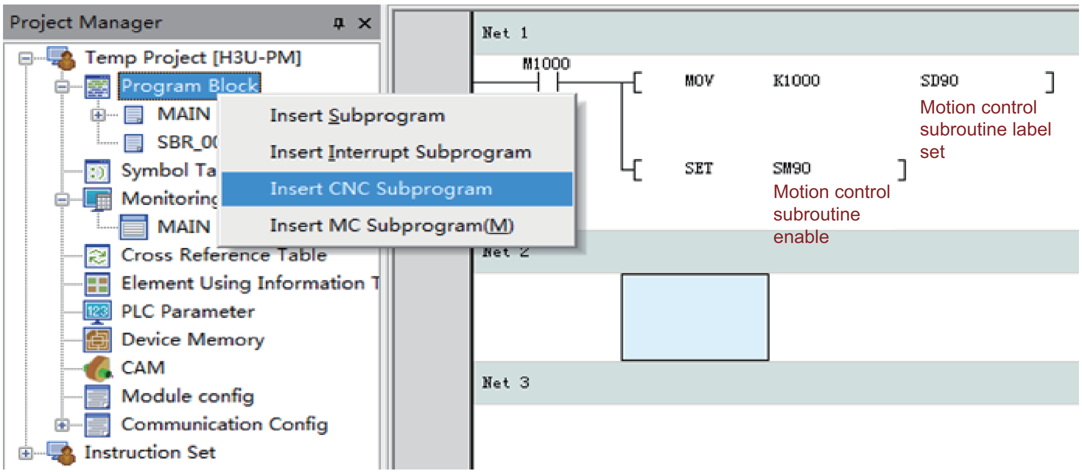


Calling an MC subprogram: The MC subprogram can be called in the main program or a subprogram by setting the MC number in SD90 and setting SM90 to ON.

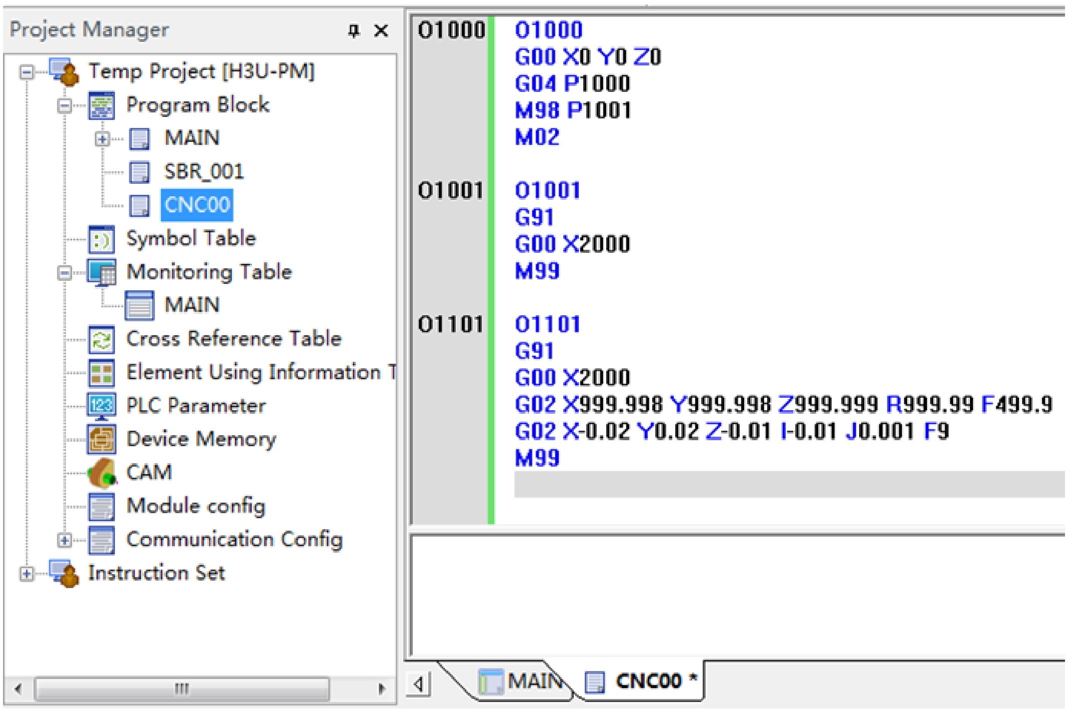


c) Creating a G-code subprogram

Create a G-code subprogram to enable the main program to call CNC00 (MC10000), as shown in the following figure.



The CNC subprogram is a G-code subprogram input in text. If G-code is generated using a third-party tool, the Import function on the menu can be used to import the G-code. The content of the created CNC subprogram is shown in the following figure.



The following table shows the relationship between the main program, subprograms, and instructions.

H3U-PM Supported Instruction	General Instruction for Logic and Process Control	High-speed Counting and Input Instruction	High-speed Output	Motion Control Instruction	SFC	Quantity and Flag
Main program (MAIN)	√	√	√	×	√	1
General subprogram	√	√	√	×	×	512
Encrypted subprogram	√	√	√	×	×	512
Parameter-carrying subprogram	√	√	√	×	×	512
MC subprogram	×	×	×	√	×	64
G-code subprogram	×	×	×	√	×	10 (to be extended)

Up to 64 MC subprograms are supported and numbered from MC00 to MC63. Only one G-code subprogram file numbered CNC00 (corresponding to MC10000) is supported. The G-code subprogram file may have multiple Oxxxxs numbered from O0000 to O9999, which can be called as subprograms.

7.4 Motion Control Instructions

1) Table of attributes of motion control instructions

The following table lists attributes of major motion control instructions.

Instruction	No Acceleration/ Deceleration	Trapezoid Acceleration/ Deceleration	S-curve Acceleration/ Deceleration	Separate Acceleration/ Deceleration Setting	Direction Change During Running	Speed OR Position Control
DRV	√	√	√ (SM)	√		Position
LIN	√	√		√		Position
INTR	√	√		√		Position
CW	√	√		√	√	Position
CCW	√	√		√	√	Position
DRVR	√	√	√ (SM)	√		Position
DRVZ	√	√		√	√	Speed+position
SINTR	√	√		√		Speed+position
DINTR	√	√		√		Speed+position
				√		
G00	√	√	√ (SM)	√		Position
G01	√	√		√		Position
G02	√	√		√	√	Position
G03	√	√		√	√	Position

NOTE

- Whether the H3U-PM motion control instructions support acceleration/deceleration is determined by the acceleration/deceleration time. Some instructions can run in both acceleration/deceleration and non-acceleration/deceleration modes. The acceleration/deceleration time of positioning and interpolation instructions ranges from 10 ms to 5000 ms and 10 ms to 500 ms respectively. If the acceleration/deceleration time is 0, the instruction does not support acceleration/deceleration; otherwise, the instruction supports acceleration/deceleration. The upper/lower limit will be used when the value is out of range.
- Current running of a motion control subprogram means that the subprogram is always in running state after the SM90 enabling flag is active. The current running is finished when the SM91 complete flag switches to ON. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope. The modal instruction enabled in the current running remains active until the execution is completed or changed. A modal instruction remains active after it is specified in a program segment until another instruction in the same group is specified or it is canceled by another instruction. A non-modal instruction is active only within the program segment in which it is specified.
- After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.

Besides, some H3U general positioning instructions are supported. These instructions can be called only in the main program and subprograms, but cannot be called in the motion control subprograms (MC subprograms and G-code subprograms), as shown in the following table.

Instruction	Pulse Direction Output	Trapezoid Acceleration/ Deceleration	S-curve Acceleration/ Deceleration	Separate Acceleration/ Deceleration Setting	Frequency Modification Supported During Running	Pulse Count Modification Supported During Running	Direction Change During Running	Speed OR Position Control
PLSY					√	√ (M)		Speed Position Speed+ position
PLSV	√				√		√	Speed
PLSV2	√	√		√ (M)	√		√	Speed
ZRN		√		√ (M)				Speed
PLSR		√	√ (M)	√ (M)		√ (M)		Position
DRVA	√	√	√ (M)	√ (M)		√ (M)		Position
DRVI	√	√	√ (M)	√ (M)		√ (M)		Position
PLSN	√	√		√ (M)				Position

7

2) Use of motion control instructions

The following table lists modals supported by the instructions.

Instruction	Running Plane		Position Mode		Default Interpolation Speed	
	XYP/ZXP/YZP	G17/G18/G19	ABST/INCT	G90/G91	Maximum speed	Inherited from the F function word
DRV	√		√		√	
LIN	√		√			√
INTR	√		√			√
CW	√		√			√
CCW	√		√			√
DRVZ	√		√		√	
DRVR	√		√		√	
SINTR	√		√		√	
DINTR	√		√		√	
Supported through calling a G-code subprogram by using an MC subprogram						
G00	√ (Followed)	√	√ (Followed)	√	√	
G01	√ (Followed)	√	√ (Followed)	√		√
G02	√ (Followed)	√	√ (Followed)	√		√
G03	√ (Followed)	√	√ (Followed)	√		√

NOTE

- Followed: The modal is not supported by the G-code, but the modal of the motion control subprogram calling the G-code is followed.
- The modal instruction in a motion control subprogram remains active until the current running is complete (SM91 = ON) or changed. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope.
- After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.

7.5 Similarities and Differences Between MC Subprograms and G-code Subprograms

Both MC subprograms and G-code subprograms are motion control subprograms used to control the motion path.

Their differences include:

Parameters of an MC subprogram can be elements or immediate operands, and thus can be more flexibly set using the ladder chart. For example, to run a subprogram for arc interpolation, set the radius, center, and the target position as parameters, and use HMI to configure and release the formula to drive the actuator to move in any circular path.

G-code is edited in text, and its parameters are generally immediate operands. G-code automatically generated by third-party software can be imported to control the path, so as to reduce the editing workload.

Their similarities include:

Both of MC subprograms and G-code subprograms use the special elements SM0 to SM299 and SD0 to SD299 for motion control (interpolation instruction and electronic cam) setting.

The following table lists same functions of the two.

Instruction	No Acceleration/ Deceleration	Trapezoid Acceleration/ Deceleration	S-curve Acceleration/ Deceleration	Separate Acceleration/ Deceleration Setting	Direction Change During Running	Speed OR Position Control
DRV/G00	√	√	√ (SM)	√		Position
LIN/G01	√	√		√		Position
CW/G02	√	√		√	√	Position
CCW/G03	√	√		√	√	Position

- SM indicates that the function can be used only after the SM special element is set.
- Whether the H3U-PM motion control instructions support acceleration/deceleration is determined by the acceleration/deceleration time. Some instructions can run in both acceleration/deceleration and non-acceleration/deceleration mode. The acceleration/deceleration time of positioning and interpolation instructions ranges from 10 ms to 5000 ms and 10 ms to 500 ms respectively. If the acceleration/deceleration time is 0, the instruction does not support acceleration/deceleration; otherwise, the instruction supports acceleration/deceleration. The upper/lower limit will be used when the value is out of range.
- Current running of a motion control subprogram means that the subprogram is always in running state after the SM90 enabling flag is active. The current running is finished when the SM91 complete flag switches to ON. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope. The modal instruction enabled in the current running remains active until the execution is completed or changed.
- After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.
- The acceleration/deceleration time of G01 and LIN instructions can be set separately (M), in the range 10 ms to 500 ms. Interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. Parameters of the x-axis prevail in case of x-y-z-axes interpolation, x-y-axes interpolation, and x-z- axes interpolation; and parameters of the y-axis prevail in case of y-z-axes interpolation.
- The acceleration/deceleration time of CW, G02, and G03 can be set separately (M), in the range 10 ms to 500 ms. More than 20 pulses shall be output along the arc during arc interpolation; otherwise, an error is returned. The number of pulses output at the third axis shall be no more than 0.9 times that to be output along the arc during helix interpolation; otherwise, an error is returned. Up to 8,000,000 pulses can be output along the radius during arc interpolation; when converted according to the default ratio, the radius is 4000 mm. Interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. For example, parameters of the x-axis prevail on the XY plane; parameters of the y-axis prevail on the YZ plane; and parameters of the z-axis prevail on the ZX plane.

7.6 List of Motion Control Instructions Supported by MC Subprograms

The following table lists instructions supported by MC subprograms.

Motion Control Instructions	"DRV: High-speed positioning"
	"LIN and INTR: Linear interpolation"
	"CW: Clockwise arc interpolation; CCW: Counterclockwise arc interpolation"
	"TIM: Delay waiting"
	"MCALL: Motion control subprogram calling"
	"MRET: Motion control subprogram return"
	"ABST: Absolute position modal; INCT: Relative position modal"
	"XYP, YZP, and ZXP: Setting a modal for the current plane"
	"SETT: Setting the current position"
	"SETR: Setting the electrical origin"
	"DRVZ: Mechanical zero return"
	"DRVR: Electrical zero return"
	"SINTR: Single-speed interrupt positioning; DINTR: Double-speed interrupt positioning"
	"MOVC: Linear displacement compensation"
	"CNTC: Arc center compensation"
	"RADC: Arc radius compensation"
	"CANC: Motion compensation cancellation"
Other Instructions	"MSET and MRST: Setting and resetting the bit element M"
	"MMOV: Value assignment; MADD, MSUB, MMUL, MDIV: Addition, subtraction, multiplication, and division"

7

7.7 Format and Use of MC Subprograms

LIN X DD 100

1 2 3 4

① Motion control instructions

② Type of the function word. The combination of ②, ③, and ④ indicates one parameter which can be omitted.

③ Type of the element, which can be omitted to indicate a floating-point number. Omission using a specific instruction indicates an integer.

④ Serial number of a element or value of an immediate operand.

Motion control subprograms can be used to perform different actions, but cannot be used to control logic contacts. In MC subprograms, motion control instructions can be entered directly by using the bus. Only one instruction can be entered in each row, and is directly connected to the bus.

The following table lists the types of elements of operands used for the above instructions.

Element Supported	Floating-point Number	Integer	HEX	D	SD	R	M	SM
Bit element	-	-	-	-	-	-	Mxxxx	SMxxxx
16-bit parameter	-	Kxxxx	Hxxxx	Dxxxx	SDxxxx	Rxxxxx	-	-
32-bit parameter	-	KKxxxx	HHxxxx	DDxxxx	SDDxxxx	RRxxxxx	-	-
Floating-point number	xxxx.xx	-	-	DExxxx	SDExxxx	RExxxxx	-	-
Displayed value	Actual value	Actual value	Actual value	Element number	Element number	Element number	Element number	Element number

For example, D100 is used for a 16-bit integer, DD100 is used for a 32-bit integer, and DE100 is used for a 32-bit floating-point number. This is also the same for other word elements.

Taking the immediate operand 100 for example, if the 16-bit integer of the immediate operand is in decimal, K100 is used; if the 32-bit integer is in decimal, KK100 is used. If the 16-bit integer of the immediate operand is in hexadecimal, H64 is used; if the 32-bit integer is in hexadecimal, HH64 is used. Floating-point numbers support 32-bit single precision; that is, 100 can be entered directly to indicate 100.000.

Motion control instructions support 16-bit and 32-bit hybrid programming. The following table lists function words supported.

	Motion Control Instruction	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Attribute
High-speed positioning	DRV	X	F	Y	F	Z	F		Combination
Linear interpolation	LIN	X	Y	Z	F				Combination
Clockwise arc interpolation	CW	X	Y	Z	I	J	K	F	Combination
		X	Y	Z	R	Default	Default	F	Combination
Counter-clockwise arc interpolation	CCW	X	Y	Z	I	J	K	F	Combination
		X	Y	Z	R	Default	Default	F	Combination
Delay waiting	TIM	Integer							
Absolute position	ABST								
Relative position	INCT								
XY plane	XYP								
ZX plane	ZXP								
YZ plane	YZP								
Mechanical zero return reset	DRVZ								
Electrical zero return	DRVR								
Electrical zero setting	SETR								
Current position setting	SETT	X	Y	Z					Combination
Linear interpolation	INTR	X	Y	Z	F				Combination
Single-speed interrupt positioning	SINTR	X/Y/Z	F						
Double-speed interrupt positioning	DINTR	X/Y/Z	F	F					
Linear displacement compensation	MOVC	X	Y	Z					Combination
Arc center compensation	CNTC	I	J	K					
Arc radius compensation	RADC	R							

	Motion Control Instruction	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Attribute
Motion compensation cancellation	CANC								
Subprogram call	MCALL	P integer							
Subprogram return	MRET								

X, Y, and Z indicate the positions of the x-axis, y-axis, and z-axis; R indicates the radius; F indicates the feed speed; I, J, and K indicate the center relative to the current position. For example, "LIN XKK1000 YHH2000 FKK50000" indicates linear interpolation at a speed of 50,000 to move to the X position 1000 and Y position 0x2000 in hexadecimal.

P indicates the subprogram. For example, "MCALL P5" means calling the fifth motion control subprogram.

In a motion control subprogram, the integer type uses the pulse unit. The floating-point number type uses the mechanical unit. It is necessary to set relevant special registers and the pulse-to-mechanical ratio within the maximum speed or stroke range; otherwise, an error is returned.

Towards the positioning or trajectory control, it supports the setting of acceleration/deceleration type/time, maximum speed, base speed, and they must be set in the special elements..

Parameters for an axis not used during multi-axes action need not be entered. In such a way, the axis resources are not occupied, and the axis can be used for positioning or electronic cam running.

The following table lists other additional instructions.

Function	Application Instruction	Instruction Format	Parameter Count
Setting	MSET	MSET S1	1
Reset	MRST	MRST S1	1
Assignment	MMOV	MMOV S1 D1	2
Addition (integer/ floating point)	MADD	MADD S1 S2 D1	3
Subtraction (integer/ floating point)	MSUB	MSUB S1 S2 D1	3
Multiplication (integer/floating point)	MMUL	MMUL S1 S2 D1	3
Division (integer/ floating point)	MDIV	MDIV S1 S2 D1	3

NOTE

- MSET and MRST support only the M and SM bit elements. Other application instructions support only the word elements, but do not support bit elements in KnX form or address indexing.
- These instructions can only be used in MC subprograms.

DRV: High-speed positioning

◆ Overview

The DRV instruction is used to drive three axes to move to the target position or output the specified number of pulses at the specified output frequency of each axis.

DRV X_ F_ Y_ F_ Z_ F_		High-speed positioning	Applicable model: H3U-PM
X	X-axis position	X-axis target position	
F	X-axis speed	X-axis output frequency	
Y	Y-axis position	Y-axis target position	
F	Y-axis speed	Y-axis output frequency	
Z	Z-axis position	Z-axis target position	
F	Z-axis speed	Z-axis output frequency	

◆ Operands

Parameter	Bit Element		Word Element									Immediate operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

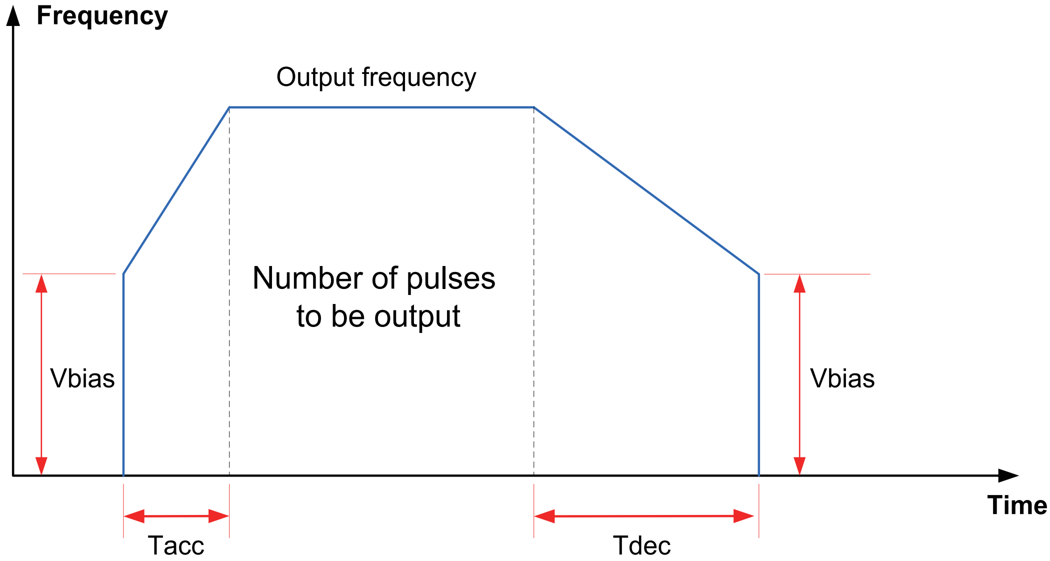
The DRV instruction is used to drive the x-, y-, and z-axes to move to the target position or output the specified number of pulses at the specified output frequency of each axis. The axis for which the F function word is omitted runs at the maximum speed.

Both the absolute position and relative position modes are supported. Both the absolute position and relative position are relative to the current position.

The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

The following is a single axis pulse output diagram.



◆ Note

1) Multiple instruction forms are supported. The axis for which the F function word is omitted runs at the maximum speed.

Instruction Form	Description
DRV X_ F_ Y_ F_ Z_ F_	Standard format
DRV X_	Single-axis
DRV X_ F_	
DRV Y_	
DRV Y_ F_	
DRV Z_	
DRV Z_ F_	
DRV X_ Y_	2-axis
DRV X_ F_ Y_	
DRV X_ Y_ F_	
DRV X_ F_ Y_ F_	
DRV Y_ Z_	
DRV Y_ F_ Z_	
DRV Y_ Z_ F_	
DRV Y_ F_ Z_ F_	
DRV X_ Z_	
DRV X_ F_ Z_	
DRV X_ Z_ F_	
DRV X_ F_ Z_ F_	

Instruction Form	Description
DRV X_ Y_ Z_	3-axis
DRV X_ F_ Y_ Z_	
DRV X_ Y_ F_ Z_	
DRV X_ Y_ Z_ F_	
DRV X_ F_ Y_ F_ Z_	
DRV X_ Y_ F_ Z_ F_	
DRV X_ F_ Y_ Z_ F_	

2) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

3) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6, SD7	SD106, SD107	SD206, SD207	Number of pulses required when the motor rotates a circle (A)
SD8, SD9	SD108, SD109	SD208, SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

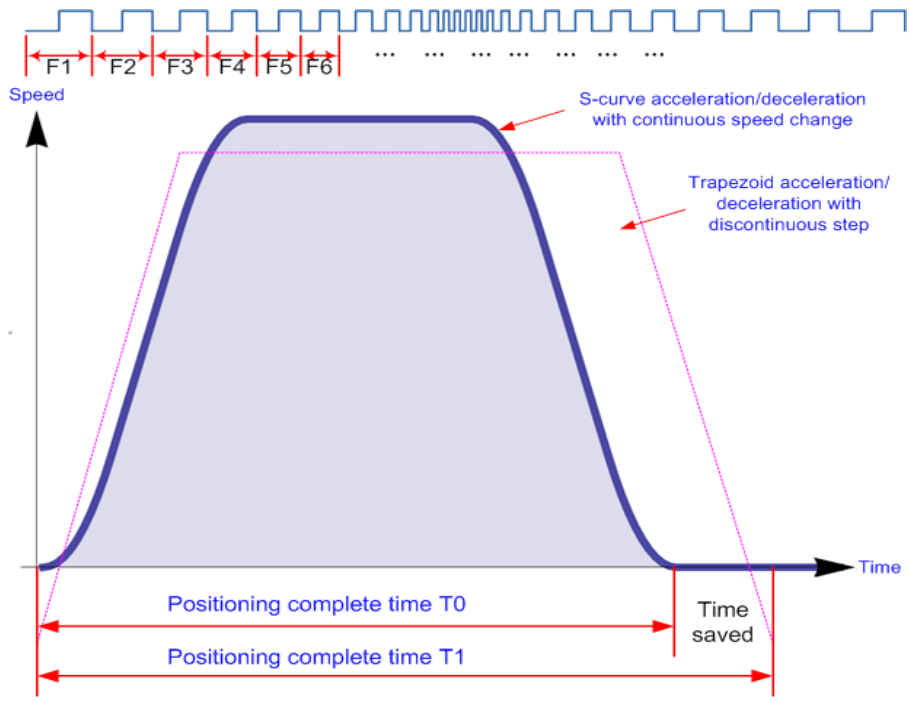
F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

4) Trapezoid acceleration/deceleration and S-curve acceleration/deceleration are supported.

The two acceleration/deceleration modes can be distinguished by setting the "S-curve acceleration/deceleration enabling flag" of special elements. If the flag is not set, the trapezoid acceleration/deceleration mode is used by default. The following table lists details about S-curve acceleration/deceleration.

X-axis	Y-axis	Z-axis	Attribute	Remarks
SM17	SM117	SM217	S-curve acceleration enabling flag	ON indicates an S-curve, and OFF indicates a trapezoid curve.

DRV, G00, and DRVR support S-curve acceleration/deceleration. Therefore, at given mechanical stability, the target speed is increased, the positioning time is shortened, and the processing efficiency is improved.



The advanced pulse-by-pulse modulation algorithm is used for S-curve acceleration/deceleration. The frequency of each pulse is adjusted to ensure more smooth positioning.

5) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms. The time of trapezoid acceleration/deceleration and S-curve acceleration/deceleration can be set separately.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

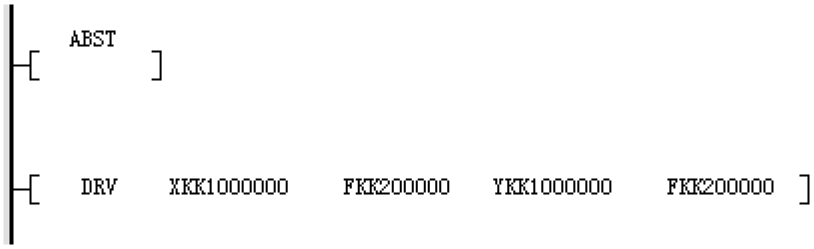
X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12, SD13	SD112, SD113	SD212, SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

7

6) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

◆ Program example



It indicates that, in absolute position mode, 10,000,000 pules are output at the x-axis at a frequency of 200,000 Hz, and 10,000,000 pules are output at the y-axis at a frequency of 200,000 Hz.

LIN and INTR: Linear interpolation

◆ Overview

These instructions are used to perform linear interpolation at up to three axes at the combined output frequency.

LIN X_ Y_ Z_ F_		Linear interpolation	Applicable model: H3U-PM	
X	X-axis position	X-axis target position		
Y	Y-axis position	Y-axis target position		
Z	Z-axis position	Z-axis target position		
F	Combined interpolation speed	Combined interpolation output frequency		

INTR X_ Y_ Z_ F_		Linear interpolation	Applicable model: H3U-PM	
X	X-axis position	X-axis target position		
Y	Y-axis position	Y-axis target position		
Z	Z-axis position	Z-axis target position		
F	Combined interpolation speed	Combined interpolation output frequency		

◆ Operands

Parameter	Bit Element		Word Element									Immediate operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

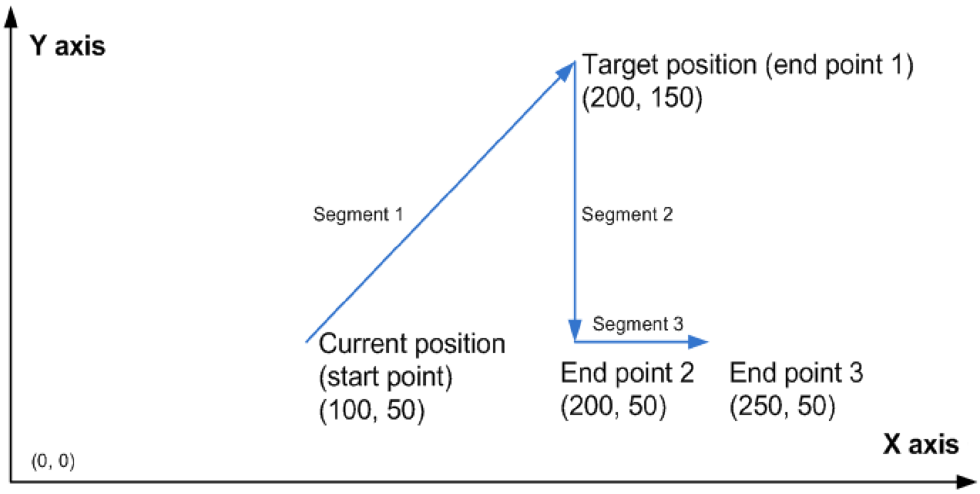
Linear interpolation can be performed at up to three axes. The axis for which the F function word is omitted inherits the running speed of the previous interpolation instruction.

Both the absolute position and relative position modes are supported. Both the absolute position and relative position are relative to the current position.

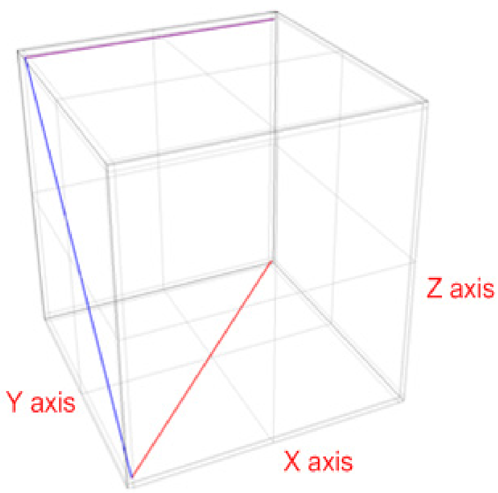
The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

The following is a pulse output diagram.



3D linear interpolation of any type is supported.



◆ Note:

1) Interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. Parameters of the x-axis prevail in case of x-y-z-axes interpolation, x-y-axes interpolation, and x-z-axes interpolation; and parameters of the y-axis prevail in case of y-z-axes interpolation.

2) Multiple instruction forms are supported. The axis for which the F function word is omitted inherits the running speed of the previous interpolation instruction.

7

Instruction Form	Description
LIN X_ Y_ Z_ F_	Standard format
LIN X_	Single-axis
LIN X_ F_	
LIN Y_	
LIN Y_ F_	
LIN Z_	
LIN Z_ F_	
LIN X_ Y_	2-axis
LIN X_ Y_ F_	
LIN Y_ Z_	
LIN Y_ Z_ F_	
LIN X_ Z_	
LIN X_ Z_ F_	
LIN X_ Y_ Z_	3-axis

INTR functions in the same way as described above.

3) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

7

4) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6, SD7	SD106, SD107	SD206, SD207	Number of pulses required when the motor rotates a circle (A)
SD8, SD9	SD108, SD109	SD208, SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

5) Only trapezoid acceleration/deceleration is supported.

6) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12, SD13	SD112, SD113	SD212, SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

7) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

7

◆ Program example

It indicates that in the relative position mode, linear interpolation is performed for the x- and y-axes relative to the current position at a frequency of 30,000 HZ. 10,000 pulses are output on the x-axis, and 20,000 pulses are output on the y-axis.

The other example indicates that in the relative position mode, linear interpolation is performed for the x-, y-, and z-axes relative to the current position at a speed of 300 mm/min. The distances are 100 mm, 200 mm, and 300 mm respectively.

CW: Clockwise arc interpolation; CCW: Counterclockwise arc interpolation

◆ Overview

These instructions are used to perform clockwise or counterclockwise arc interpolation at combined output frequency at two axes. 3-axis helix interpolation on three planes is supported.

CW X_ Y_ Z_ I_ J_ K_ F_ CW X_ Y_ Z_ R_ F_		Clockwise arc interpolation, helix interpolation	Applicable model: H3U-PM	
X	X-axis position	X-axis target position		
Y	Y-axis position	Y-axis target position		
Z	Z-axis position	Z-axis target position		
I	X-axis center	X-axis center, used in the IJK (center) mode		
J	Y-axis center	Y-axis center, used in the IJK (center) mode		
K	Z-axis center	Z-axis center, used in the IJK (center) mode		
R	Arc radius	Arc radius, used in the R (radius) mode		
F	Combined interpolation speed	Combined interpolation output frequency		

CCW X_ Y_ Z_ I_ J_ K_ F_ CCW X_ Y_ Z_ R_ F_		Counterclockwise arc interpolation, helix interpolation	Applicable model: H3U-PM	
X	X-axis position	X-axis target position		
Y	Y-axis position	Y-axis target position		
Z	Z-axis position	Z-axis target position		
I	X-axis center	X-axis center, used in the IJK (center) mode		
J	Y-axis center	Y-axis center, used in the IJK (center) mode		
K	Z-axis center	Z-axis center, used in the IJK (center) mode		
R	Arc radius	Arc radius, used in the R (radius) mode		
F	Combined interpolation speed	Combined interpolation output frequency		

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
I	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
J	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
K	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
R	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

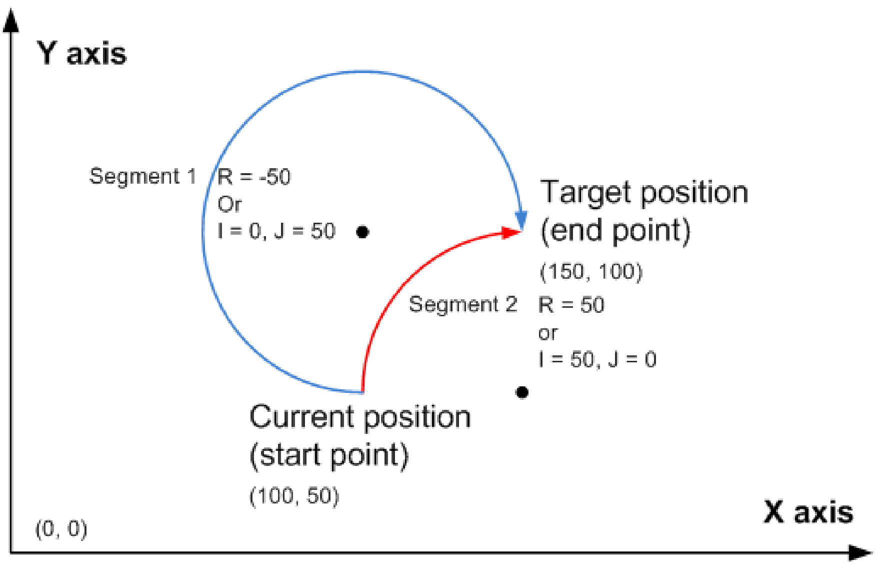
These instructions are used to perform 2-axis arc interpolation or 3-axis helix interpolation on three planes. The axis for which the F function word is omitted inherits the running speed of the previous interpolation instruction.

Both the absolute position and relative position modes are supported. Both the absolute position and relative position are relative to the current position.

The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

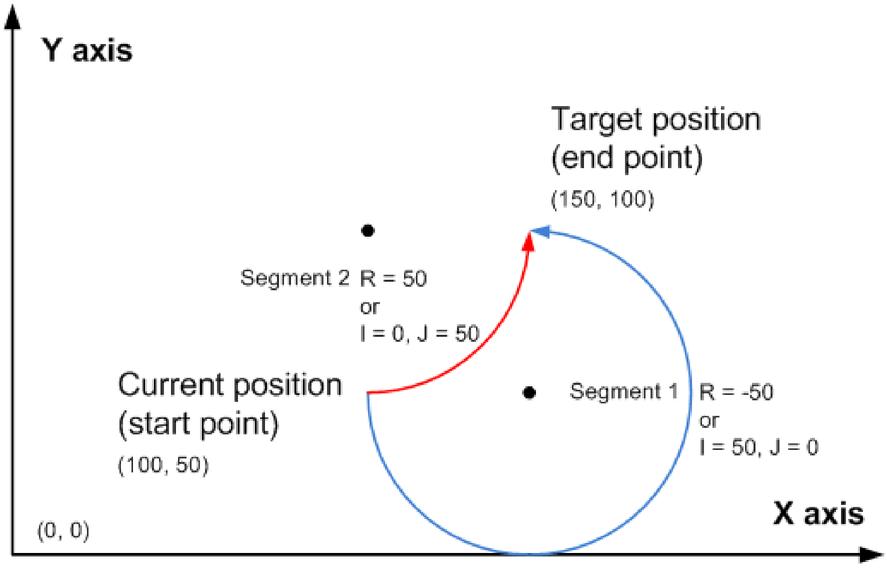
X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

The following is a diagram of clockwise arc interpolation.



It indicates a clockwise arc interpolation, wherein (150, 100) indicates the target absolute position of the x- and y-axes. When the target position is the same, an example is provided on generation of an arc less than 180 degrees and more than 180 degrees in IJ (center coordinate) mode and in R (radius) mode respectively.

The following is a diagram of counterclockwise arc interpolation.



It indicates an counterclockwise arc interpolation, wherein (150, 100) indicates the target absolute position of the x- and y-axes. When the target position is the same, an example is provided on generation of an arc less than 180 degrees and more than 180 degrees in IJ (center coordinate) mode and in R (radius) mode respectively.

1) The user needs to set an appropriate target position so that the correct target circular path can be generated. In absolute position mode, when the specified target position of the axis at which the arc interpolation is performed equals its current position, a complete circle is generated. In relative position mode, when the specified target position of the axis at which the arc interpolation is performed is 0, a complete circle is generated.

2) Either the IJ (center coordinate) or R (radius) mode is supported.

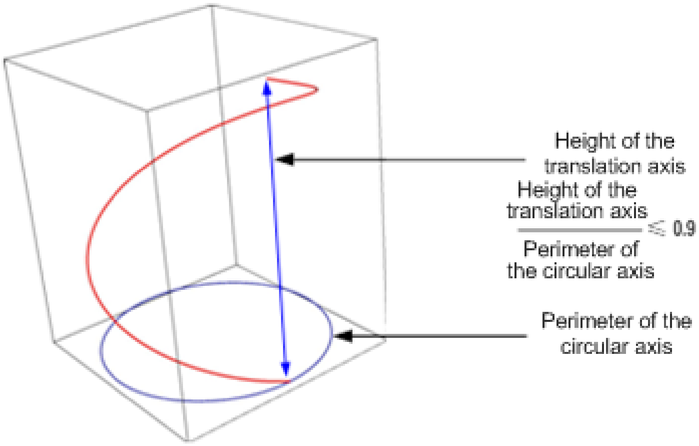
3) In IJ (center coordinate) mode, no matter it is absolute position interpolation or relative position interpolation, I, J, or K only indicates the difference (offset) of the central coordinate relative to the current position on the x-, y-, and z-axes.

4) In R (radius) mode, when the R value is larger than or equal to 0, it indicates an arc less than or equal to 180 degrees. When the R value is smaller than 0, it indicates an arc more than 180 degrees. In R (radius) mode, no complete circle can be generated.

5) More than 20 pulses shall be output along the arc during arc interpolation; otherwise, an error is returned.

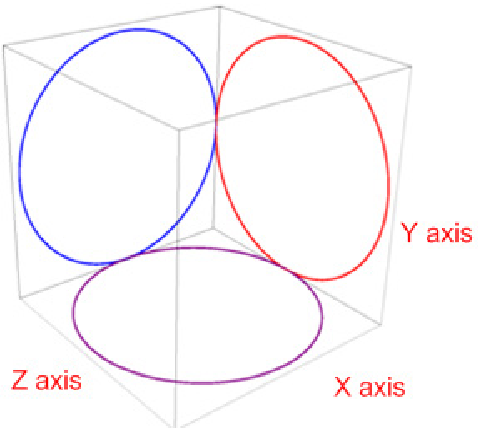
6) Up to 8,000,000 pulses can be output along the radius during arc interpolation. When converted according to the default ratio, the radius is 4000 mm

7) The number of pulses output at the third axis shall be no more than 0.9 times that to be output along the arc during helix interpolation; otherwise, an error is returned.

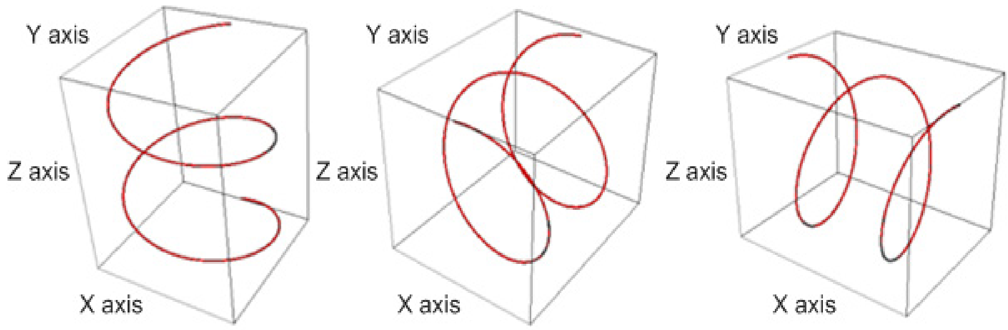


8) During arc interpolation (CW, CCW, G02, or G03), interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. For example, parameters of the x-axis prevail on the XY plane; parameters of the y-axis prevail on the YZ plane; and parameters of the z-axis prevail on the ZX plane.

- Arc interpolation, and switchover of arc interpolation among XY, YZ, and XZ planes are supported.



- Helix interpolation. To perform helix interpolation, set a non-zero value on an axis (the third axis) on which the current interpolation is not performed. For example, when arc interpolation is performed on the XY plane, set Z to 100 to perform helix interpolation.



◆ Note:

1) Interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. For example, parameters of the x-axis prevail on the XY plane; parameters of the y-axis prevail on the YZ plane; and parameters of the z-axis prevail on the ZX plane.

2) Multiple instruction forms are supported. The axis for which the function words X, Y, Z, and F are omitted inherits the running speed of the previous interpolation instruction. If I, J or K is omitted, it indicates 0. R cannot be omitted.

Instruction Form	Description
CW/CCW X_ I_	2-axis arc interpolation on the XY plane
CW/CCW X_ I_ F_	
CW/CCW X_ J_	
CW/CCW X_ J_ F_	
CW/CCW Y_ I_	
CW/CCW Y_ I_ F_	
CW/CCW Y_ J_	
CW/CCW Y_ J_ F_	
CW/CCW X_ I_ J_	
CW/CCW X_ I_ J_ F_	
CW/CCW Y_ I_ J_	
CW/CCW Y_ I_ J_ F_	
CW/CCW X_ Y_ I_	
CW/CCW X_ Y_ I_ F_	
CW/CCW X_ Y_ J_	
CW/CCW X_ Y_ J_ F_	
CW/CCW X_ Y_ I_ J_	
CW/CCW X_ Y_ I_ J_ F_	
CW/CCW X_ Z_ I_	3-axis helix interpolation on the XY plane, with the z-axis used as the third axis
CW/CCW X_ Z_ I_ F_	
CW/CCW X_ Z_ J_	
CW/CCW X_ Z_ J_ F_	
CW/CCW Y_ Z_ I_	
CW/CCW Y_ Z_ I_ F_	
CW/CCW Y_ Z_ J_	
CW/CCW Y_ Z_ J_ F_	
CW/CCW X_ Z_ I_ J_	
CW/CCW X_ Z_ I_ J_ F_	
CW/CCW Y_ Z_ I_ J_	
CW/CCW Y_ Z_ I_ J_ F_	
CW/CCW X_ Y_ Z_ I_	
CW/CCW X_ Y_ Z_ I_ F_	
CW/CCW X_ Y_ Z_ J_	
CW/CCW X_ Y_ Z_ J_ F_	
CW/CCW X_ Y_ Z_ I_ J_	
CW/CCW X_ Y_ Z_ I_ J_ F_	
CW/CCW X_ R_	2-axis arc interpolation on the XY plane in R mode
CW/CCW X_ R_ F_	
CW/CCW Y_ R_	
CW/CCW Y_ R_ F_	
CW/CCW X_ Y_ R_	
CW/CCW X_ Y_ R_ F_	
CW/CCW X_ Z_ R_	3-axis helix interpolation on the XY plane in R mode, with the z-axis used as the third axis
CW/CCW X_ Z_ R_ F_	
CW/CCW Y_ Z_ R_	
CW/CCW Y_ Z_ R_ F_	
CW/CCW X_ Y_ Z_ R_	
CW/CCW X_ Y_ Z_ R_ F_	

Instruction Form	Description
CW/CCW Y_ J_	2-axis arc interpolation on the YZ plane
CW/CCW Y_ J_ F_	
CW/CCW Z_ K_	
CW/CCW Z_ K_ F_	
CW/CCW Y_ K_	
CW/CCW Y_ K_ F_	
CW/CCW Z_ J_	
CW/CCW Z_ J_ F_	
CW/CCW Y_ J_ K_	
CW/CCW Y_ J_ K_ F_	
CW/CCW Z_ J_ K_	
CW/CCW Z_ J_ K_ F_	
CW/CCW Y_ Z_ J_	
CW/CCW Y_ Z_ J_ F_	
CW/CCW Y_ Z_ K_	
CW/CCW Y_ Z_ K_ F_	
CW/CCW Y_ Z_ J_ K_	
CW/CCW Y_ Z_ J_ K_ F_	
CW/CCW X_ Y_ J_	3-axis helix interpolation on the YZ plane, with the x-axis used as the third axis
CW/CCW X_ Y_ J_ F_	
CW/CCW X_ Z_ K_	
CW/CCW X_ Z_ K_ F_	
CW/CCW X_ Y_ K_	
CW/CCW X_ Y_ K_ F_	
CW/CCW X_ Z_ J_	
CW/CCW X_ Z_ J_ F_	
CW/CCW X_ Y_ J_ K_	
CW/CCW X_ Y_ J_ K_ F_	
CW/CCW X_ Z_ J_ K_	
CW/CCW X_ Z_ J_ K_ F_	
CW/CCW X_ Y_ Z_ J_	
CW/CCW X_ Y_ Z_ J_ F_	
CW/CCW X_ Y_ Z_ K_	
CW/CCW X_ Y_ Z_ K_ F_	
CW/CCW X_ Y_ Z_ J_ K_	
CW/CCW X_ Y_ Z_ J_ K_ F_	
CW/CCW Y_ R_	2-axis arc interpolation on the YZ plane in R mode
CW/CCW Y_ R_ F_	
CW/CCW Z_ R_	
CW/CCW Z_ R_ F_	
CW/CCW Y_ Z_ R_	
CW/CCW Y_ Z_ R_ F_	
CW/CCW X_ Y_ R_	3-axis helix interpolation on the YZ plane in R mode, with the x-axis used as the third axis
CW/CCW X_ Y_ R_ F_	
CW/CCW X_ Z_ R_	
CW/CCW X_ Z_ R_ F_	
CW/CCW X_ Y_ Z_ R_	
CW/CCW X_ Y_ Z_ R_ F_	

Instruction Form	Description
CW/CCW X_ I_	2-axis arc interpolation on the ZX plane
CW/CCW X_ I_ F_	
CW/CCW X_ K_	
CW/CCW X_ K_ F_	
CW/CCW Z_ I_	
CW/CCW Z_ I_ F_	
CW/CCW Z_ K_	
CW/CCW Z_ K_ F_	
CW/CCW X_ I_ K_	
CW/CCW X_ I_ K_ F_	
CW/CCW Z_ I_ K_	
CW/CCW Z_ I_ K_ F_	
CW/CCW X_ Z_ I_	
CW/CCW X_ Z_ I_ F_	
CW/CCW X_ Z_ K_	
CW/CCW X_ Z_ K_ F_	
CW/CCW X_ Z_ I_ K_	
CW/CCW X_ Z_ I_ K_ F_	
CW/CCW X_ Y_ I_	3-axis helix interpolation on the ZX plane, with the y-axis used as the third axis
CW/CCW X_ Y_ I_ F_	
CW/CCW X_ Y_ K_	
CW/CCW X_ Y_ K_ F_	
CW/CCW Y_ Z_ I_	
CW/CCW Y_ Z_ I_ F_	
CW/CCW Y_ Z_ K_	
CW/CCW Y_ Z_ K_ F_	
CW/CCW X_ Y_ I_ K_	
CW/CCW X_ Y_ I_ K_ F_	
CW/CCW Y_ Z_ I_ K_	
CW/CCW Y_ Z_ I_ K_ F_	
CW/CCW X_ Y_ Z_ I_	
CW/CCW X_ Y_ Z_ I_ F_	
CW/CCW X_ Y_ Z_ K_	
CW/CCW X_ Y_ Z_ K_ F_	
CW/CCW X_ Y_ Z_ I_ K_	
CW/CCW X_ Y_ Z_ I_ K_ F_	
CW/CCW X_ R_	2-axis arc interpolation on the ZX plane in R mode
CW/CCW X_ R_ F_	
CW/CCW Z_ R_	
CW/CCW Z_ R_ F_	
CW/CCW X_ Z_ R_	
CW/CCW X_ Z_ R_ F_	
CW/CCW X_ Y_ R_	3-axis helix interpolation on the ZX plane in R mode, with the y-axis used as the third axis
CW/CCW X_ Y_ R_ F_	
CW/CCW Y_ Z_ R_	
CW/CCW Y_ Z_ R_ F_	
CW/CCW X_ Y_ Z_ R_	
CW/CCW X_ Y_ Z_ R_ F_	

3) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

4) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

7

X-axis	Y-axis	Z-axis	Attribute
SD6, SD7	SD106, SD107	SD206, SD207	Number of pulses required when the motor rotates a circle (A)
SD8, SD9	SD108, SD109	SD208, SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

5) Only trapezoid acceleration/deceleration is supported.

6) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

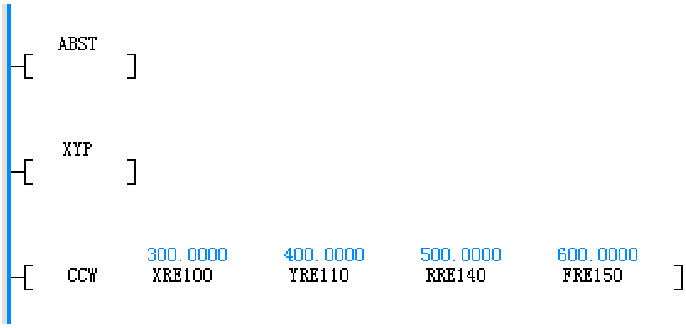
The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12, SD13	SD112, SD113	SD212, SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

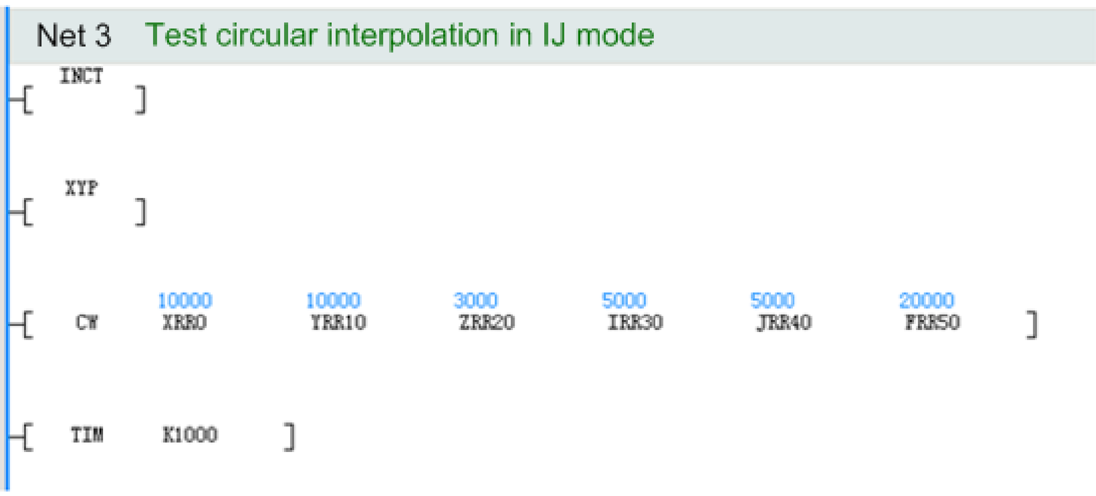
7) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$Vmin = \sqrt{\frac{Vset\ (Hz)}{2 \times Tacc\ (ms) / 1000}}$$

■ Program example



It indicates that, in absolute position mode, counterclockwise arc interpolation is performed on the x- and y-axes from the current position to (300 mm, 400 mm), and the radius is 500 mm. The interpolation speed is 600 mm/min.



It indicates that, in relative position mode, helix interpolation is performed on x-, y, and x-axes relative to the current position on the XY plane. Arc interpolation is performed on the x- and y-axes. (10000, 10000) pulses are output at the end point relative to the current position, and (5000, 5000) pulses are output at the center coordinate relative to the current position. Besides, 3000 pulses are output relative to the current position on the z-axis. The interpolation speed is 20,000 Hz.

TIM: Delay waiting

◆ Overview

The TIM instruction is used to set the delay before the next motion control instruction is executed.

TIM 1		Delay waiting	Applicable model: H3U-PM	
1	Delay	Delay, in ms		

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
1	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported.

◆ Functions and actions

The next instruction can be executed when the specified delay expires. The unit is ms.

Example:

It indicates that the delay is 2000 ms.

MCALL: Motion control subprogram calling

◆ Overview

The MCALL instruction is used to call the specified motion control subprogram continuously for the specified times.

MCALL P_ L_		Motion control subprogram calling	Applicable model: H3U-PM	
P	Subprogram number	Serial number of the motion control subprogram to be called		
L	Number of calls	Number of times the subprogram is called		

◆ Operands

Parameter	Bit Element		Word Element									Immediate operand				
P	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
L	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The immediate operand type is not displayed. For example, P10 indicates that P is the integer 10.

L can be omitted, indicating that the subprogram is called once by default. Currently, the number of calls cannot be set, and the subprogram can be called only once.

◆ Functions and actions

An MC subprogram can execute MCALL to call other MC subprograms and G-code subprograms, wherein the G-code subprograms can be called only as a whole.

The called motion control subprograms will be executed first, and other subprograms will be called starting from the current call point after MRET is returned. Up to six nested layers are supported.

MRET: Motion control subprogram return

◆ Overview

The MRET instruction is used to return to the call layer of the previous level after execution of the current motion control subprogram is finished.

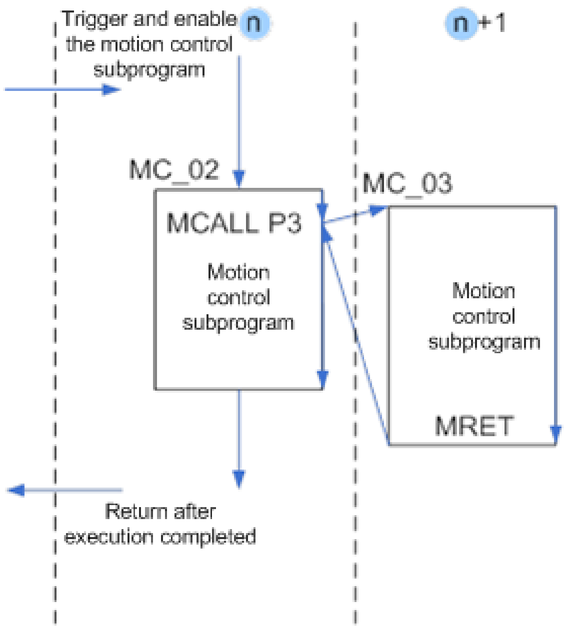
MRET	Motion control subprogram return	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

The following is an example.



MC02 runs MCALL P3 to call MC03, and MRET is returned after execution of MC03 is finished.

◆ Note

1) MCALL can be used only in MC subprograms but cannot be used to call itself. Up to six nested layers are supported.

2) MC subprograms (CNC00, corresponding to MC10000) can call G-code subprograms. MCALL P10000 indicates calling MC10000.

3) This instruction is automatically added by programming tools and needs not be entered manually.

7

ABST: Absolute position modal; INCT: Relative position modal

◆ Overview

The ABST and INCT modal instructions are used to configure the current motion control coordinate system in absolute or relative position mode.

ABST	Absolute position modal	Applicable model: H3U-PM

INCT	Relative position modal	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

Current running of a motion control subprogram means that the subprogram is always in running state after the SM90 enabling flag is active. The current running is finished when the SM91 complete flag switches to ON. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope. The modal instruction enabled in the current running remains active until the execution is completed or changed.

After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.

ABST and INCT are mutually exclusive modal instructions. The current modal remains unchanged after being enabled until another modal is enabled. The default modal is in absolute position mode on the XY plane.

XYP, YZP, and ZXP: Setting a modal for the current plane

◆ Overview

The XYP, YZP, and ZXP modal instructions are used to configure the main plane of the current motion control coordinate system as the XY plane, YZ plane, or ZX plane. They are mainly used for arc and helix interpolation.

XYP	Modal instruction for the XY plane	Applicable model: H3U-PM

YZP	Modal instruction for the YZ plane	Applicable model: H3U-PM

ZXP	Modal instruction for the ZX plane	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

Current running of a motion control subprogram means that the subprogram is always in running state after the SM90 enabling flag is active. The current running is finished when the SM91 complete flag switches to ON. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope. The modal instruction enabled in the current running remains active until the execution is completed or changed.

After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.

XYP, YZP, and ZXP are mutually exclusive modal instructions. The current modal remains unchanged after being enabled until another modal is enabled. The default modal is in absolute position mode on the XY plane.

SETT: Setting the current position

◆ Overview

The SETT instruction is used to set the current position of the x-, y, and z-axes to the designated value.

SETT X_ Y_ Z_			Setting current position	Applicable model: H3U-PM	
X	Position setting of the x-axis	Current position setting of the x-axis			
Y	Position setting of the y-axis	Current position setting of the y-axis			
Z	Position setting of the z-axis	Current position setting of the z-axis			

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

It is used to set the current position. The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

This instruction is used to modify the current position to the value set by the user.

For example, if the current position is (100, 200, 300), when SETT XKK123 YKK456 ZKK789 is used, the current position is changed to (123, 456, 789). The registers are modified synchronously.

7

◆ Note:

1) Multiple instruction forms are supported. Instructions can be used in combination. The current position of the axis not involved in the instruction remains unchanged.

Instruction Form	Description
SETT X_ Y_ Z_	Standard format
SETT X_	
SETT Y_	
SETT Z_	
SETT X_ Y_	
SETT Y_ Z_	
SETT X_ Z_	

2) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6, SD7	SD106, SD107	SD206, SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

SETR: Setting the electrical origin

◆ Overview

The SETR instruction is used to set the current position of the x-, y, and z-axes to the electrical origin. It is related to DRVR.

SETR	Setting electrical origin	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

It is used to set the current position to the electrical origin. The regression electrical origin of DRVR is the electrical origin set by SETR. The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

This instruction is used to set the current position to the electrical origin. The current electrical origin position can be queried in special registers, for example, the 32-bit registers in the following table. The default value is (0, 0, 0).

X-axis	Y-axis	Z-axis	Attribute
SD26, SD27	SD126, SD127	SD226, SD227	Electrical origin position (only for display purpose)

The registers are modified synchronously after the setting.

DRVZ: Mechanical zero return

◆ Overview

The x-, y-, and z-axes regress to the mechanical origin independently. Each axis searches for the DOG signal and moves toward the origin (DOG). After a DOG signal is detected, the system decelerates to the creep speed. After an origin signal is detected following DOG = OFF, the system immediately stops pulse output.

DRVZ	Mechanical zero return	Applicable Model: H3U-PM

◆ Operands

None

◆ Functions and actions

Similar to DSZR, this instruction is finished and the next instruction can be executed only when the x-, y-, and z-axes all return to the origin. Dedicated input/output signal is used during instruction execution.

The axis zero return disabling function can be used to disable the specified axis to return to the origin, and allow other axes to return to the origin.

The special SM elements are defined in the following table.

X-axis	Y-axis	Z-axis	Attribute
SM12	SM112	SM212	Flag of DRVZ zero return direction
SM18	SM118	SM218	Axis origin return disabled

The special SD elements are defined in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12 and SD13	SD112 and SD113	SD212 and SD213	Base speed (starting speed) (Vbias)
SD16 and SD17	SD116 and SD117	SD216 and SD217	Zero return speed (VRT)
SD18 and SD19	SD118 and SD119	SD218 and SD219	Zero return creep speed (VCR)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)
SD80	SD180	SD280	Selection of the input pole

- For details about the maximum speed, zero return speed, creep speed, and base speed, see Page 704, "A.2 Special Soft Element Register Range". The following rules must be followed:

  ① Base speed ≤ zero return speed ≤ maximum speed

  ② Base speed ≤ creep speed ≤ maximum speed

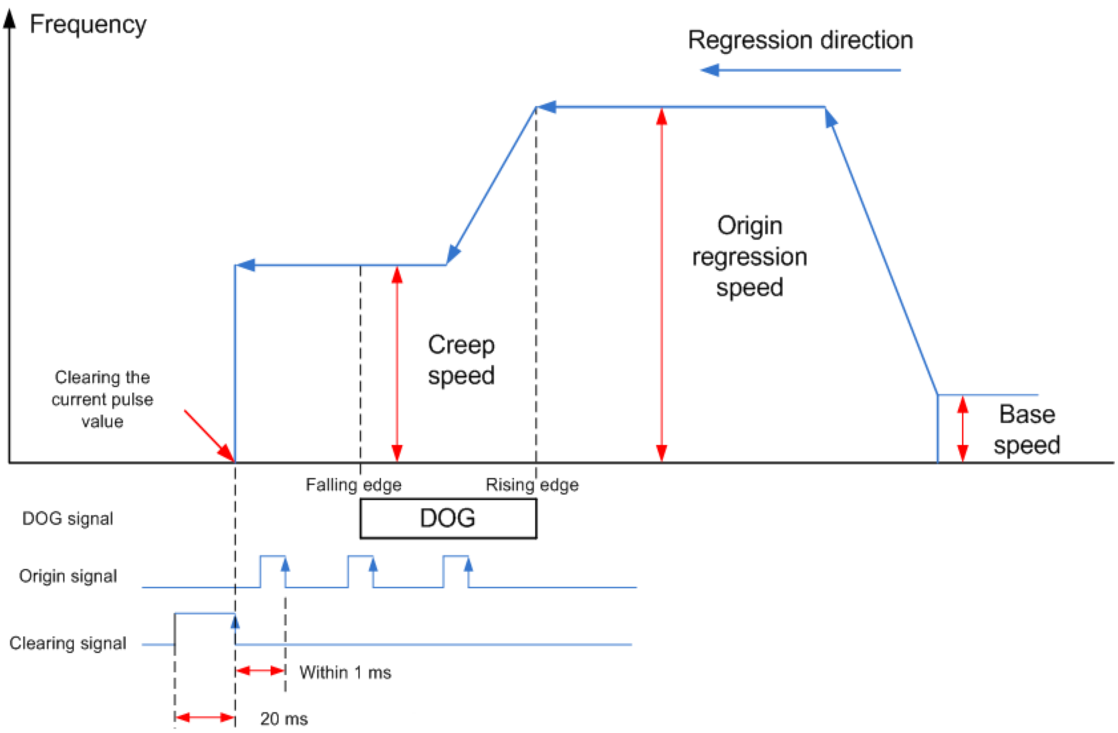
- Select appropriate parameters based on the set parameter value range. It is suggested that the creep speed be smaller than or equal to the zero return speed.

  ① The zero return speed ranges from 10 to 200,000 Hz.

② The creep speed ranges from 10 to 32,767 Hz.

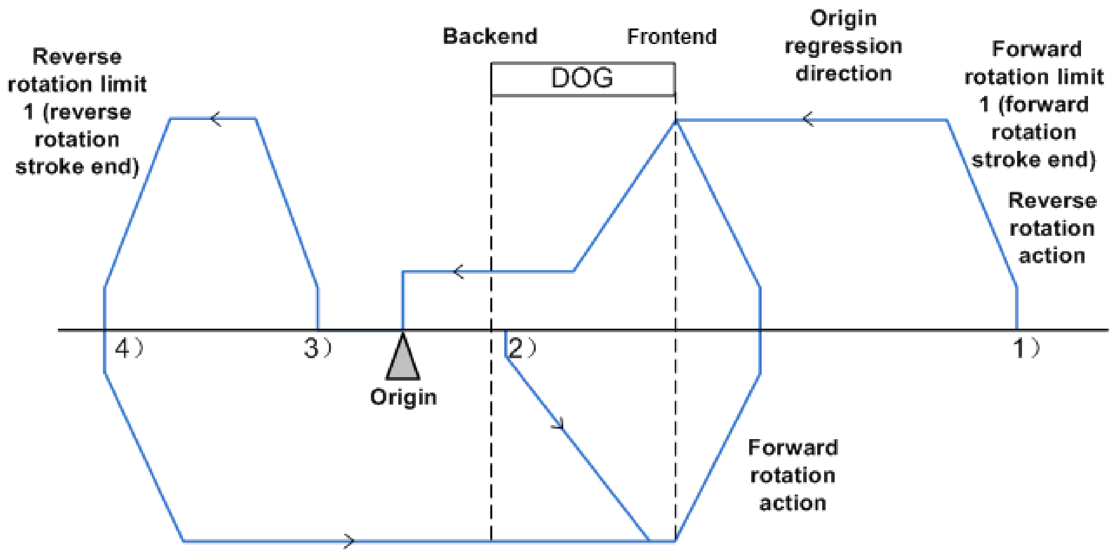
③ The base speed ranges from 10 to 32,767 Hz.

- When the PLC works in combination with the servo driver, this instruction is used to output pulses at the pulse speed specified by the special register through the pulse output port, so that the actuator moves to the action origin according to the preset action sequence; when the near point signal (DOG) switches from ON to OFF during running, and the origin signal switches from OFF to ON, the PLC immediately stops outputting the pulse. A clearing signal is output when regression is finished.
- In a system with forward/reverse rotation limit, the DSZR with DOG search mode can be enabled. In a system without forward/reverse rotation limit or not using the forward/reverse rotation limit for zero return, the zero return can be performed by specifying the zero return direction.
- You can set the zero return direction flag to ON or OFF to specify the zero return direction. The base speed is accelerated to the zero return speed and the actuator moves toward the direction specified by the zero return direction flag. When the system detects that the near point signal (DOG) is ON, the system decelerates to the creep speed. After the near point signal (DOG) switches from ON to OFF, if the system detects that the origin signal switches from OFF to ON, the system stops outputting the pulse immediately.
- After the pulse output stops (within 1 ms), the clearing signal keeps ON within 20 ms. The zero return action is finished.
- This is the description about the case where the logical inversion flag of the near point signal and the origin signal is set to OFF. If the logical inversion flag is set to ON, the ON and OFF states of the corresponding near point signal and origin signal must be changed to each other.
- The following is a single axis pulse output diagram.



◆ DOG search

Under the circumstance that there're forward/reverse limits in the system, when the zero return instruction with DOG search function is executed, the zero return action is subject to the start position of zero return



1) When the start position is before DOG (including the case where the forward rotation limit 1 is set to ON):

- The system executes the zero return instruction to start the zero return action.
- The system moves to the zero return direction at the zero return speed.
- The system decelerates to the creep speed upon detecting the DOG frontend.
- After detecting the DOG backend, the system stops upon detecting the first origin signal.

2) When the start position is within the DOG:

- The system executes the zero return instruction to start the zero return action.
- The system moves to the direction opposite to the zero return direction at the zero return speed.
- The system decelerates to stop upon detecting the DOG frontend. (Leaving DOG)
- The system moves to the zero return direction at the zero return speed. (Entering DOG again)
- The system decelerates to the creep speed upon detecting the DOG frontend.
- After detecting the DOG backend, the system stops upon detecting the first origin signal.

3) When the start position is after the DOG (the near point signal is set to OFF):

- The system executes the zero return instruction to start the zero return action.
- The system moves to the zero return direction at the zero return speed.
- The system decelerates to stop upon detecting the reverse rotation limit 1 (reverse rotation limit).
- The system moves to the direction opposite to the zero return direction at the zero return speed.
- The system decelerates to stop upon detecting the DOG frontend. (Detecting [Leaving] DOG)
- The system moves to the zero return direction at the zero return speed. (Entering DOG again)
- The system decelerates to the creep speed upon detecting the DOG frontend.
- After detecting the DOG backend, the system stops upon detecting the first origin signal.

4) When the limit switch (reverse rotation limit 1) of the zero return direction is set to ON:

- The system executes the zero return instruction to start the zero return action.
- The system moves to the direction opposite to the zero return direction at the zero return speed.
- The system decelerates to stop upon detecting the DOG frontend. (Detecting [Leaving] DOG)
- The system moves to the zero return direction at the zero return speed. (Entering DOG again)
- The system decelerates to the creep speed upon detecting the DOG frontend.
- After detecting the DOG backend, the system stops upon detecting the first origin signal.

5) Note: When designing the near point signal (DOG), you need to design a sufficient ON time for the system to decelerate to the creep speed. The creep speed must be as low as possible. If the system stops immediately without deceleration, a great creep speed may result in position offset.

◆ Note

1) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

2) Only trapezoid acceleration/deceleration is supported.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12, SD13	SD112, SD113	SD212, SD213	Base speed (starting speed) (Vbias)
SD16 and SD17	SD116 and SD117	SD216 and SD217	Zero return speed (VRT)
SD18, SD19	SD118, SD119	SD218, SD219	Zero return creep speed (VCR)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

4) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

5) Specifying the zero return direction

The zero return direction flag of special elements can be used to specify the zero return direction as forward direction. See the following table.

X-axis	Y-axis	Z-axis	Attribute
SM12	SM112	SM212	Flag of DRVZ zero return direction

6) Disabling zero return for a specific axis

The axis zero return disabling function of special elements can be used to disable the specified axis to return to the origin, and allow other axes to return to the origin. See the following table.

X-axis	Y-axis	Z-axis	Attribute
SM18	SM118	SM218	Axis origin return disabled

7) Logic signal inversion

See the following table.

OFF: Positive logic (when the input is ON, the near point/origin signal is ON).

ON: Negative logic (when the input is OFF, the near point/origin signal is ON).

X-axis	Y-axis	Z-axis	Attribute
SD80	SD180	SD280	Selection of the input pole

SDX80: Selects the input pole

Bits of this element are defined in the following table.

b0	Enter the A signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b1	Enter the B signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b2	Enter the START signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b3	Enter the DOG signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b4	Enter the STOP signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b5	Enter the LSN signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b6	Enter the LSP signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b7	Enter the PG signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b15 to b8	Reserved

DRVR: Electrical zero return

◆ Overview

The DRVR instruction is used to enable the x-, y-, and z-axes to return to the electrical origin independently at their specified maximum speed. It is related to SETR.

DRVR	Electrical zero return	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

The x-, y-, and z-axes return to the electrical origin specified by SETR at the specified maximum speed. This instruction is finished and the next instruction can be executed only when all of the three axes return to the origin. This instruction is used to perform 3-axis absolute positioning.

The regression electrical origin of DRVR is the electrical origin set by SETR. The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

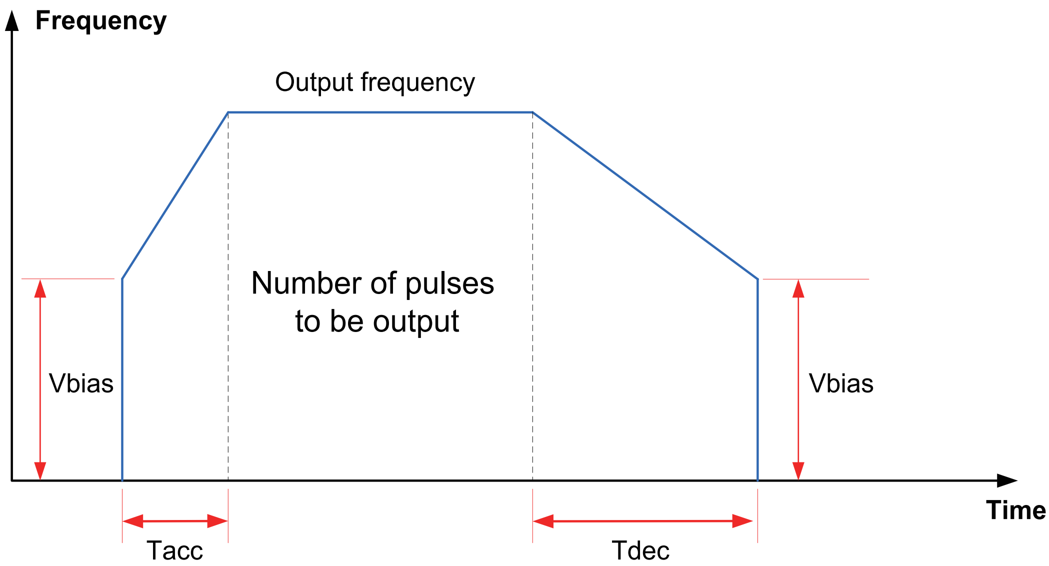
The current electrical origin position can be queried in special registers, for example, the 32-bit registers in the following table. The default value is (0, 0, 0).

X-axis	Y-axis	Z-axis	Attribute
SD26, SD27	SD126, SD127	SD226, SD227	Electrical origin position (only for display purpose)

The maximum speed of each axis can be set by special registers, for example, the following 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)

The following is a single axis pulse output diagram.



◆ Note

1) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

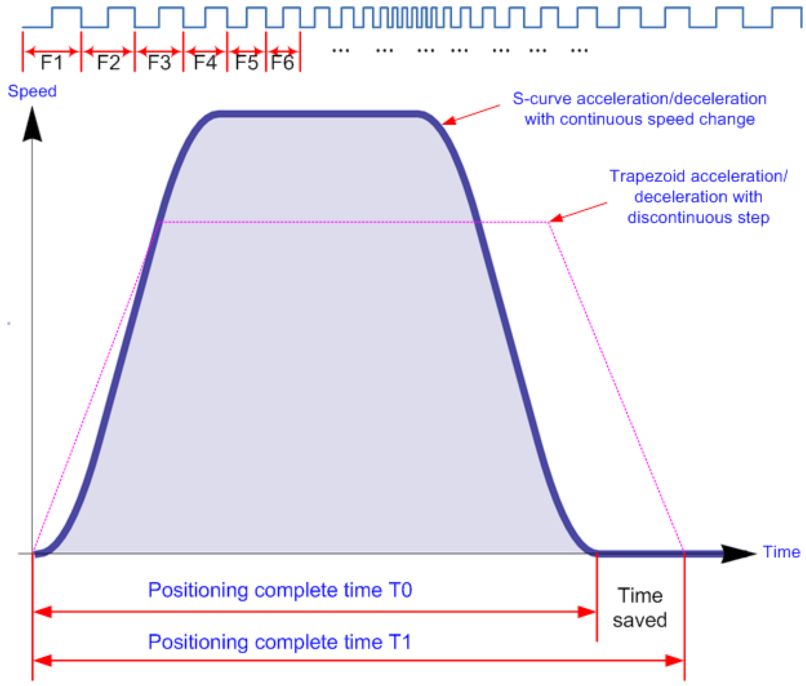
7

2) Trapezoid acceleration/deceleration and S-curve acceleration/deceleration are supported.

The two acceleration/deceleration modes can be distinguished by setting the "S-curve acceleration/deceleration enabling flag" of special elements. If the flag is not set, the trapezoid acceleration/deceleration mode is used by default. The following table lists details about S-curve acceleration/deceleration.

X-axis	Y-axis	Z-axis	Attribute	Remarks
SM17	SM117	SM217	S-curve acceleration enabling flag	ON indicates an S-curve, and OFF indicates a trapezoid curve.

DRV, G00, and DRVR support S-curve acceleration/deceleration. Therefore, at given mechanical stability, the target speed is increased, the positioning time is shortened, and the processing efficiency is improved.



The advanced pulse-by-pulse modulation algorithm is used for S-curve acceleration/deceleration. The frequency of each pulse is adjusted to ensure more smooth positioning.

3) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms. The time of trapezoid acceleration/deceleration and S-curve acceleration/deceleration can be set separately.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12, SD13	SD112, SD113	SD212, SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

4) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

SINTR: Single-speed interrupt positioning; DINTR: Double-speed interrupt positioning

◆ Overview

The SINTR and DINTR instructions are used to output pulses at the specified frequency. When an interrupt input signal (DOG) is detected, the single-speed interrupt positioning speed remains unchanged, the double-speed interrupt positioning speed accelerates or decelerates to the output frequency after interrupt, and the specified number of pulses are output. During single-speed and double-speed interrupt positioning, the number of pulses is an incremental value independent of the coordinate system.

SINTR X/Y/Z_ F_		Single-speed interrupt positioning	Applicable model: H3U-PM	
X	Interrupt fixed length	Number of pulses to be output after interrupt in relative position mode		
Y	Interrupt fixed length	Number of pulses to be output after interrupt in relative position mode		
Z	Interrupt fixed length	Number of pulses to be output after interrupt in relative position mode		
F	Output frequency	Specified output frequency		

DINTR X/Y/Z_ F_ F_		Double-speed interrupt positioning	Applicable model: H3U-PM	
X	Interrupt fixed length	Number of pulses to be output after interrupt in relative position mode		
Y	Interrupt fixed length	Number of pulses to be output after interrupt in relative position mode		
Z	Interrupt fixed length	Number of pulses to be output after interrupt in relative position mode		
F	Output frequency before interrupt	Specified output frequency before an interrupt signal is detected		
F	Output frequency after interrupt	Specified output frequency after an interrupt signal is detected		

◆ Operands

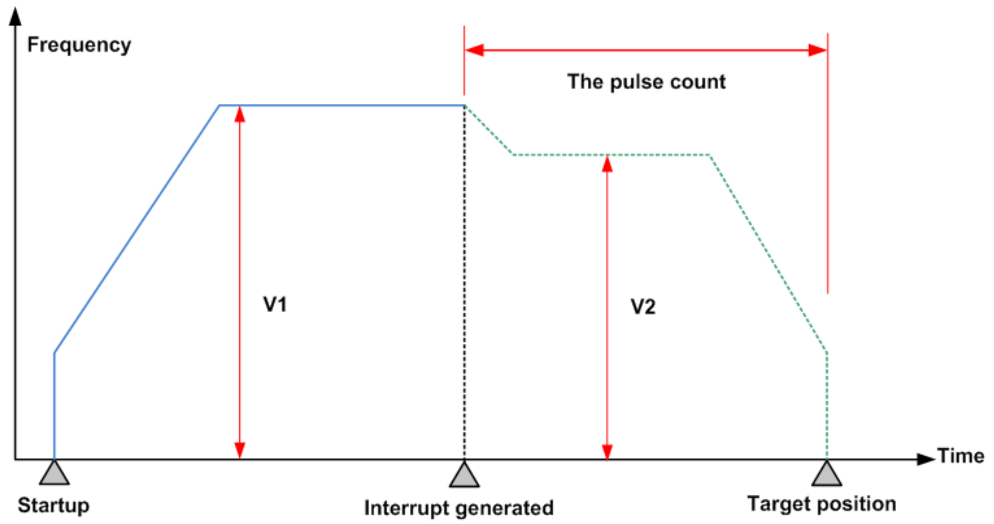
Parameter	Bit Element		Word Element									Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

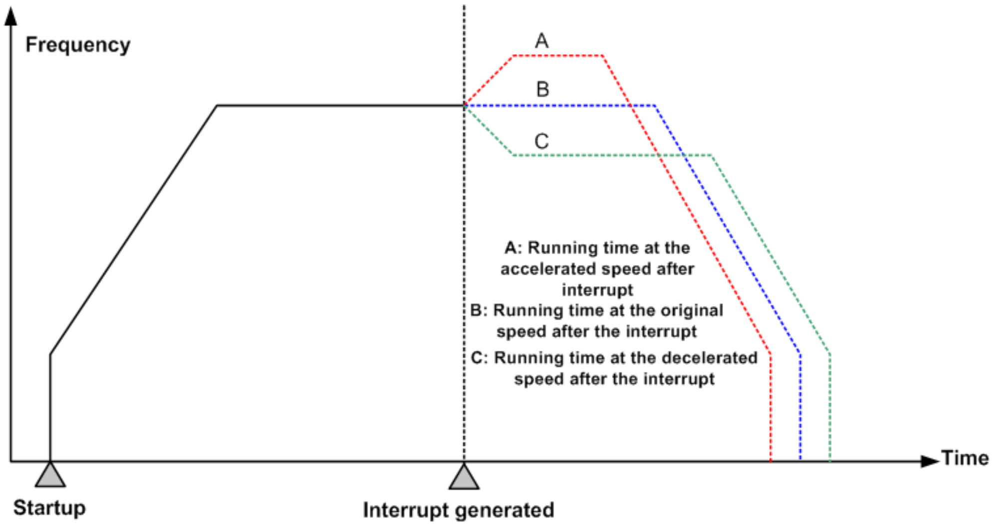
◆ Functions and actions

The SINTR and DINTR instructions are used to output pulses at the specified frequency. When an interrupt input signal (DOG) is detected, the single-speed interrupt positioning speed remains unchanged, the double-speed interrupt positioning speed accelerates or decelerates to the output frequency after interrupt, and the specified number of pulses are output. During single-speed and double-speed interrupt positioning, the number of pulses is an incremental value independent of the coordinate system.

The following is a single axis pulse output diagram.



The pulse output frequency at the speed segment before DINTR interrupts may be different from that at the position segment after the interrupt, as shown in the following figure.



◆ Note

1) Multiple instruction forms are supported. The instruction can drive only one axis.

Instruction Form	Description
SINTR X_ F_	Single-axis single-speed
SINTR Y_ F_	
SINTR Z_ F_	
DINTR X_ F_ F_	Single-axis double-speed
DINTR Y_ F_ F_	
DINTR Z_ F_ F_	

2) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6, SD7	SD106, SD107	SD206, SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

3) Only trapezoid acceleration/deceleration is supported.

4) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms.

7

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

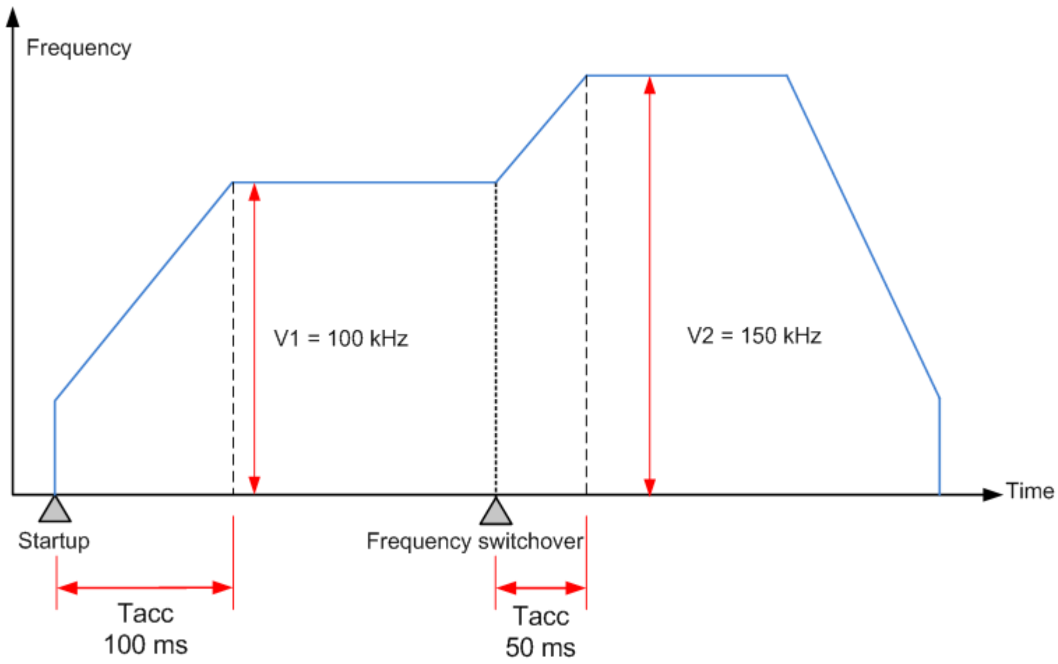
X-axis	Y-axis	Z-axis	Attribute
SD10, SD11	SD110, SD111	SD210, SD211	Maximum speed (Vmax)
SD12, SD13	SD112, SD113	SD212, SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

5) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

6) During execution of a multi-segment pulse output instruction, the acceleration/deceleration computation depends on the first speed. The acceleration/deceleration slope (speed) remains unchanged during frequency switch.

For example, if the output frequency is 100 kHz and the acceleration time is 100 ms at the first speed while the output frequency at the second speed is 150kHz, it takes about 50 ms to accelerate from the first speed to the second speed. It works similarly in deceleration mode. See the following figure.


Frequency
V1 = 100 kHz
V2 = 150 kHz
Time
Startup
Frequency switchover
Tacc
100 ms
Tacc
50 ms

MOVC: Linear displacement compensation

◆ Overview

The MOVC instruction is used to compensate each axis according to the specified value during linear interpolation. The compensation value is a relative value independent of the coordinate system. This instruction relates to LIN, INTR, and G01.

MOVC X_ Y_ Z_			Linear displacement compensation	Applicable model: H3U-PM	
X	Compensation value for the x-axis	Compensation for the x-axis during linear interpolation			
Y	Compensation value for the y-axis	Compensation for the y-axis during linear interpolation			
Z	Compensation value for the z-axis	Compensation for the z-axis during linear interpolation			

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

The MOVC instruction is used to compensate each axis according to the specified value during linear interpolation. The compensation value is a relative value independent of the coordinate system. This instruction relates to LIN, INTR, and G01.

After the compensation value is set, each axis is compensated according to the compensation value during linear interpolation.

CNTC: Arc center compensation

◆ Overview

The CNTC instruction is used to compensate the center according to the specified value during arc interpolation in IJK (center) mode. The compensation value is a relative value independent of the coordinate system. This instruction relates to CW, CCW, G02, and G03.

CNTC I_ J_ K_			Arc center compensation	Applicable model: H3U-PM	
I	Center compensation value for the x-axis	Compensation for the center on the x-axis			
J	Center compensation value for the y-axis	Compensation for the center on the y-axis			
K	Center compensation value for the z-axis	Compensation for the center on the z-axis			

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
I	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
J	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
K	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

The CNTC instruction is used to compensate the center according to the specified value during arc interpolation in IJK (center) mode. The compensation value is a relative value independent of the coordinate system. This instruction relates to CW, CCW, G02, and G03.

After the compensation value is set, each axis is compensated according to the compensation value during arc interpolation.

RADC: Arc radius compensation

◆ Overview

The RADC instruction is used to compensate the radius according to the specified value during arc interpolation in R (radius) mode. The compensation value is a relative value independent of the coordinate system. This instruction relates to CW, CCW, G02, and G03.

RADC R_			Arc radius compensation	Applicable Model: H3U-PM	
R	Radius compensation value	Compensation for the radius during arc interpolation			

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
R	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The floating-point immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

The RADC instruction is used to compensate the radius according to the specified value during arc interpolation in R (radius) mode. The compensation value is a relative value independent of the coordinate system. This instruction relates to CW, CCW, G02, and G03.

7

After the compensation value is set, each axis is compensated according to the compensation value during arc interpolation.

CANC: Motion compensation cancellation

◆ Overview

The CANC instruction is used to cancel the setting of displacement compensation, arc center compensation, and arc radius compensation.

CANC	Motion compensation cancellation	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

The CANC instruction is used to cancel all setting of linear compensation, arc center compensation, and arc radius compensation. It relates to such instructions as MOVC, CNTC, RADC, LIN, INTR, G01, CW, CCW, G02, and G03.

◆ Note

1) MOVC, CNTC, RADC, and CANC are modal instructions and remain active after being enabled unless the instruction is modified or canceled.

Multiple instruction forms are supported.

Instruction Form	Description
MOVC X_ Y_ Z_	Standard format
MOVC X_	
MOVC Y_	
MOVC Z_	
MOVC X_ Y_	
MOVC Y_ Z_	
MOVC X_ Z_	
CNTC I_ J_ K_	Standard format
CNTC I_	
CNTC J_	
CNTC K_	
CNTC I_ J_	
CNTC J_ K_	
CNTC I_ K_	
RADC R_	Standard format
CANC	Standard format

2) Displaying the compensation value

X-axis	Y-axis	Z-axis	Attribute
SD50 and SD51	SD150 and SD151	SD250 and SD251	Axis offset compensation value (DRV, LIN, and INTR) (only for display purpose)
SD52 and SD53	SD152 and SD153	SD252 and SD253	Axis center coordinate offset compensation value (CW, CCW, G02, and G03) (only for display purpose)
SD54 and SD55	SD154 and SD155 (reserved)	SD254 and SD255 (reserved)	Axis arc radius coordinate offset compensation value (CW, CCW, G02, and G03) (only for display purpose)

3) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6 and SD7	SD106 and SD107	SD206 and SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

MSET and MRST: Setting and resetting the bit element M

◆ Overview

The MSET and MRST instructions are used to set and reset bit elements. They are used for interaction with main programs or logic control.

MSET 1			Bit element setting	Applicable model: H3U-PM	
1	Element to be set	Bit element to be set			

MRST 1			Bit element resetting	Applicable model: H3U-PM	
1	Element to be reset	Bit element to be reset			

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
1	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported.

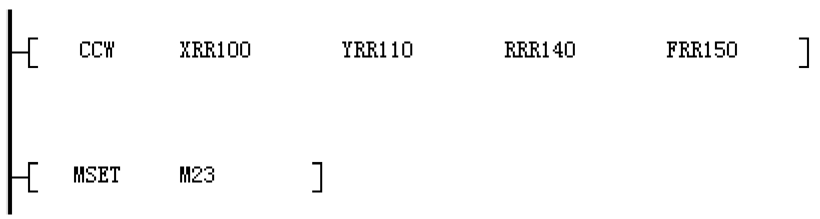
◆ Functions and actions

The MSET and MRST instructions are used to set or reset corresponding M or SM elements. They are used for interaction with main programs or logic control.

◆ Note

1. MSET and MRST support only the M and SM bit elements. They do not support the word elements, bit elements in KnX form or address indexing.
2. Function words are omitted in MSET and MRST.

◆ Application:



M23 is set after arc interpolation for other control purposes.

MMOV: Value assignment; MADD, MSUB, MMUL, MDIV: Addition, subtraction, multiplication, and division

◆ Overview

These instructions are used for value assignment, addition, subtraction, multiplication, and division operation in subprograms. 16-bit and 32-bit integers, and floating-point numbers can be used in combination for computation.

MMOV 1 2		Value assignment	Applicable model: H3U-PM	
1	Source data	Source data		
2	Target data	Operation result		

MADD 1 2 3		Addition operation	Applicable model: H3U-PM	
1	Augend	Augend		
2	Addend	Addend		
3	Target data	Operation result		

MSUB 1 2 3		Subtraction operation	Applicable model: H3U-PM	
1	Subtrahend	Subtrahend		
2	Minuend	Minuend		
3	Target data	Operation result		

MMUL 1 2 3		Multiplication operation	Applicable model: H3U-PM	
1	Multiplicand	Multiplicand		
2	Multiplier	Multiplier		
3	Target data	Operation result		

MDIV 1 2 3		Division operation	Applicable model: H3U-PM	
1	Dividend	Dividend		
2	Divider	Divider		
3	Target data	Operation result		

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
1	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
2	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
3	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported.

◆ Functions and actions

MMOV is used for value assignment.

MADD, MSUB, MMUL, and MDIV are used for addition, subtraction, multiplication, and division operations respectively.

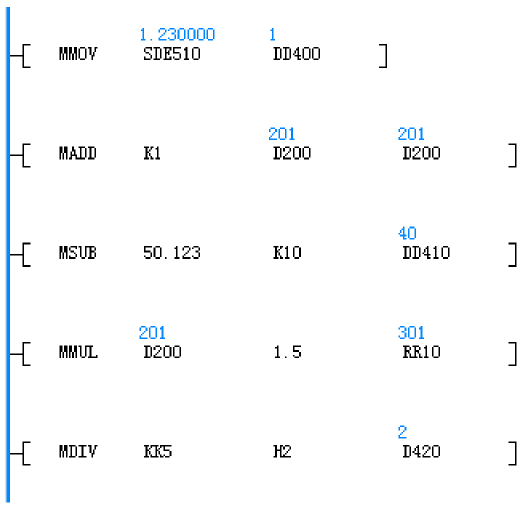
◆ Note

1) MMOV, MADD, MSUB, MMUL, and MDIV support only D, R, SD, K, H, and E. They do not support bit elements, word elements (for example, KnX) which are combination of bit elements, or address indexing.

2) Function words are omitted in MMOV, MADD, MSUB, MMUL, and MDIV.

3) MMOV, MADD, MSUB, MMUL, and MDIV support the combination of 16-bit integers, 32-bit integers, and floating-point numbers for computation.

◆ Application



The MMOV instruction is used to assigns the value of a floating-point number to a 32-bit integer.

The 16-bit integer is added by 1.

The value of the floating-point number minus 10 is assigned to a 32-bit integer.

The 16-bit integer is multiplied by the floating-point number, and the result is assigned to a 32-bit integer.

The value is divided by an integer, and the result is assigned to a 16-bit integer.

7

7.8 List of Motion Control Instructions Supported by G-code Subprograms

The following table lists instructions supported by G-code subprograms.

Motion Control Instructions	"G00: High-speed positioning"	High-speed positioning; moving to the specified position at the highest speed set. The three axes run separately.
	"G01: Linear interpolation"	Linear interpolation
	"G02: Clockwise arc interpolation; G03: Counterclockwise arc interpolation"	Clockwise arc interpolation and counterclockwise arc interpolation
	"G04: Delay waiting"	Delay waiting
	"G90: Absolute position modal; G91: Relative position modal"	Modal instruction setting. G90 indicates absolute position, and G91 indicates relative position.
	"G17, G18, and G19"	Modal instruction setting: G17 switches to the XY plane
		Modal instruction setting: G18 switches to the ZX plane
		Modal instruction setting: G19 switches to the YZ plane
	"M: Auxiliary parameter number", and "Mxxxx: Element setting parameter number"	Special auxiliary parameter numbers, including M00 to M99, wherein M02, M30, M98, and M99 have been defined.
		M element setting, including M100 to M7679
	"S and T: Auxiliary parameter number"	S auxiliary parameter number used for setting the master axis rotational speed
		T auxiliary parameter number used for selecting the tool to be used

7.9 Format and Use of G-code Subprograms

G01 X 100

① ② ③

① G-code instructions, as shown in the following table

② Type of function words

③ Parameter value

The following table lists supported instructions and function words.

	G-code	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Attribute
High-speed positioning	G00	X	Y	Z					Combination
Linear interpolation	G01	X	Y	Z	F				Combination
Clockwise arc interpolation	G02	X	Y	Z	I	J	K	F	Combination
		X	Y	Z	R	Default	Default	F	Combination
Counter-clockwise arc interpolation	G03	X	Y	Z	I	J	K	F	Combination
		X	Y	Z	R	Default	Default	F	Combination
Delay waiting	G04	P integer							
Absolute position	G90	-							
Relative position	G91	-							
XY plane	G17	-							
ZX plane	G18	-							
YZ plane	G19	-							
Auxiliary function	M00 to M99	-							
G-code-based main program end	M02	-							
G-code main program end	M30	-							
G-code subprogram call	M98	P integer	L integer						
G-code subprogram return	M99	-							
M element setting	M	M100-M7679							
Major axis speed	S								
Tool number	T								

NOTE

- These instructions can only be used in G-code subprograms.
- M00 to M99 indicate auxiliary functions which are reserved here. M100 to M7679 are used to set the M elements.

G-code instructions support only immediate operands of the floating-point type (the immediate operands can be converted to the target number of pulses according to the manually set conversion coefficient).

Parameter Supported	Floating-point Number	Integer
32-bit integer	-	G04 Pxxxx; M98 Pxxxx Lxxxx; (P and L can use only 32-bit integers, and their usage is described in the following table.)
Floating-point number	xxxx.xx	-

Among function words used in G-code subprogram instructions, X, Y, and Z indicate the position, R indicates the radius, I, J, and K indicate the center relative to the current position, and F indicates the feed speed. All these input values are in mechanical unit, and the value range is limited. See the following table.

Motion Instruction	Parameter Type	Unit	Range
X-axis position	X (floating-point)	mm	–9999.999 to –0.001, 0 0.001 to 9999.999
Y-axis position	Y (floating-point)	mm	–9999.999 to –0.001, 0 0.001 to 9999.999
Z-axis position	Z (floating-point)	mm	–9999.999 to –0.001, 0 0.001 to 9999.999
Offset of the center on the x-axis	I (floating-point)	mm	–9999.999 to –0.001, 0 0.001 to 9999.999
Offset of the center on the y-axis	J (floating-point)	mm	–9999.999 to –0.001, 0 0.001 to 9999.999
Offset of the center on the z-axis	K (floating-point)	mm	–9999.999 to –0.001, 0 0.001 to 9999.999
Radius	R (floating-point)	mm	–9999.999 to –0.001, 0.001 to 9999.999 (an error is returned when the value is 0)
Running speed	F (floating-point)	mm/min	0.01 to 99999.99 (two decimals) (an error is returned when the value is 0)
Major axis speed	S (floating-point)	r/min	0.01 to 99999.99 (two decimals) (an error is returned when the value is 0)
Tool number	T (integer)		0 to 9999
Number of the M98 subprogram	P (integer)		0 to 9999
G04 function word	P (integer)		0 to 3,000,000
Number of subprogram calls	L (integer)		0 to 9999
General M setting instruction	M (integer)		100 to 7679

For better interaction between a general logic control program and a G-code program, the system supports M element control by G-code and M element setting. M100 is used as an example. The logic control program can take relevant action when M100 is ON, and can set M100 to OFF. M0 to M99 are reserved key words for M code.

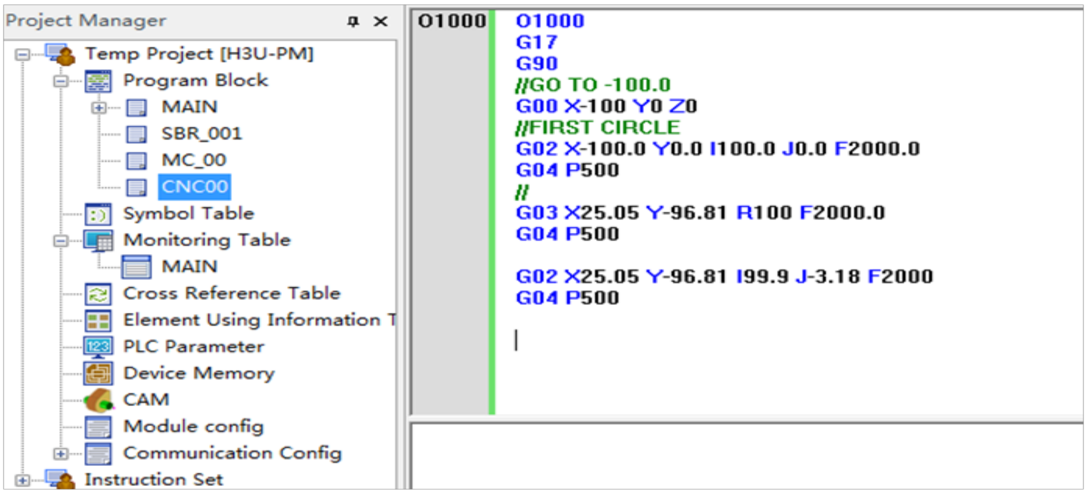
7

Towards the positioning or trajectory control, it supports the setting of acceleration/deceleration type/time, maximum speed, base speed, and they must be set in the special elements. The parameters must be set correctly; otherwise, an error may occur during running. For example, if the mechanical unit is incorrect, the pulse output frequency exceeds the maximum speed.

Parameters for an axis not used during multi-axes action need not be entered. In such a way, the axis resources are not occupied, and the axis can be used for positioning or electronic cam running.

Helix interpolation can also be performed by using the arc interpolation instruction. For example, when arc interpolation is performed on the XY plane, helix interpolation is performed by setting the displacement of the z-axis. Note that, the helix can rotate only one circle each time, and the number of pulses output on the axis in linear motion within one instruction shall not be greater than the master axis length; that is, the helix distance shall be no more than 0.9 times the perimeter.

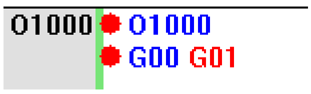
The following is a G-code editing page.



A red mark appears in the line of G-code with a syntactic error. The following rules shall be followed when G-code is input.

- One line can have only one executable instruction (G00, G01, G02, G03, or G04).

Incorrect format:



- The executable instruction can only be followed by parameters.
- Multiple environment instructions (G17, G18, G19, G90, or G91) can be placed in one line, but cannot be placed after an executable instruction, and the last parameter shall prevail. For example, G17G90G18G91G01X10 indicates that the plane and coordinate type of the current instruction are G18 and G91 respectively.
- H3U-PM supports one G-code subprogram which can be divided into O subprograms (starting with the key word 0, ranging from O0000 to O9999. They can be called by using M98.) The .nc file can be imported into G-code subprograms.

G00: High-speed positioning

◆ Overview

The G00 instruction is used to drive three axes to move to the target position or output the specified number of pulses at the specified maximum output frequency of each axis.

G00 X_ Y_ Z_		High-speed positioning	Applicable model: H3U-PM	
X	X-axis position	X axis target position		
Y	Y-axis position	Y axis target position		
Z	Z-axis position	Z axis target position		

◆ Operands

Parameter	Bit Element		Word Element										Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

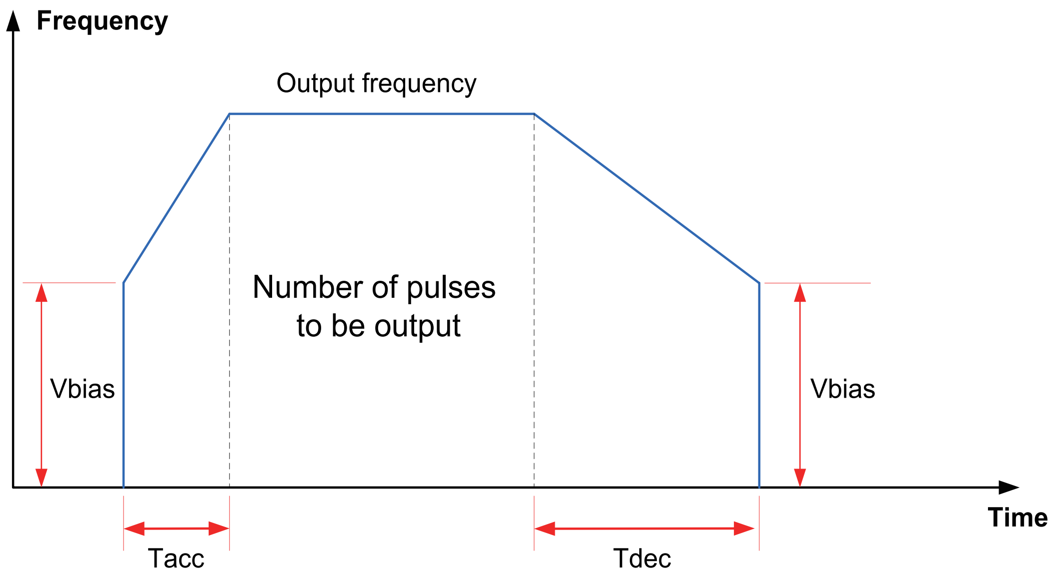
The G00 instruction is used to drive the x-, y-, and z-axes to move to the target position at the specified maximum output frequency of each axis.

Both the absolute position and relative position modes are supported.

The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36 and SD37	SD136 and SD137	SD236 and SD237	Current position (PLS), only for display purpose
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (mechanical, floating point), only for display purpose
D8340 and D8341	D8360 and D8361	D8380 and D8381	Current position (PLS)

The following is a single-pulse output diagram.



◆ Note

1) Multiple instruction forms are supported.

Instruction Form	Description
G00 X_ Y_ Z_	Standard format, 3-axis
G00 X_	Single-axis
G00 Y_	
G00 Z_	
G00 X_ Y_	2-axis
G00 Y_ Z_	
G00 X_ Z_	

2) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36 and SD37	SD136 and SD137	SD236 and SD237	Current position (PLS), only for display purpose
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (mechanical, floating point), only for display purpose
D8340 and D8341	D8360 and D8361	D8380 and D8381	Current position (PLS)

Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6 and SD7	SD106 and SD107	SD206 and SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

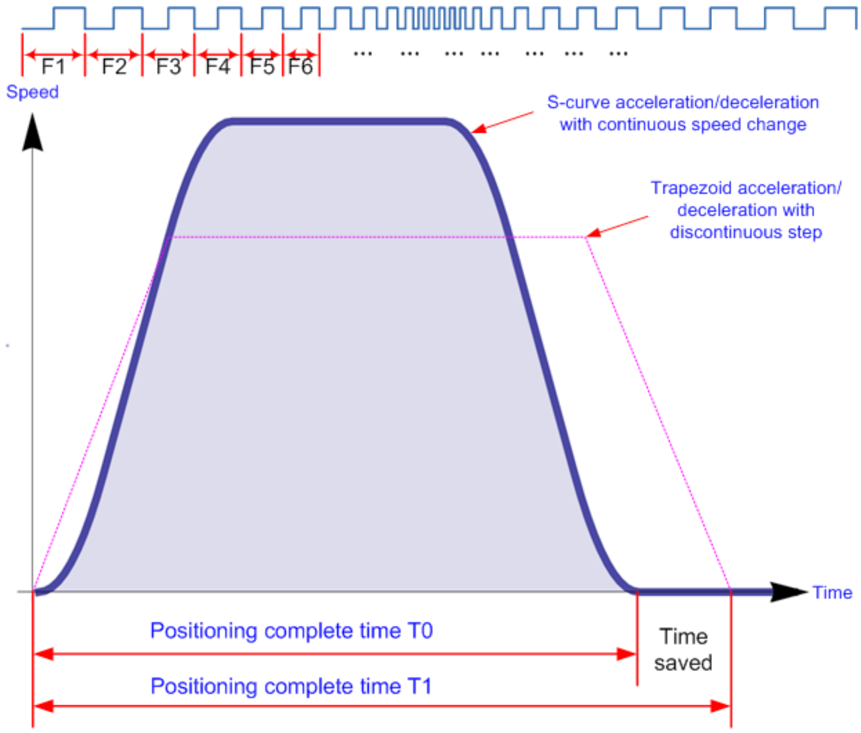
7

3) Trapezoid acceleration/deceleration and S-curve acceleration/deceleration are supported.

The two acceleration/deceleration modes can be distinguished by setting the "S-curve acceleration/deceleration enabling flag" of special elements. If the flag is not set, the trapezoid acceleration/deceleration mode is used by default. The following table lists details about S-curve acceleration/deceleration.

X-axis	Y-axis	Z-axis	Attribute	Remarks
SM17	SM117	SM217	S-curve acceleration enabling flag	ON indicates an S-curve, and OFF indicates a trapezoid curve.

DRV, G00, and DRVR support S-curve acceleration/deceleration. Therefore, at given mechanical stability, the target speed is increased, the positioning time is shortened, and the processing efficiency is improved.



The advanced pulse-by-pulse modulation algorithm is used for S-curve acceleration/deceleration. The frequency of each pulse is adjusted to ensure more smooth positioning.

4) The acceleration/deceleration time can be set separately, within the range 10 to 5000 ms. The time of trapezoid acceleration/deceleration and S-curve acceleration/deceleration can be set separately.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10 and SD11	SD110 and SD111	SD210 and SD211	Maximum speed (Vmax)
SD12 and SD13	SD112 and SD113	SD212 and SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

5) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

◆ Program example

```
G91
G00 X100.0 Y200.0 Z300.0
```

It indicates that, in relative position mode, the x-, y-, and z-axes move 100 mm, 200 mm, and 300 mm respectively relative to the current position.

7

G01: Linear interpolation

◆ Overview

The G01 instruction is used to perform linear interpolation at up to three axes at the combined output frequency.

G01 X_ Y_ Z_ F_ combination		Linear interpolation	Applicable model: H3U-PM	
X	X-axis position	X axis target position		
Y	Y-axis position	Y axis target position		
Z	Z-axis position	Z axis target position		
F	Combined interpolation speed	Combined interpolation output frequency		

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

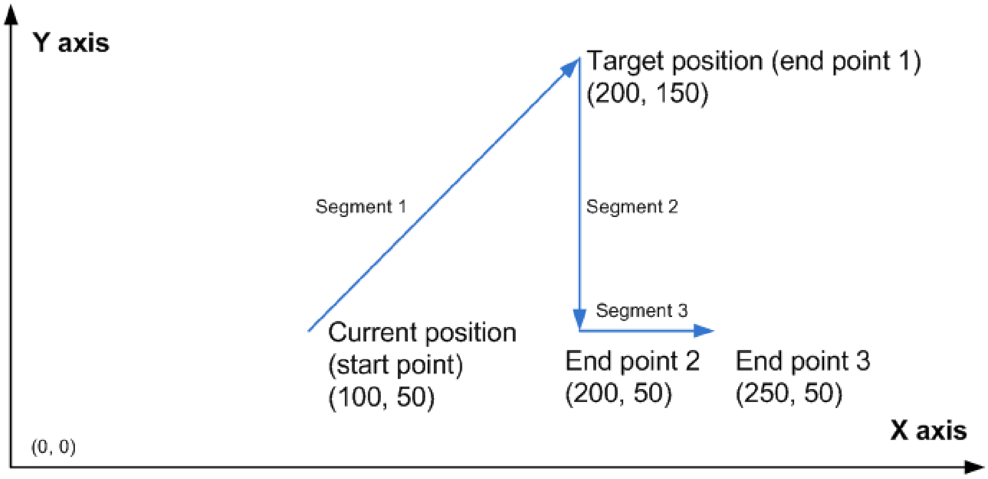
Linear interpolation can be performed at up to three axes. The axis for which the F function word is omitted inherits the running speed of the previous interpolation instruction.

Both the absolute position and relative position modes are supported.

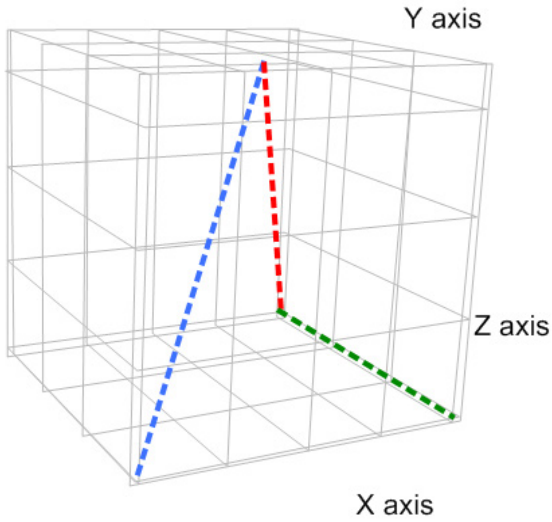
The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36 and SD37	SD136 and SD137	SD236 and SD237	Current position (PLS), only for display purpose
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (mechanical, floating point), only for display purpose
D8340 and D8341	D8360 and D8361	D8380 and D8381	Current position (PLS)

The following figure shows a pulse output diagram.



3D linear interpolation of any type is supported.



◆ Note

1) Interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. Parameters of the x-axis prevail in case of x-y-z-axes interpolation, x-y-axes interpolation, and x-z-axes interpolation; and parameters of the y-axis prevail in case of y-z-axes interpolation.

2) Multiple instruction forms are supported. The axis for which the F function word is omitted inherits the running speed of the previous interpolation instruction.

Instruction Form	Description
G01 X_ Y_ Z_ F_	Standard format
G01 X_	Single-axis
G01 X_ F_	
G01 Y_	
G01 Y_ F_	
G01 Z_	
G01 Z_ F_	
G01 X_ Y_	2-axis
G01 X_ Y_ F_	
G01 Y_ Z_	
G01 Y_ Z_ F_	
G01 X_ Z_	
G01 X_ Z_ F_	
G01 X_ Y_ Z_	3-axis

3) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36 and SD37	SD136 and SD137	SD236 and SD237	Current position (PLS), only for display purpose
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (mechanical, floating point), only for display purpose
D8340 and D8341	D8360 and D8361	D8380 and D8381	Current position (PLS)

7

4) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6 and SD7	SD106 and SD107	SD206 and SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

5) Only trapezoid acceleration/deceleration is supported.

6) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10 and SD11	SD110 and SD111	SD210 and SD211	Maximum speed (Vmax)
SD12 and SD13	SD112 and SD113	SD212 and SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

7) The minimum output frequency that can be realized actually (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

7

◆ Program example

```
G90
G01 X100.0 Y200.0 Z300.0 F300
```

It indicates that, in absolute position mode, perform linear interpolation to move from the current position to a position of 100 mm, 200 mm, and 300 mm on the x-, y-, and z-axes respectively. The interpolation speed is 300 mm/min.

G02: Clockwise arc interpolation; G03: Counterclockwise arc interpolation

◆ Overview

These instructions are used to perform clockwise or counterclockwise arc interpolation at combined output frequency at two axes. 3-axis helix interpolation on three planes is supported.

G02 X_ Y_ Z_ I_ J_ K_ F_ G02 X_ Y_ Z_ R_ F_		Clockwise arc or helix interpolation	Applicable model: H3U-PM	
X	X-axis position	X axis target position		
Y	Y-axis position	Y axis target position		
Z	Z-axis position	Z axis target position		
I	X axis center	X-axis center position, used in the IJK (center) mode		
J	Y axis center	Y axis target position, used in IJK (center) mode		
K	Z axis center	Z axis target position, used in IJK (center) mode		
R	Arc radius	Arc radius, used in R (radius) mode		
F	Combined interpolation speed	Combined interpolation output frequency		

G03 X_ Y_ Z_ I_ J_ K_ F_ G03 X_ Y_ Z_ R_ F_		Counterclockwise arc or helix interpolation	Applicable model: H3U-PM	
X	X-axis position	X axis target position		
Y	Y-axis position	Y axis target position		
Z	Z-axis position	Z axis target position		
I	X axis center	X-axis center, used in the IJK (center) mode		
J	Y axis center	Y axis target position, used in IJK (center) mode		
K	Z axis center	Z axis target position, used in IJK (center) mode		
R	Arc radius	Arc radius, used in R (radius) mode		
F	Combined interpolation speed	Combined interpolation output frequency		

7

◆ Operands

Parameter	Bit Element		Word Element										Immediate Operand				
X	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Y	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
Z	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
I	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
J	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
K	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
R	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
F	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The immediate operand type is not displayed. For example, X100 indicates the X floating point 100.00.

◆ Functions and actions

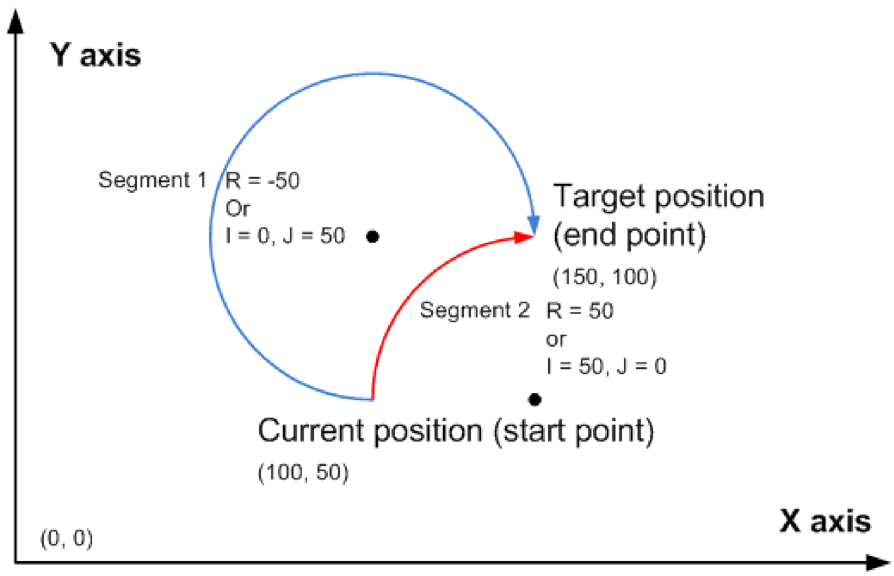
The G02 and G03 instructions are used to perform 2-axis circular arc interpolation or 3-axis helix interpolation on three planes. The axis for which the F function word is omitted inherits the running speed of the previous interpolation instruction.

Both the absolute position and relative position modes are supported. Both the absolute position and relative position are relative to the current position.

The current position can be queried in special registers, for example, the 32-bit registers listed in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD36, SD37	SD136, SD137	SD236, SD237	Current position (PLS), only for display purpose
SD40, SD41	SD140, SD141	SD240, SD241	Current position (mechanical, floating point), only for display purpose
D8340, D8341	D8360, D8361	D8380, D8381	Current position (PLS)

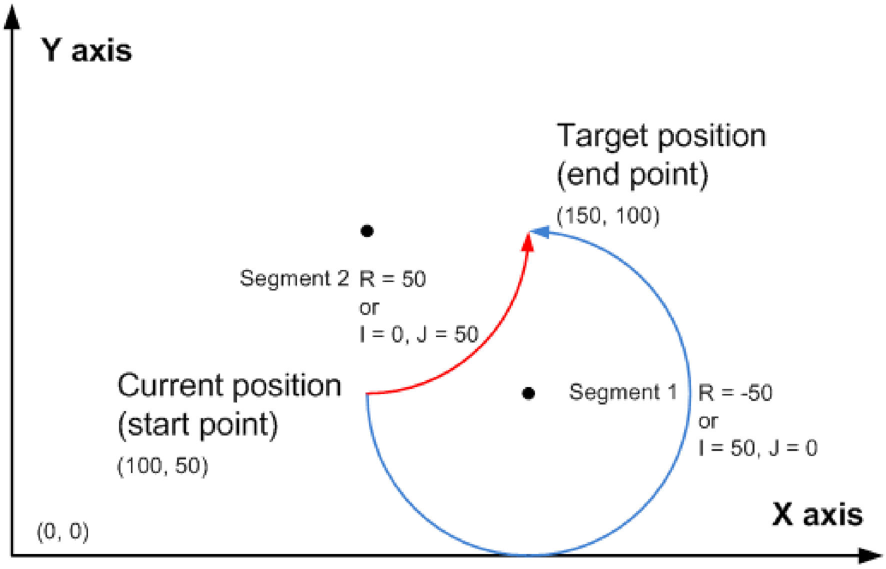
The following is a diagram of clockwise arc interpolation.



It indicates a clockwise arc interpolation, wherein (150, 100) indicates the target absolute position of the x- and y-axes. When the target position is the same, an example is provided on generation of an arc less than 180 degrees and more than 180 degrees in IJ (center coordinate) mode and in R (radius) mode respectively.

7

The following is a diagram of counterclockwise arc interpolation.



It indicates an counterclockwise arc interpolation, wherein (150, 100) indicates the target absolute position of the x- and y-axes. When the target position is the same, an example is provided on generation of an arc less than 180 degrees and more than 180 degrees in IJ (center coordinate) mode and in R (radius) mode respectively.

1) The user needs to set an appropriate target position so that the correct target circular path can be generated. In absolute position mode, when the specified target position of the axis at which the arc interpolation is performed equals its current position, a complete circle is generated. In relative position mode, when the specified target position of the axis at which the arc interpolation is performed is 0, a complete circle is generated.

2) Either the IJ (center coordinate) or R (radius) mode is supported.

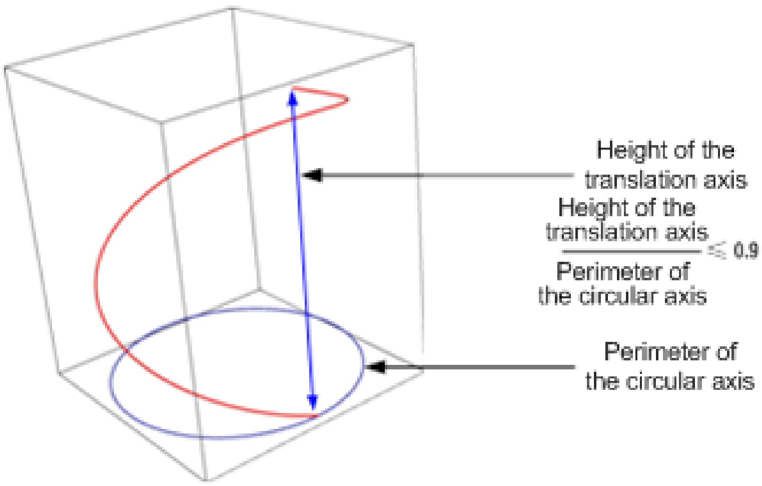
3) In IJ (center coordinate) mode, no matter it is absolute position interpolation or relative position interpolation, I, J, or K only indicates the difference (offset) of the central coordinate relative to the current position on the x-, y-, and z-axes.

4) In R (radius) mode, when the R value is larger than or equal to 0, it indicates an arc less than or equal to 180 degrees. When the R value is smaller than 0, it indicates an arc more than 180 degrees. In R (radius) mode, no complete circle can be generated.

5) More than 20 pulses shall be output along the arc during arc interpolation; otherwise, an error is returned.

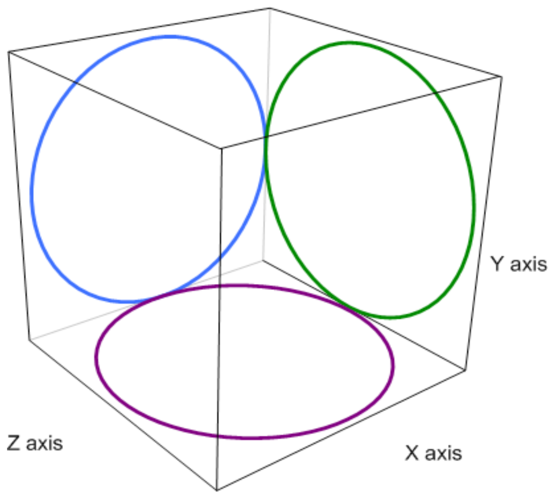
6) Up to 8,000,000 pulses can be output along the radius during arc interpolation. When converted according to the default ratio, the radius is 4000 mm

7) The number of pulses output at the third axis shall be no more than 0.9 times that to be output along the arc during helix interpolation; otherwise, an error is returned.

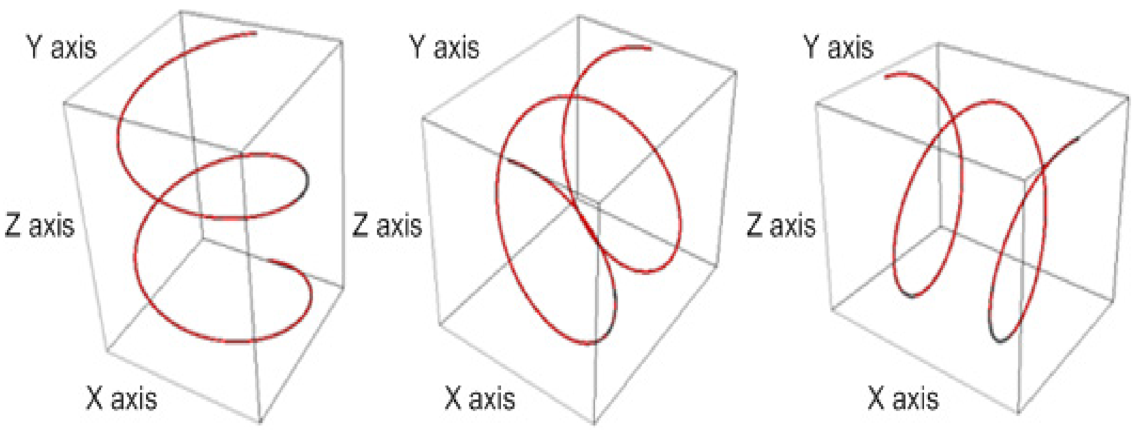


8) During arc interpolation (CW, CCW, G02, or G03), interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. For example, parameters of the x-axis prevail on the XY plane; parameters of the y-axis prevail on the YZ plane; and parameters of the z-axis prevail on the ZX plane.

Arc interpolation, and switchover of arc interpolation among XY, YZ, and XZ planes are supported.



Helix interpolation. To perform helix interpolation, set a non-zero value on an axis (the third axis) on which the current interpolation is not performed. For example, when arc interpolation is performed on the XY plane, set Z to 100 to perform helix interpolation.



◆ Note

1) Interpolation parameters, such as the acceleration/deceleration time, of the master axis prevail. For example, parameters of the x-axis prevail on the XY plane; parameters of the y-axis prevail on the YZ plane; and parameters of the z-axis prevail on the ZX plane.

2) Multiple instruction forms are supported. The axis for which the function words X, Y, Z, and F are omitted inherits the running speed of the previous interpolation instruction. If I, J or K is omitted, it indicates 0. R cannot be omitted.

Instruction Form	Description
G02/G03 X_ I_	2-axis arc interpolation on the XY plane
G02/G03 X_ I_ F_	
G02/G03 X_ J_	
G02/G03 X_ J_ F_	
G02/G03 Y_ I_	
G02/G03 Y_ I_ F_	
G02/G03 Y_ J_	
G02/G03 Y_ J_ F_	
G02/G03 X_ I_ J_	
G02/G03 X_ I_ J_ F_	
G02/G03 Y_ I_ J_	
G02/G03 Y_ I_ J_ F_	
G02/G03 X_ Y_ I_	
G02/G03 X_ Y_ I_ F_	
G02/G03 X_ Y_ J_	
G02/G03 X_ Y_ J_ F_	
G02/G03 X_ Y_ I_ J_	
G02/G03 X_ Y_ I_ J_ F_	
G02/G03 X_ Z_ I_	3-axis helix interpolation on the XY plane, with the z-axis used as the third axis
G02/G03 X_ Z_ I_ F_	
G02/G03 X_ Z_ J_	
G02/G03 X_ Z_ J_ F_	
G02/G03 Y_ Z_ I_	
G02/G03 Y_ Z_ I_ F_	
G02/G03 Y_ Z_ J_	
G02/G03 Y_ Z_ J_ F_	
G02/G03 X_ Z_ I_ J_	
G02/G03 X_ Z_ I_ J_ F_	
G02/G03 Y_ Z_ I_ J_	
G02/G03 Y_ Z_ I_ J_ F_	
G02/G03 X_ Y_ Z_ I_	
G02/G03 X_ Y_ Z_ I_ F_	
G02/G03 X_ Y_ Z_ J_	
G02/G03 X_ Y_ Z_ J_ F_	
G02/G03 X_ Y_ Z_ I_ J_	
G02/G03 X_ Y_ Z_ I_ J_ F_	

Instruction Form	Description
G02/G03 X_ R_	2-axis arc interpolation on the XY plane in R mode
G02/G03 X_ R_ F_	
G02/G03 Y_ R_	
G02/G03 Y_ R_ F_	
G02/G03 X_ Y_ R_	
G02/G03 X_ Y_ R_ F_	
G02/G03 X_ Z_ R_	3-axis helix interpolation on the XY plane in R mode, with the z-axis used as the third axis
G02/G03 X_ Z_ R_ F_	
G02/G03 Y_ Z_ R_	
G02/G03 Y_ Z_ R_ F_	
G02/G03 X_ Y_ Z_ R_	
G02/G03 X_ Y_ Z_ R_ F_	
G02/G03 Y_ J_	2-axis arc interpolation on the YZ plane
G02/G03 Y_ J_ F_	
G02/G03 Z_ K_	
G02/G03 Z_ K_ F_	
G02/G03 Y_ K_	
G02/G03 Y_ K_ F_	
G02/G03 Z_ J_	
G02/G03 Z_ J_ F_	
G02/G03 Y_ J_ K_	
G02/G03 Y_ J_ K_ F_	
G02/G03 Z_ J_ K_	
G02/G03 Z_ J_ K_ F_	
G02/G03 Y_ Z_ J_	
G02/G03 Y_ Z_ J_ F_	
G02/G03 Y_ Z_ K_	
G02/G03 Y_ Z_ K_ F_	
G02/G03 Y_ Z_ J_ K_	
G02/G03 Y_ Z_ J_ K_ F_	
G02/G03 X_ Y_ J_	3-axis helix interpolation on the YZ plane, with the x-axis used as the third axis
G02/G03 X_ Y_ J_ F_	
G02/G03 X_ Z_ K_	
G02/G03 X_ Z_ K_ F_	
G02/G03 X_ Y_ K_	
G02/G03 X_ Y_ K_ F_	
G02/G03 X_ Z_ J_	
G02/G03 X_ Z_ J_ F_	
G02/G03 X_ Y_ J_ K_	
G02/G03 X_ Y_ J_ K_ F_	
G02/G03 X_ Z_ J_ K_	
G02/G03 X_ Z_ J_ K_ F_	
G02/G03 X_ Y_ Z_ J_	
G02/G03 X_ Y_ Z_ J_ F_	
G02/G03 X_ Y_ Z_ K_	
G02/G03 X_ Y_ Z_ K_ F_	
G02/G03 X_ Y_ Z_ J_ K_	
G02/G03 X_ Y_ Z_ J_ K_ F_	

Instruction Form	Description
G02/G03 Y_ R_	2-axis arc interpolation on the YZ plane in R mode
G02/G03 Y_ R_ F_	
G02/G03 Z_ R_	
G02/G03 Z_ R_ F_	
G02/G03 Y_ Z_ R_	
G02/G03 Y_ Z_ R_ F_	
G02/G03 X_ Y_ R_	3-axis helix interpolation on the YZ plane in R mode, with the x-axis used as the third axis
G02/G03 X_ Y_ R_ F_	
G02/G03 X_ Z_ R_	
G02/G03 X_ Z_ R_ F_	
G02/G03 X_ Y_ Z_ R_	
G02/G03 X_ Y_ Z_ R_ F_	
G02/G03 X_ I_	2-axis arc interpolation on the ZX plane
G02/G03 X_ I_ F_	
G02/G03 X_ K_	
G02/G03 X_ K_ F_	
G02/G03 Z_ I_	
G02/G03 Z_ I_ F_	
G02/G03 Z_ K_	
G02/G03 Z_ K_ F_	
G02/G03 X_ I_ K_	
G02/G03 X_ I_ K_ F_	
G02/G03 Z_ I_ K_	
G02/G03 Z_ I_ K_ F_	
G02/G03 X_ Z_ I_	
G02/G03 X_ Z_ I_ F_	
G02/G03 X_ Z_ K_	
G02/G03 X_ Z_ K_ F_	
G02/G03 X_ Z_ I_ K_	
G02/G03 X_ Z_ I_ K_ F_	
G02/G03 X_ Y_ I_	3-axis helix interpolation on the ZX plane, with the Y axis used as the third axis
G02/G03 X_ Y_ I_ F_	
G02/G03 X_ Y_ K_	
G02/G03 X_ Y_ K_ F_	
G02/G03 Y_ Z_ I_	
G02/G03 Y_ Z_ I_ F_	
G02/G03 Y_ Z_ K_	
G02/G03 Y_ Z_ K_ F_	
G02/G03 X_ Y_ I_ K_	
G02/G03 X_ Y_ I_ K_ F_	
G02/G03 Y_ Z_ I_ K_	
G02/G03 Y_ Z_ I_ K_ F_	
G02/G03 X_ Y_ Z_ I_	
G02/G03 X_ Y_ Z_ I_ F_	
G02/G03 X_ Y_ Z_ K_	
G02/G03 X_ Y_ Z_ K_ F_	
G02/G03 X_ Y_ Z_ I_ K_	
G02/G03 X_ Y_ Z_ I_ K_ F_	

Instruction Form	Description
G02/G03 X_ R_	2-axis arc interpolation on the ZX plane in R mode
G02/G03 X_ R_ F_	
G02/G03 Z_ R_	
G02/G03 Z_ R_ F_	
G02/G03 X_ Z_ R_	
G02/G03 X_ Z_ R_ F_	
G02/G03 X_ Y_ R_	3-axis helix interpolation on the ZX plane in R mode, with the y-axis used as the third axis
G02/G03 X_ Y_ R_ F_	
G02/G03 Y_ Z_ R_	
G02/G03 Y_ Z_ R_ F_	
G02/G03 X_ Y_ Z_ R_	
G02/G03 X_ Y_ Z_ R_ F_	

3) The user may monitor the special registers for checking current pulse position.

The following table lists details about 32-bit registers.

X-axis	Y-axis	Z-axis	Attribute
SD36 and SD37	SD136 and SD137	SD236 and SD237	Current position (PLS), only for display purpose
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (mechanical, floating point), only for display purpose
D8340 and D8341	D8360 and D8361	D8380 and D8381	Current position (PLS)

4) Conversion between mechanical unit and pulse unit

In H3U-PM model, if a floating-point number is used to indicate the position function word (XYZ or IJK), it is in a mechanical unit (mm). If an integer is used, it indicates the number of pulses. If a floating-point number is used to indicate the speed function word (F and so on), it is in a mechanical unit (mm/min). If an integer is used, it indicates the frequency, as shown in the following table.

	Floating-point number format	Integer format
Position (XYZ)	X100 indicates 100 (mm).	XKK100 indicates 100 Pls. XDD100 indicates DD100 Pls.
Speed (F)	F60 indicates 60 (mm/min).	FKK200 indicates 200 Hz. FRR200 indicates RR200 Hz.

The conversion ratio shall be set based on the special register. The default value of A is 2000 PLS, and the default value of B is 1000 um.

X-axis	Y-axis	Z-axis	Attribute
SD6 and SD7	SD106 and SD107	SD206 and SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)

$$\text{Mechanical position} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)}} = \text{Number of pulses}$$

$$\text{Mechanical speed} \times \frac{\text{A (number of pulses per cycle)} \times 1000}{\text{B (distance per cycle)} \times 60} = \text{Output frequency}$$

X100 indicates 100 mm. After conversion, the number of pulses is 100 x 2000 x 1000/1000 = 200,000.

F60 indicates 60 mm/min. After conversion, the output frequency is 2000 Hz.

5) Only trapezoid acceleration/deceleration is supported.

6) The acceleration/deceleration time can be set separately, within the range 10 to 500 ms.

The maximum speed, base speed, acceleration/deceleration time, and other parameters of the high-speed output axes can be separately set for each axis.

X-axis	Y-axis	Z-axis	Attribute
SD10 and SD11	SD110 and SD111	SD210 and SD211	Maximum speed (Vmax)
SD12 and SD13	SD112 and SD113	SD212 and SD213	Base speed (starting speed) (Vbias)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)

7) The actual minimum output frequency (that is, the minimum base output frequency) is calculated according to the following formula:

$$\text{Vmin} = \sqrt{\frac{\text{Vset (Hz)}}{2 \times \text{Tacc (ms)} / 1000}}$$

■ Program example

```
G91
G17
G02 X500 Y866.025  R1000
G02 X1500 Y866.025  R1000
G02 X1500 Y-866.025  R-1000
G02 X500 Y-866.025 R-1000

G03 X500 Y-866.025  R1000
G03 X1500 Y-866.025  R1000
G03 X1500 Y866.025 R-1000
G03 X500 Y866.025 R-1000
```

It indicates that, in relative position mode, a 60-degree arc (the data is obtained through calculation), 120-degree arc, 240-degree arc, and 300-degree arc are drawn clockwise relative to the current position on the XY plane, and the arc radius is 1000 mm. A 60-degree arc (the data is obtained through calculation), 120-degree arc, 240-degree arc, and 300-degree arc are drawn counterclockwise, and the arc radius is 1000 mm. The interpolation speed inherits the previous speed.

```
G19
G91
G02 X500.000 Y500.000 Z500.000 J250.000 K250.000 F6000
```

It indicates that, in relative position mode, helix interpolation is performed on x-, y, and x-axes relative to the current position on the YZ plane. YZ arc interpolation is performed. The end point relative to the current position is (500 mm, 500 mm), and the center coordinate relative to the current position is (250 mm, 250 mm). Besides, interpolation is performed on the x-axis to the position 500 mm relative to the current position. The interpolation speed is 6000 mm/min.

7

G04: Delay waiting

◆ Overview

The TIM instruction is used to set the delay before the next motion control instruction is executed.

G04 P_			Delay waiting	Applicable model: H3U-PM	
P	Delay	Delay, in ms			

◆ Operands

Parameter	Bit Element		Word Element										Immediate Operand				
P	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The immediate operand type is not displayed. For example, P10 indicates that P is the integer 10.

◆ Functions and actions

The next instruction can be executed when the specified delay expires. The unit is ms.

Example:

```
G04 P1000
```

It indicates that the delay is 1000 ms.

G90: Absolute position modal; G91: Relative position modal

◆ Overview

The ABST and INCT modal instructions are used to configure the current motion control coordinate system in absolute or relative position mode.

G90	Absolute position modal	Applicable model: H3U-PM
G91	Relative position modal	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

Current running of a motion control subprogram means that the subprogram is always in running state after the SM90 enabling flag is active. The current running is finished when the SM91 complete flag switches to ON. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope. The modal instruction enabled in the current running remains active until the execution is completed or changed.

After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.

G90 and G91 are mutually exclusive modal instructions. The current modal remains unchanged after being enabled until another modal is enabled. The default modal is in absolute position mode on the XY plane.

G17, G18, and G19

◆ Overview

The G17, G18, and G19 modal instructions are used to configure the main plane of the current motion control coordinate system as the XY plane, YZ plane, or ZX plane. They are mainly used for arc and helix interpolation.

G17	Selection of the XY-plane modal instruction	Applicable model: H3U-PM

G18	Selection of the ZX-plane modal instruction	Applicable model: H3U-PM

G19	Selection of the YZ-plane modal instruction	Applicable model: H3U-PM

◆ Operands

None

◆ Functions and actions

Current running of a motion control subprogram means that the subprogram is always in running state after the SM90 enabling flag is active. The current running is finished when the SM91 complete flag switches to ON. If the subprogram calls another motion control subprogram, the called subprogram is also within the current running scope. The modal instruction enabled in the current running remains active until the execution is completed or changed.

After being started, the motion control subprogram is executed in the default modal, and is always active when the current modal remains unchanged.

G17, G18, and G19 are mutually exclusive modal instructions. The current modal remains unchanged after being enabled until another modal is enabled. The default modal is in absolute position mode on the XY plane.

M: Auxiliary parameter number

◆ Overview

The M instruction is used to enable specific auxiliary functions.

M00 – M99	Auxiliary parameter number	Applicable model: H3U-PM

Note: M auxiliary parameter numbers range from 0 to 99. M02, M30, M98, and M99 have been defined; whereas others are reserved.

A G-code subprogram file can have multiple Oxxxx subprograms ranging from O0000 to O9999. Oxxxx indicates the main program, and the remaining numbers indicate subprograms. The Oxxxx main program is determined by the position instead of the serial number. O1000 is used as the Oxxxx main program by default.

M02 and M30: auxiliary parameter numbers

◆ Overview

The M02 and M30 instructions are used to return to the main program or the caller of the previous layer when the Oxxxx main program in the G-code subprogram is finished (that is, the G-code subprogram is finished).

M02	End of the Oxxxx main program	Applicable model: H3U-PM

M30	End of the Oxxxx main program	Applicable model: H3U-PM

M98: Auxiliary parameter number

◆ Overview

The M98 instruction is used to call the specified Oxxxx subprogram continuously for the specified times.

M98 P_ L_		Call of the Oxxxx subprogram	Applicable model: H3U-PM	
P	Number of the O subprogram	Serial number of the subprograms O000 to O9999 called		
L	Number of calls	Number of times the subprogram is called		

◆ Operands

Parameter	Bit Element		Word Element									Immediate Operand				
P	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E
L	M	SM	D	DD	DE	R	RR	RE	SD	SDD	SDE	K	KK	H	HH	E

Note: The elements in gray background are supported. The immediate operand type is not displayed. For example, P10 indicates that P is the integer 10.

L can be omitted, indicating that the subprogram is called once by default. Currently, the number of calls cannot be set, and the subprogram can be called only once.

M99: Auxiliary parameter number

◆ Overview

The M99 instruction is used to return to the Oxxxx main program or the caller of the previous layer when the current Oxxxx subprogram is finished.

M99	Return of the Oxxxx subprogram	Applicable model: H3U-PM

◆ Example:

```
O1000  O1000

       G00 X0 Y0 Z0

       G04 P1000

       M98 P1001
       M02


O1001  O1001
       G91
       G00 X2000
       M99
```

In the G-code subprogram file, O1000 is the main program, and O1001 is the subprogram. In O1000, O1001 is called by M98. When L is omitted, it indicates that the subprogram is called once by default. When O1001 is finished, M99 returns to O1000. When O1000 is finished, M02 or M30 returns to the main program or the caller of the previous layer.

Mxxxx: Element setting parameter number

◆ Overview

The Mxxxx instruction is used to set the M elements. They are used for interaction with the main program or logic control.

M100 – M7679	Auxiliary parameter number	Applicable model: H3U-PM

Note: M elements 100 to 7679 can be set.

◆ Example:

```
G91
G00 X2000
M300
G02 X9999.998 Y9999.998 Z9999.999 R4999.999  F99999.99
```

It indicates M300 is set after quick positioning and then arc interpolation is performed.

S and T: Auxiliary parameter number

◆ Overview

The S and T instructions are used for setting the rotational speed of the master axis.

S0.01 – S99999.99	Major axis rotational speed	Applicable model: H3U-PM

Note: The master axis rotational speed is expressed by a floating-point number in r/min. The range is 0.01 to 99999.99 (two decimals). An error is returned if the value is 0.

Only program uploading/downloading is supported, but the function cannot be enabled.

◆ Overview

These instructions are used to select tools to be used.

T0 – T9999	Tool number	Applicable model: H3U-PM

Note: It indicates the tool number ranging from 0 to 9999.

Only program uploading/downloading is supported, but the function cannot be enabled.

7.10 General Positioning Instructions Supported by the H3U-PM Model

The H3U-PM model supports some positioning instructions whose usage is the same as that in the H3U model. These instructions can be called only in the main program and subprograms, but cannot be called in the motion control subprograms (MC subprograms and G-code subprograms), as shown in the following table.

Instruction	Pulse Direction Output	Trapezoid Acceleration/ Deceleration	S-curve Acceleration/ Deceleration	Separate Acceleration/ Deceleration Setting	Frequency Modification Supported During Running	Pulse Count Modification Supported During Running	Direction Change During Running	Speed OR Position Control
PLSY					√	√ (M)		Speed Position Speed +Position
PLSV	√				√		√	Speed
PLSV2	√	√		√ (M)	√		√	Speed
ZRN		√		√ (M)				Speed
PLSR		√	√ (M)	√ (M)		√ (M)		Position
DRVA	√	√	√ (M)	√ (M)		√ (M)		Position
DRVI	√	√	√ (M)	√ (M)		√ (M)		Position
PLSN	√	√		√ (M)				Position

NOTE

- When the positioning instruction is used in the H3U-PM model, parameters of Y0 are applied to the x-axis, parameters of Y1 are applied to the y-axis, and parameters of Y2 are applied to the z-axis. The direction pin is a dedicated pin, and the instruction parameters can be used to set any elements. For the use of positioning instructions in the H3U-PM model, refer to the use of these instructions in the H3U standard model and special element D.

	H3U-PM Model	H3U-PM Model
Correspondence	X-axis	Y0
Correspondence	Y-axis	Y1
Correspondence	Z-axis	Y2

- The main program (including its subprograms), motion control program (MC subprogram and G-code subprogram), and electronic cam cannot drive the same axis simultaneously. For example, if a motion control subprogram (MC subprogram or G-code subprogram) is executed to perform linear interpolation on the x- and y-axes, and PLSY is executed to drive Y0 to output pulses, an error is returned because the x-axis is used in two cases. In general, an axis can only be driven by one actuator at a time.

7.11 Special Registers for Motion Control in the H3U-PM Model

Special element registers range from SM0 to SM299, as shown in the following table.

X-axis	Y-axis	Z-axis	Attribute
SM0 to 11	SM100 to 111	SM200 to 211	Reserved
SM12	SM112	SM212	Flag of DRVZ zero return direction
SM13	SM113	SM213	Specified flag of the ZRN signal, which is DOG signal by default, or PG signal after setting
SM14 to 16	SM114 to 116	SM214 to 216	Reserved
SM17	SM117	SM217	S-curve acceleration/deceleration enabling flag
SM18	SM118	SM218	Axis origin return disabled
SM19	SM119	SM219	Reserved
SM20	SM120 (reserved)	SM220 (reserved)	Flag of enabling continuous interpolation
SM21 to 69	SM121 to 169	SM221 to 269	Reserved
SM70	SM170	SM270	Axis trigger mode selection for an electronic cam OFF: Software trigger; ON: Hardware trigger
SM71	SM171	SM271	Axis input source selection for an electronic cam OFF: Internal virtualization; ON: External input
SM72	SM172	SM272	Synchronization of x-, y-, and z-axes for an electronic cam OFF: Disabled; ON: Enabled
SM73	SM173	SM273	Cyclic execution of an electronic cam OFF: No; ON: Yes
SM74	SM174	SM274	External hardware stop OFF: Disabled; ON: Enabled
SM75	SM175	SM275	Startup delay enabling for an electronic cam OFF: Disabled; ON: Enabled
SM76	SM176	SM276	Left limit enabling OFF: Disabled; ON: Enabled
SM77	SM177	SM277	Right limit enabling OFF: Disabled; ON: Enabled
SM78	SM178	SM278	Electronic cam enabling OFF: Disabled; ON: Enabled
SM79	SM179	SM279	Cam cycle end flag OFF: Unfinished; ON: Finished
SM80	SM180	SM280	Electronic cam/gear stop flag OFF: Unfinished; ON: Finished

X-axis	Y-axis	Z-axis	Attribute
SM81	SM181	SM281	Stop mode selection OFF: Stop after the current cycle ON: Immediate stop
SM82	SM182 (reserved)	SM282 (reserved)	Electronic cam modification complete flag
SM83	SM183	SM283	Key point modification mode selection for an electronic cam OFF: Effective upon restart ON: Effective during the next cam cycle
SM84 to 88	SM184 to 188	SM284 to 288	Reserved
SM89	SM189	SM289	Initialization complete flag OFF: Initialization started ON: Initialization completed
SM90	SM190 (reserved)	SM290 (reserved)	Motion control subprogram MCX enabling flag
SM91	SM191 (reserved)	SM291 (reserved)	Motion control subprogram MCX execution complete flag
SM92 to 99	SM192 to 199	SM292 to 299	Reserved

Special element registers range from SD0 to SD299, as shown in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD0	SD100	SD200	Reserved
SD1	SD101	SD201	Reserved
SD2	SD102	SD202	Reserved
SD3	SD103	SD203	Reserved
SD4	SD104	SD204	Reserved
SD5	SD105	SD205	Reserved
SD6, SD7	SD106, SD107	SD206, SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor rotates a circle (B)
SD10 and SD11	SD110 and SD111	SD210 and SD211	Maximum speed (Vmax)
SD12 and SD13	SD112 and SD113	SD212 and SD213	Base speed (starting speed) (Vbias)
SD16 and SD17	SD116 and SD117	SD216 and SD217	Zero return speed (VRT)
SD18 and SD19	SD118 and SD119	SD218 and SD219	Zero return creep speed (VCR)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)
SD22	SD122	SD222	Number of PG signals (N)
SD23	SD123	SD223	Number of pulses for zero return (P), pulse offset in case of a DOG signal
SD24 and SD25	SD124 and SD125	SD224 and SD225	Home position (HP)
SD26 and SD27	SD126 and SD127	SD226 and SD227	Electrical origin position
SD28 and SD29	SD128 and SD129	SD228 and SD229	Target position I (P [I])
SD30 and SD31	SD130 and SD131	SD230 and SD231	Running speed I (V [I])
SD32 and SD33	SD132 and SD133	SD232 and SD233	Target position II (P [II])
SD34 and SD35	SD134 and SD135	SD234 and SD235	Running speed II (V [II])
SD36 and SD37	SD136 and SD137	SD236 and SD237	Current position (CP [PLS])
SD38 and SD39	SD138 and SD139	SD238 and SD239	Current speed (CS [PPS])
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (CP [mechanical and floating-point])
SD42 and SD43	SD142 and SD143	SD242 and SD243	Current speed (CS [mechanical and floating-point])
SD44	SD144	SD244	Electronic gear ratio numerator

X-axis	Y-axis	Z-axis	Attribute
SD45	SD145	SD245	Electronic gear ratio denominator
SD46 and SD47	SD146 and SD147	SD246 and SD247	Current input frequency
SD48 and SD49	SD148 and SD149	SD248 and SD249	Cumulative number of pulses input by hand gear
SD50 and SD51	SD150 and SD151	SD250 and SD251	Axis offset compensation value (DRV, LIN, and INTR)
SD52 and SD53	SD152 and SD153	SD252 and SD253	Axis center coordinate offset compensation value (CW and CCW)
SD54 and SD55	SD154 and SD155 (reserved)	SD254 and SD255 (reserved)	Axis arc radius offset compensation value (CW and CCW)
SD56 to 59	SD156 to 159	SD256 to 259	Reserved
SD60	SD160	SD260	Setting of high-speed pulse input and count
SD61	SD161	SD261	High-speed pulse output setting
SD62	SD162	SD262	Display of the status of special PM input point
SD63	SD163	SD263	Display of the status of special PM output point
SD64 to 69	SD164 to 169	SD264 to 269	Reserved
SD70	SD170	SD270	Electronic cam axis selection table: 0: Cam and hand gear disabled by default; 10: Hand gear; 11: Cam 1; 12: Cam 2; 13: Cam 3
SD71	SD171	SD271	Setting of electronic cam input axis numbers
SD72	SD172	SD272	Times of non-cyclic cam execution
SD73	SD173	SD273	Reserved
SD74 and SD75	SD174 and SD175	SD274 and SD275	Lower limit of cam synchronization position
SD76 and SD77	SD176 and SD177	SD276 and SD277	Upper limit of cam synchronization position
SD78 and SD79	SD178 and SD179	SD278 and SD279	Number of pulses (startup delay)
SD80	SD180	SD280	Selection of the input pole
SD81 and SD82	SD181 and SD182	SD281 and SD282	Number of finished cam cycles
SD83 to SD89	SD183 to SD189	SD283 to SD289	Reserved
SD90	SD190 (reserved)	SD290 (reserved)	Motion control subprogram MCX marker setting register
SD91 to SD99	SD191 to SD199	SD291 to SD299	Reserved

- (SD6, SD7), (SD106, SD107), and (SD206, SD207): Number of pulses required when the motor rotates a circle (A)

If A is the number of pulses needed for the motor to rotate one circle and F is the electronic gear ratio inside the servo, then A x F = the number of pulses generated when the encoder rotates one circle.

- (SD8, SD9), (SD108, SD109), and (SD208, SD209): Movement distance when the motor rotates a circle (B)

It indicates the movement distance when the motor rotates a circle, in the unit of um or 0.001° (mechanical unit).

- (SD10, SD11), (SD110, SD111), and (SD210, SD211): Maximum speed (Vmax)

1. It is the upper limit of the speeds in different operation modes, ranging from 0 to 2,147,483,647.

2. The maximum speed is 500 kHz. If the set value is larger than 500 kHz, 500 kHz is used by default.

- (SD12, SD13), (SD112, SD113), (SD212, SD213): Start speed (Vbias)

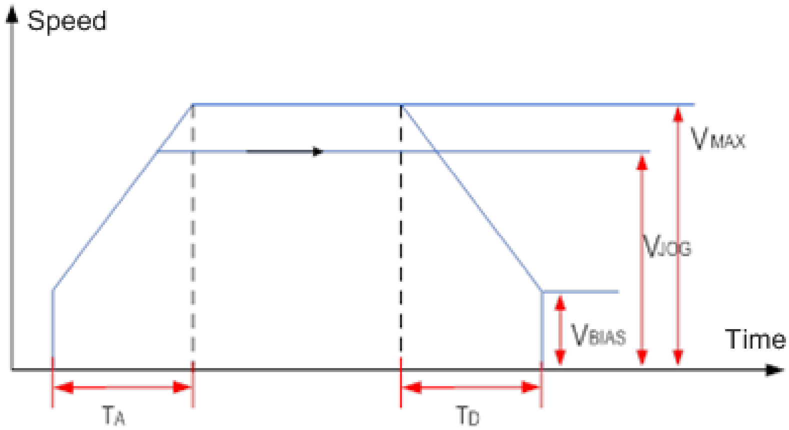
It indicates the start speed of pulse output, ranging from 0 to 2,147,483,647.

- (SD16, SD17), (SD116, SD117), and (SD216, SD217): Zero return speed (VRT)

1. It indicates the zero return speed ranging from 0 to 2,147,483,647.

2. The setting range is Vmax > VRT > Vbias.

- (SD18, SD19), (SD118, SD119), and (SD218, SD219): Origin



regression creep speed (VCR)

1. It indicates the zero return creep speed ranging from 0 to 2,147,483,647.

2. VRT must be larger than VCR.

3. When zero return is executed, pulses are output at VRT. When a near point (DOG) signal is detected, the motor decelerates to VCR.

- (SD20), (SD120), and (SD220): Acceleration time (Vacc)

1. It indicates the acceleration time in ms.

2. The value ranges from 0 to 32,767. If 0 is set, pulses are output at the base speed without acceleration.

- (SD21), (SD121), and (SD221): Deceleration time (Vdec)

1. It indicates the deceleration time in ms.

2. The value ranges from 0 to 32,767. If 0 is set, Vdec equals Vacc.

- (SD22), (SD122), and (SD222): Number of PG signals (N)

The value range is –32,768 to +32,767 (pulse). The positive number indicates N in the forward direction, and negative number indicates N in the reverse direction.

- (SD23), (SD123), and (SD223): Number of pulses of zero return (p)

The value range is –32,768 to +32,767 (pulse). The positive number indicates the P in the forward direction, and negative number indicates the P in the reverse direction.

- (SD24, SD25), (SD124, SD125), and (SD224, SD225): Origin position (HP)

1. The value range is 0 to ±999,999.
2. When the zero return is finished, CP is updated to HP.

- (SD26, SD27), (SD126, SD127), and (SD226,SD227): Electrical origin position

1. The value range is 0 to ±999,999.
2. When zero return is finished, CP is updated to HP.

- (SD28, SD29), (SD128, SD129), and (SD228, SD229): Target position I (P [I])

1. The value range is –2,147,483,648 to +2,147,483,647.

- (SD30, SD31), (SD130, SD131), and (SD230, SD231): Running speed I (V [I])

1. The value range is –2,147,483,648 to +2,147,483,647.
2. The setting range is V max > V(I) > Vbias.
3. When V (I) is positive, the motor runs in the forward direction; when V (I) is negative, the motor runs in the reverse direction.

- (SD32, SD33), (SD132, SD133), and (SD232, SD233): Target position II (P [II])

1. The value range is –2,147,483,648 to +2,147,483,647.

- (SD34, SD35), (SD134, SD135), and (SD234,SD235): Running speed II (V [II])

1. The value range is –2,147,483,648 to +2,147,483,647.
2. The setting range is Vmax > V(II) > Vbias.

- (SD36, SD37), (SD136, SD137), and (SD236, SD237): Current position (CP [PLS])

1. The value range is –2,147,483,648 to +2,147,483,647, and the unit is pulses.
2. When zero return is finished, CP is updated to the number of pulses set by HP.

- (SD38, SD39), (SD138, SD139), and (SD238, SD239): Current speed (CS [PPS])

1. The value range is –2,147,483,648 to +2,147,483,647, and the unit is pps.

- (SD40, SD41), (SD140, SD141), and (SD240, SD241): Current mechanical position (CP [UNIT])

1. The current position is displayed in mm.
2. When zero return is finished, CP is updated to the number of pulses set by HP.

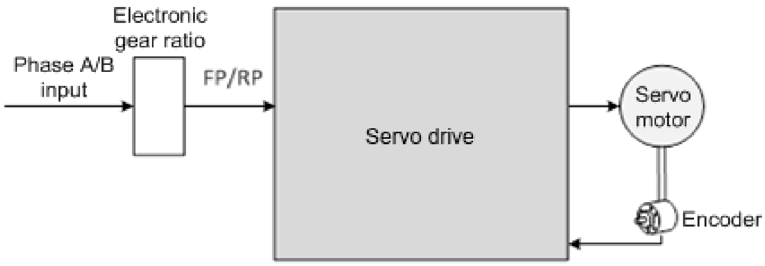
- (SD42, SD43), (SD142, SD143), and (SD242, SD243): Current mechanical speed (CS [UNIT])

The current speed is displayed in mm/min.

- (SD44), (SD144), and (SD244): Electronic cam numerator

- (SD45), (SD145), and (SD245): Electronic cam denominator

1. The hand gear is used to output A/B phase pulses to Ax+, Ax - (x = 0,1,2), Bx+, Bx - (x = 0,1,2). The relationship between input pulse and output pulse is shown in the following figure.



2. During running, if LSP or LSN is enabled, pulse output stops immediately. If LSP is enabled, pulses in forward direction are prohibited, and pulses in reverse direction are allowed. If LSN is enabled, the pulses in reverse direction are prohibited, and pulses in forward direction are allowed.

3. The output running speed indicates the relationship between the pulse input frequency of the hand gear and the electronic gear ratio.

- (SD46,SD47),(SD146,SD147),(SD246,SD247): Current input frequency

The current input frequency is displayed, and the value is not affected by the electronic gear ratio.

- (SD48,SD49),(SD148,SD149),(SD248,SD249): Cumulative number of pulses input by hand gear

1. The accumulative number of pulses input by hand gear is not affected by the electronic gear ratio.

2. For pulse input in forward direction, the count is added; and for pulse input in reverse direction, the count is subtracted.

7

- (SD50,SD51),(SD150,SD151),(SD250,SD251): Axis offset compensation

It indicates the axis offset compensation (DRV, LIN, and INTR).

- (SD52,SD53),(SD152,SD153),(SD252,SD253): Axis center coordinate offset compensation value (CW and CCW)

It indicates the axis center coordinate offset compensation value (CW and CCW).

- (SD54,SD55),(SD154,SD155),(SD254,SD255): Axis arc radius coordinate offset compensation value

It indicates the axis center coordinate offset compensation value (CW and CCW).

- (SD60),(SD160),(SD260): High-speed pulse input and count setting (including the hand-gear-base pulse input mode)

Input Mode Fed Back by Actual Position	
0	Pulse+Direction
1	Phase A/B
2	CW/CCW
Others	Reserved

Note: To switch the input mode, settings must be completed before the function (for example, high-speed counting instruction and electronic cam) is enabled.

- (SD61),(SD161),(SD261): High-speed pulse output setting

Pulse Output Mode	
0	Pulse+Direction
1	Phase A/B
2	CW/CCW
Others	Reserved

Note: To switch the output mode, settings must be completed before the function (for example, high-speed output instruction and motion control instruction) is enabled.

- (SD62),(SD162),(SD262): Input point status display

Bits of elements are defined in the following table.

b0	Enter the A-phase signal status 0: OFF 1: ON
b1	Enter the B-phase signal status 0: OFF 1: ON
b2	Enter the START signal status 0: OFF 1: ON
b3	Enter the DOG signal status 0: OFF 1: ON
b4	Enter the STOP signal status 0: OFF 1: ON
b5	Enter the LSN signal status 0: OFF 1: ON
b6	Enter the LSP signal status 0: OFF 1: ON
b7	Enter the PG signal status 0: OFF 1: ON
b15 to b8	Reserved

- (SD63),(SD163),(SD263): Output point status display

Bits of elements are defined in the following table.

b0	Output the CLR signal status 0: OFF 1: ON
b1	Reserved
b2	Reserved
b3	Reserved
b4	Reserved
b5	Reserved
b6	Reserved
b7	Reserved
b15 to b8	Reserved

- (SD78,SD79),(SD178,SD179),(SD278,SD279): Element for setting the number of pulses (delayed startup)

The set value takes effect when SMX75 delayed startup is enabled.

The value range is 0x0 to 0xFFFFFF.

- (SD80),(SD180),(SD280): Input pole selection

Bits of this element are defined in the following table.

b0	Enter the A signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b1	Enter the B signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b2	Enter the START signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b3	Enter the DOG signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b4	Enter the STOP signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b5	Enter the LSN signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b6	Enter the LSP signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b7	Enter the PG signal polarity 0 indicates the positive polarity, which is valid when ON is entered. 1 indicates the negative polarity, which is valid when OFF is entered.
b15 to b8	Reserved

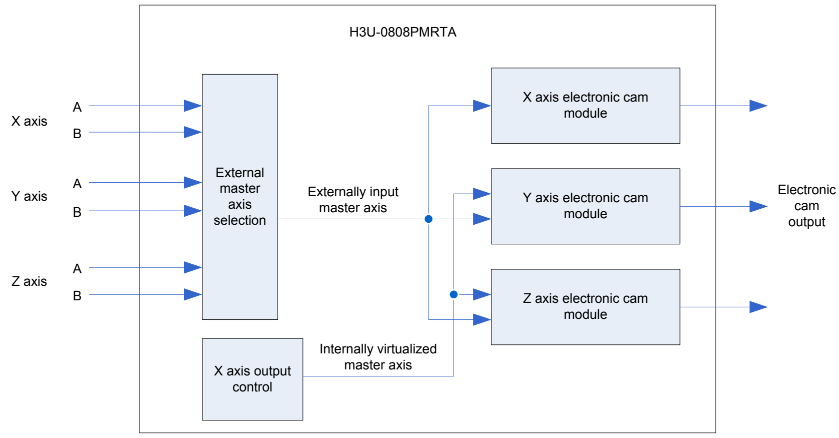
- (SD81,SD82),(SD181,SD182),(SD281,SD282): Finished electronic cam cycles

It counts the finished electronic cam cycles. The start number is 0 when the electronic cam/gear starts. The count is increased by 1 each time a cycle is finished. When the electronic cam/gear stops, the count remains unchanged.

8 Electronic Cam

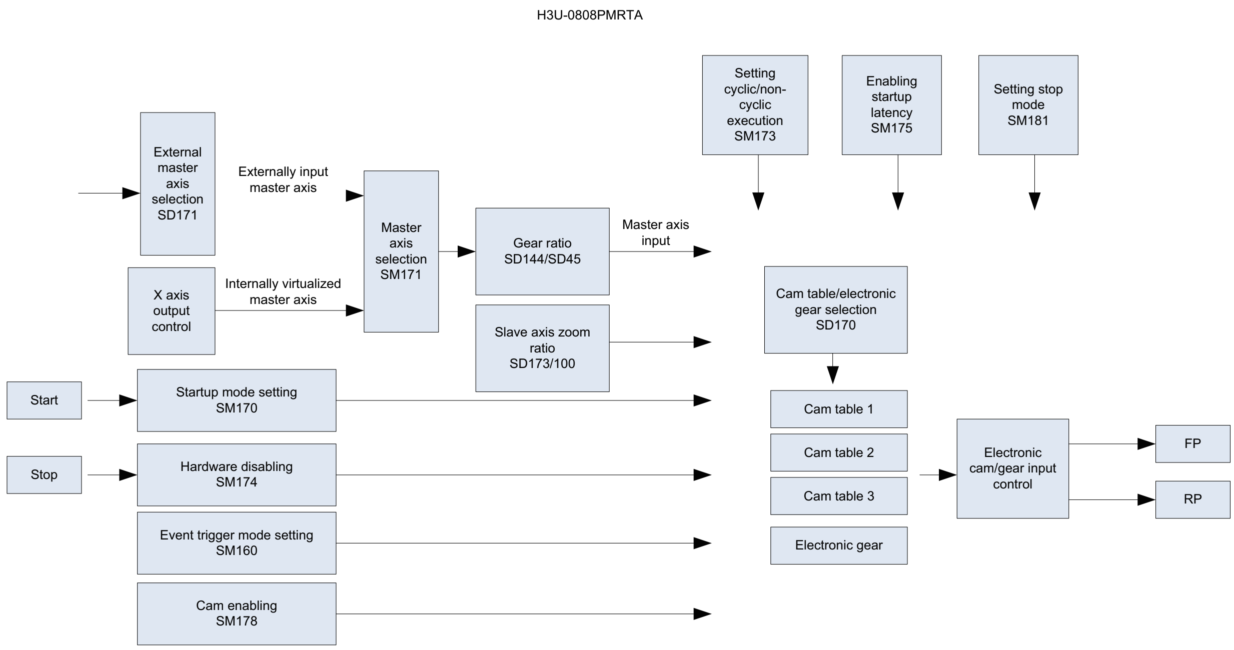
Chapter 8 Electronic Cam

Two models of H3U PLCs are available: common H3U PLC and H3U-PM motion control PLC. The H3U-PM motion control PLC model has 3-axes electronic cams and hand gears (also called electronic gears). The following figure shows the structure of the 3-axes electronic cam module of an H3U-PM motion control PLC:



The 3-axes electronic cam module supports electronic cam table following or electronic gears. A master axis can be externally input or internally virtualized. The X, Y, or Z axis can be input at a high speed as a master axis of the electronic cam module. The X axis is internally virtualized as a master axis. Therefore, only the Y and Z axes can use the internally virtualized axis.

Electronic cam modules based on the three axes share the same basic functions. With an example of the Y axis, the following figure shows the basic structure of a single-axis electronic cam module.



Operations of electronic cams:

① Establishing a cam table

② Selecting a master axis

③ Setting cyclic or non-cyclic execution

④ Starting electronic cams

⑤ Stopping electronic cams

⑥ Dynamically modifying electronic cam data

8.1 Establishing a Cam Table

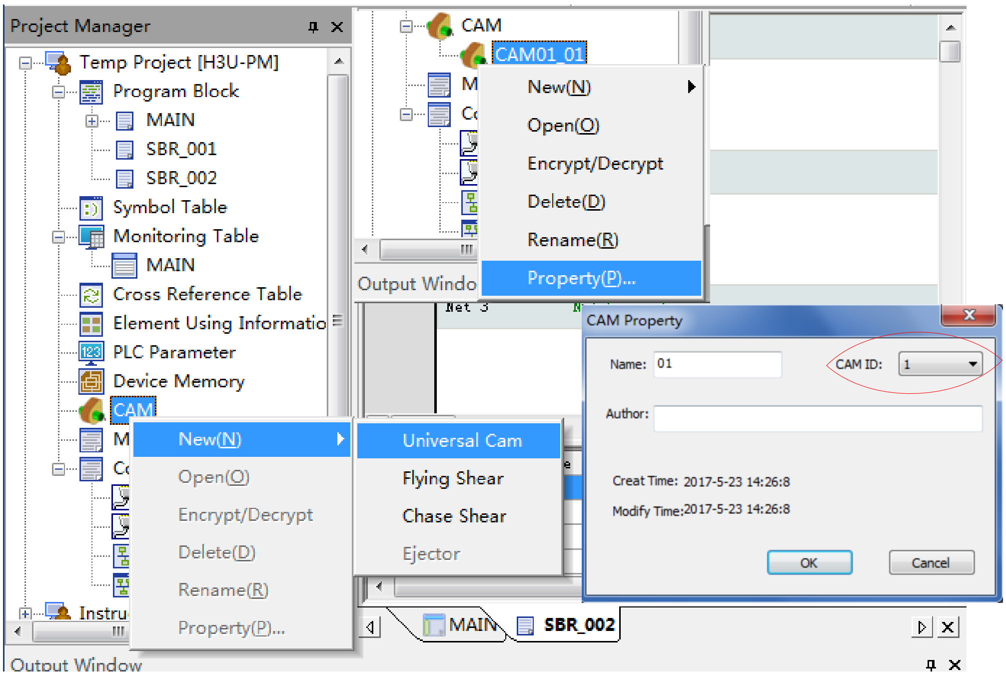
The slave axis is driven by the master axis. The relationship between master and slave axes can be indicated by cam table data or electronic gear ratio. You can use cam table data to establish a maximum of 360 key points. The electronic gear ratio of master to slave axis is fixed.

If electronic gears are used, you only need to set the numerator and denominator without setting the cam table. If electronic cams are used, you need to set the electronic cam table first.

8.1.1 Creating a Cam Table

On the **Project Manager** page of AutoShop, right-click **CAM** and select **New** to create a cam table. You can set a maximum of 16 cam tables by using AutoShop. Three cam tables can be downloaded to the PLC, and the others are saved only in project files.

Cam tables downloaded to the PLC can be identified by IDs. Right-click the icon of a created cam table and select **Property** to display or set the ID. IDs 1, 2, and 3 correspond to cam tables 1, 2, and 3 respectively. These cam tables can be downloaded to the PLC. IDs of the others are -1. You can change IDs in the cam property dialog box.



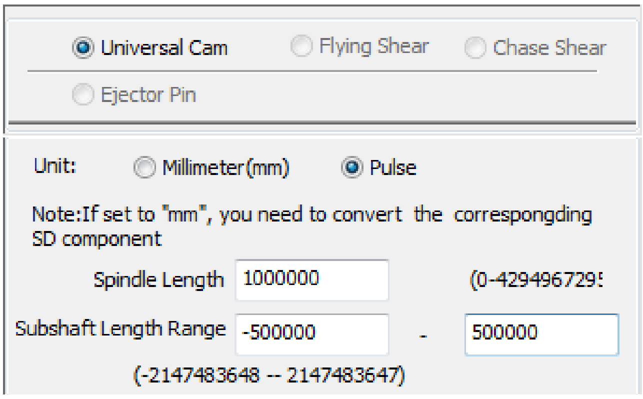
8

8.1.2 Editing a Cam Table

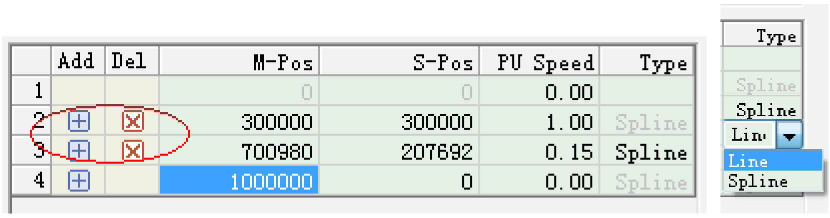
Double-click the icon of a created cam table to open the key point editing page.

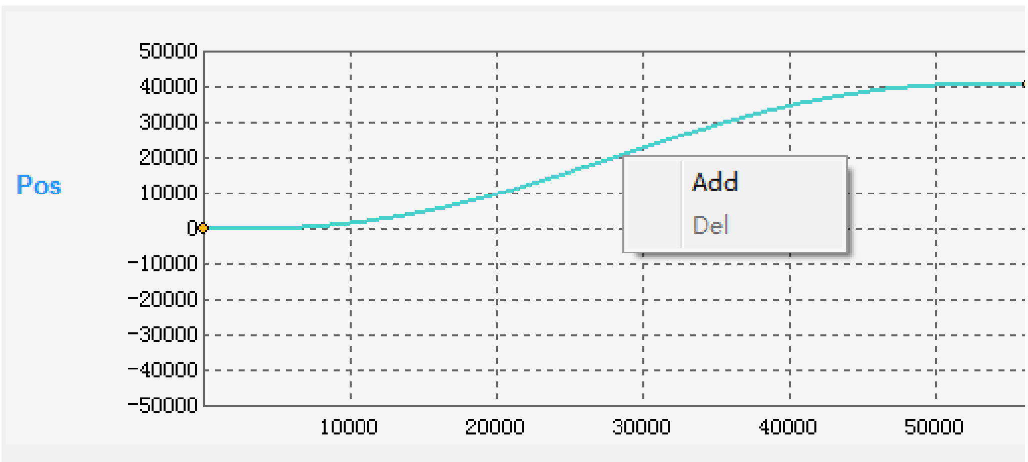
On the key point editing page, select master axis length and slave axis range based on the selected unit. The master axis length is the travel distance of a master axis per cam cycle. The slave axis range indicates the stroke of a slave axis for display of graphics and easy editing.

When the unit is millimeter, the master axis length ranges from 0 to 100,000 mm, and the slave axis ranges from –100,000 to +100,000. When the unit is number of pulses, the master axis length ranges from 0 to 4,294,967,296, and the slave axis ranges from –2,147,483,648 to +2,147,483,647.

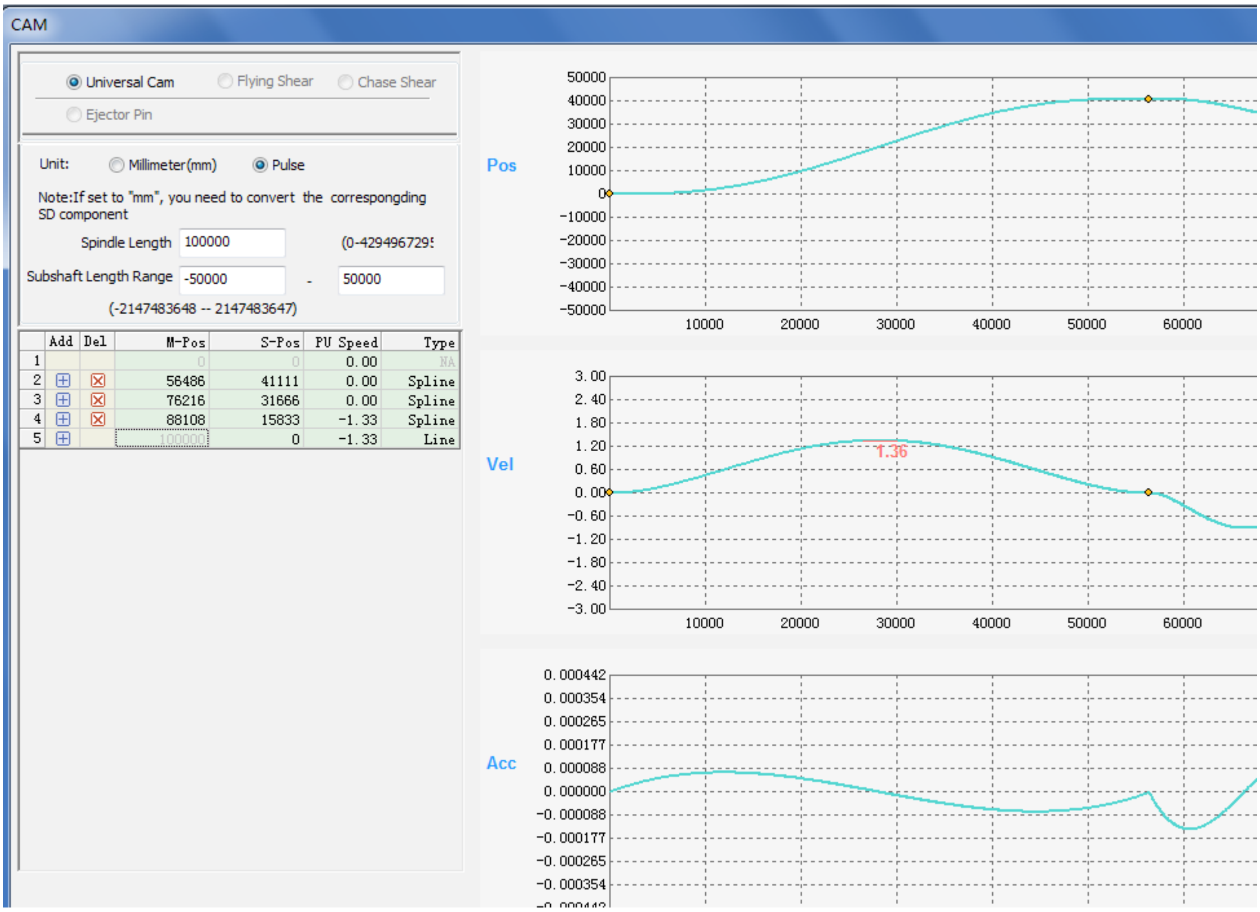


After the master axis length and slave axis range are set, you can set key points. As shown in the following figure, you can click the **Add** or **Del** icon to add or delete key points respectively, or right-click the displacement diagram to select **Add** or **Del**. You can set a maximum of 360 key points for each cam. In the table, you can set the master axis position (M-Pos), slave axis position (S-Pos), speed ratio (PU speed), and type of curve between key points (Type). The curve type can be interpolated and fitted by splines or lines for five times. In addition, you can drag key points in the displacement or speed ratio diagram to adjust the relationship between position and speed.



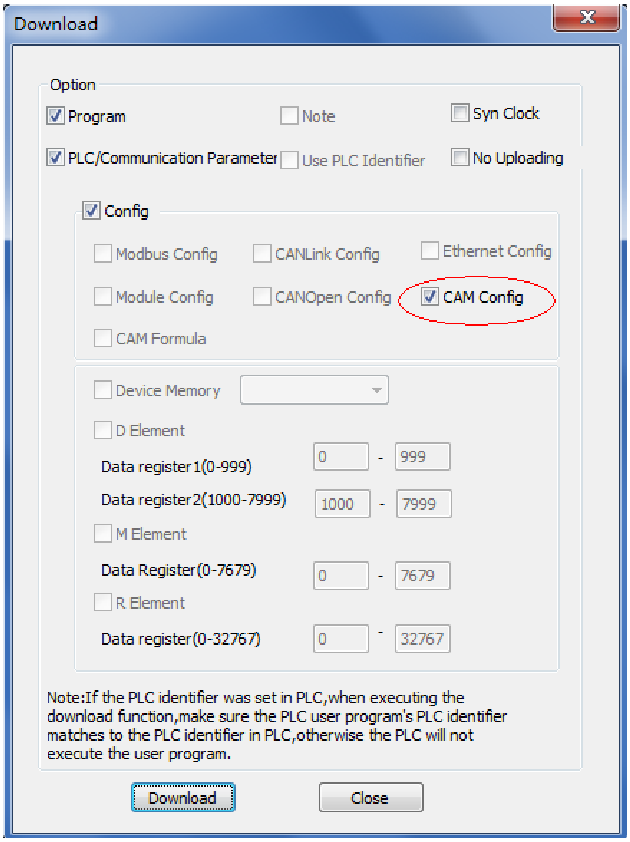


8



8.1.3 Downloading Cam Tables

Save the settings and exit. Select **CAM Config** to download cam tables 1, 2, and 3 to the PLC.



8

8.2 Selecting a Master Axis

Signals of a master axis are needed to enable electronic cams or gears. The signals can be generated through external input or internal virtual connection.

The following table lists special elements for master axis selection.

SM Element for Master Axis Selection			SD Element for Master Axis Selection		
X axis	Y axis	Z axis	X axis	Y axis	Z axis
SM71	SM171	SM271	SD71	SD171	SD271
OFF: Internal virtual connection			1: Internal connection to X output axis		
ON: External input			1: X input channel 2: Y input channel 3: Z input channel		

If internal virtual connection is selected, the X output axis will be virtually controlled, and the master axis will be internally virtualized for an electronic cam or gear without external connection. When the X axis serves as a slave axis of an electronic cam, internal virtual connection is unavailable.

If external input is selected, based on values of SD elements, you can select X, Y, or Z input channel as the master axis. If signals of a master axis are externally input, modes of external input (SD60, SD160, and SD260) must match input signals.

8.3 Setting Cyclic or Non-cyclic Execution

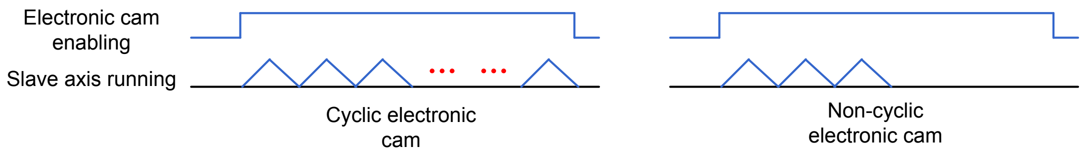
You can use special SM or SD elements to set cyclic or non-cyclic execution for electronic cams.

The following table lists special elements for setting cyclic or non-cyclic execution.

SM for Setting Cyclic or Non-cyclic Execution			SD for Setting Number of Cycles		
X axis	Y axis	Z axis	X axis	Y axis	Z axis
SM73	SM173	SM273	SD72	SD172	SD272
OFF: Non-cyclic execution			You can set a maximum of 255 cycles for non-cyclic execution.		
ON: Cyclic execution			N/A		

Cyclic execution: The electronic cam cyclically executes relationships set in the electronic cam table until it receives the command to stop.

Non-cyclic execution: The electronic cam automatically stops after a specified cycles of execution. SD elements (SD72, SD172, and SD272) can be used to set the number of cycles (maximum: 255) for non-cyclic execution.



8.4 Starting Electronic Cams or Electronic Gears

8.4.1 Selecting Cam Tables or Electronic Gears

You can set different cam tables and select SD element values to select different cam tables or electronic gears.

The following table lists special elements for cam table selection.

Cam Table Selection		
X axis	Y axis	Z axis
SD70	SD170	SD270

The following table shows the relationship between element value and cam table.

Value of SD Element for Cam Table Selection	Description
10	Electronic gear
11	Cam table 1 (ID 1)
12	Cam table 2 (ID 2)
13	Cam table 3 (ID 3)
Other	The command is not executed, and error 16262 is returned.

8.4.2 Startup

Electronic cams or electronic gears can be enabled by software or triggered by events:

Software: You can enable cams by an OFF-to-ON switch of SM elements for cam enabling.

Event trigger: You can trigger cams by external inputs or comparison interrupts. When SM elements for cam enabling are ON, cams can be triggered by an OFF-to-ON switch of external start signals or by comparison interrupts.

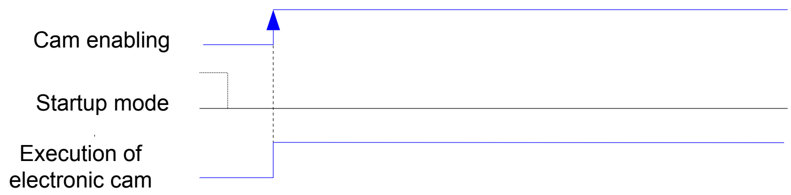
The following table lists SM elements for cam enabling.

Cam Enabling		
X axis	Y axis	Z axis
SM78	SM178	SM278

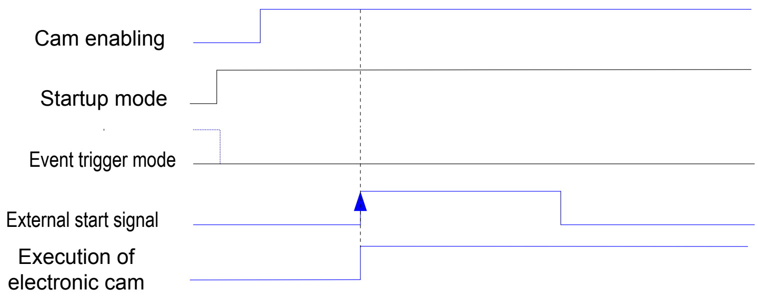
The following table lists SM elements for cam startup modes.

Startup Mode Setting			Event Trigger Mode Setting		
X axis	Y axis	Z axis	X axis	Y axis	Z axis
SM70	SM170	SM270	SM60	SM160	SM260
OFF: Software			N/A		
ON: Event trigger			OFF: External input ON: Comparison interrupt		

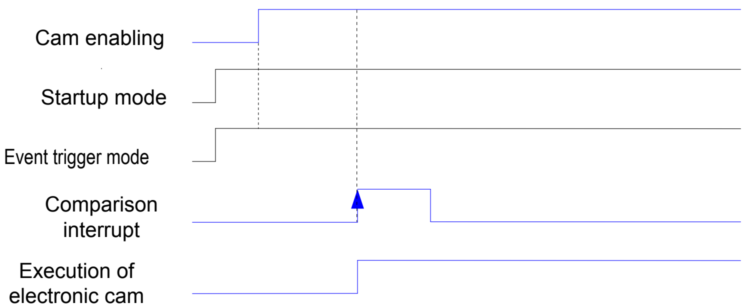
Timing for startup by software:



Timing for trigger by external inputs:



Timing for trigger by comparison interrupts:



8.4.3 Setting Startup Latency

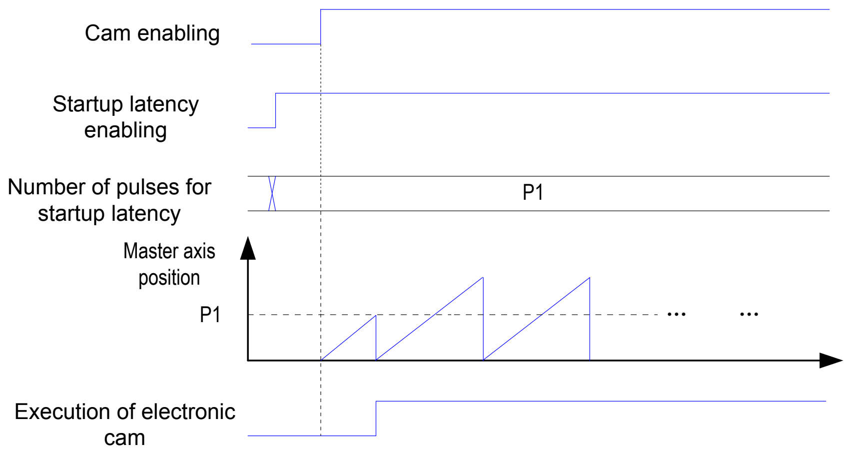
You can enable startup latency of electronic cams or electronic gears based on the settings. After startup latency is enabled by software or triggered by events, electronic cams are executed after the number of pulses of the master axis is set.

The following table lists special elements for setting startup latency.

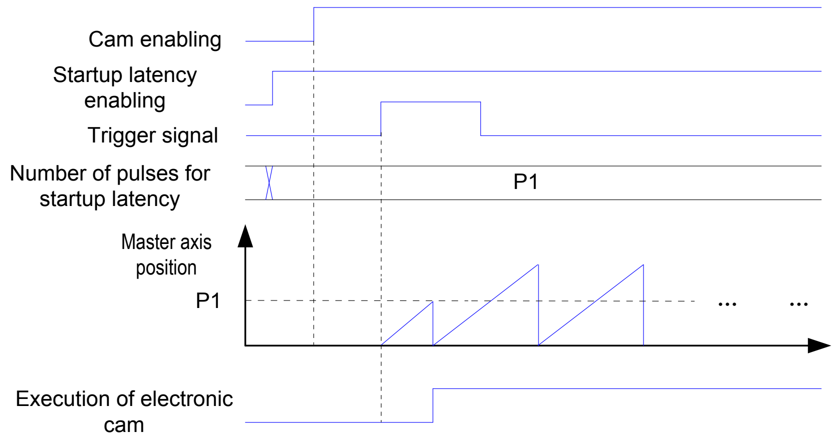
Startup Latency Enabling			Number of Pulses for Startup Latency		
X axis	Y axis	Z axis	X axis	Y axis	Z axis
SM75	SM175	SM275	SD78 and SD79	SD178 and SD179	SD278 and SD279
OFF: Startup latency disabled			N/A		
ON: Startup latency enabled			You can set a maximum of 1,000,000 pulses for startup latency.		

8

Startup latency in software startup mode:



Startup latency in event trigger mode:



8.4.4 Using Comparison Interrupt

You can use comparison instructions to trigger electronic cams or electronic gears by comparison interrupts.

DHSCS and DHSOS comparison instructions can be used to trigger electronic cams or electronic gears.

DHSCS is an instruction for high speed counter comparison interrupts. It compares a high speed counter with the counter of the input channel and thereby generates an interrupt.

DHSOS is an instruction for high speed interrupts. It compares the position of the high speed output axis with the current position of the output axis and thereby generates an interrupt.

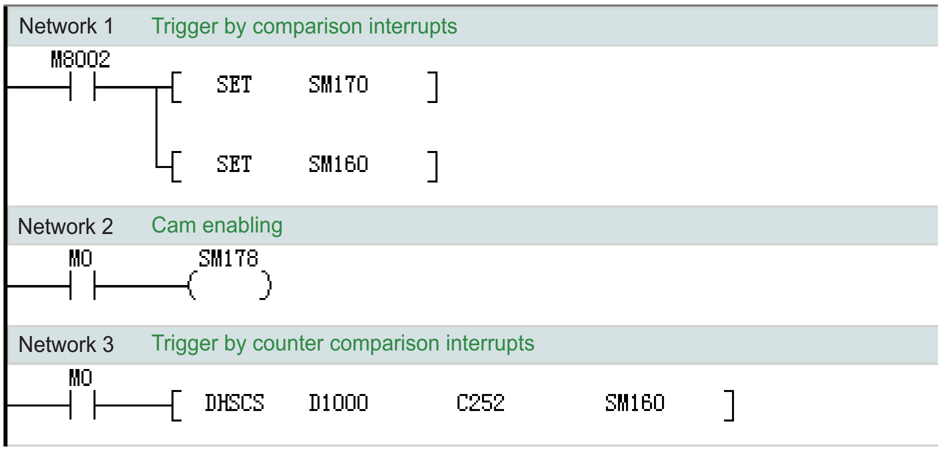
For information on usage and parameters of the preceding instructions, see "5.4.2 HSCS Comparison Setting" on page 358 and "5.4.5 DHSOS High-speed Interrupt Comparison Setting" on page 370.

The following conditions must be met to trigger electronic cams or electronic gears by comparison interrupts:

- When SM elements for the startup mode are set to ON, electronic cams or gears will be triggered by events.
- When SM elements for the event trigger mode are set to ON, electronic cams or gears will be triggered by comparison interrupts.
- SM elements for cam enabling must be ON.
- The DHSCS or DHSOS instruction specifies SM elements in event trigger mode.

8

Application:



8.5 Stopping Electronic Cams or Electronic Gears

8.5.1 Setting Stop Mode

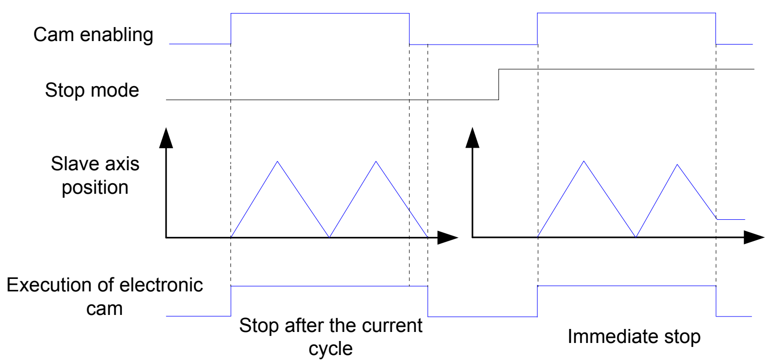
You can use special SM elements to set the stop mode for electronic cams: a stop after the current cycle or an immediate stop.

The following table lists special elements for setting the stop mode.

Stop Mode Setting		
X axis	Y axis	Z axis
SM81	SM181	SM281
OFF: Stop after the current cycle ON: Immediate stop		

Stop after the current cycle: When an electronic cam is disabled or a stop signal is valid, the electronic cam stops after the current execution cycle is finished.

Immediate stop: When an electronic cam is disabled or a stop signal is valid, the electronic cam stops immediately.



8.5.2 Setting an Externally Triggered Stop

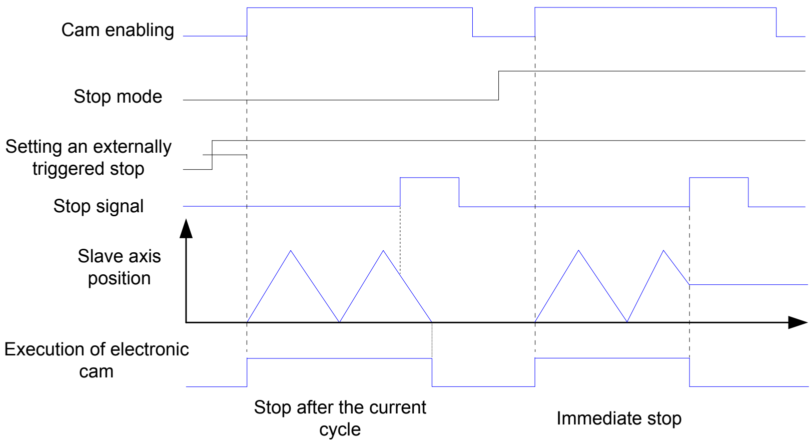
You can stop electronic cams by external stop signals based on the settings.

The following table lists special elements for setting an externally triggered stop.

8

Setting an Externally Triggered Stop		
X axis	Y axis	Z axis
SM74	SM174	SM274
OFF: Externally triggered stop disabled ON: Externally triggered stop enabled		

If an externally triggered stop is enabled, the electronic cam stops based on stop mode settings upon an OFF-to-ON switch of an external stop signal.



8.5.3 Stop

You can stop electronic cams or electronic gears in two ways:

- Turn off special elements for cam enabling.
- Set an externally triggered stop to stop electronic cams or gears by an OFF-to-ON switch of external stop signals.

In both ways, electronic gears can be immediately stopped, while electronic cams can be stopped after the current cycle or immediately, depending on stop mode settings.

8.5.4 Stopping Electronic Cams

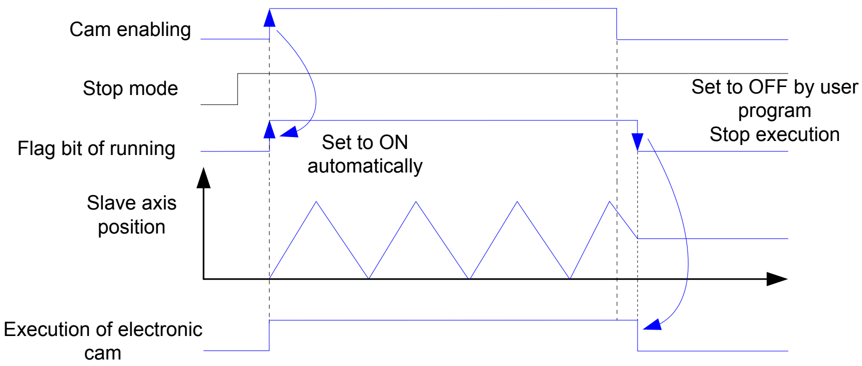
If the stop mode for electronic cams is set to "stop after the current cycle", when you stop electronic cams in either of the ways described in Section 8.5.3, they will stop after the current cycle is finished. To immediately stop electronic cams, you can use special SM elements.

The following table lists special elements for stopping electronic cams.

Flag of Running/Stopping		
X axis	Y axis	Z axis
SM89	SM189	SM289
ON: Electronic cam running OFF: Stopping electronic cam		

When a cam is enabled, the system initializes cam data. After initialization, the flag bit of running is automatically set to ON.

When the stop mode is set to "stop after the current cycle", the cam is disabled. To stop the cam immediately, you can set the flag bit of running to OFF.



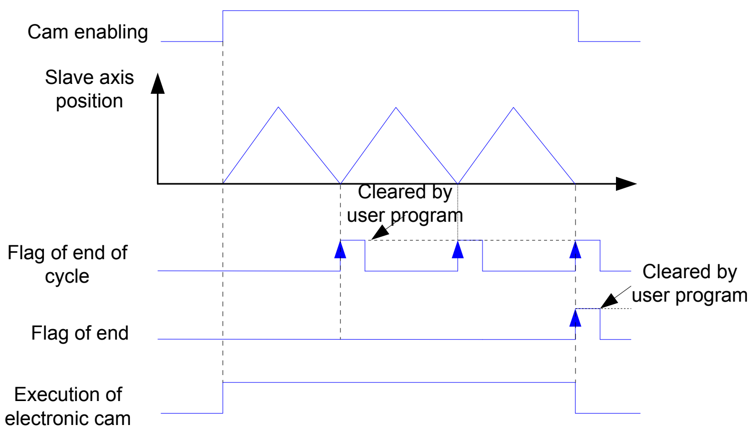
8.5.5 Flag of End of Cycle and Flag of End

When a cam cycle is finished, the system will automatically set to ON special SM elements for the flag of end of cycle. The flag of end of cycle remains ON. To detect end of the next cycle, you need to use the user program to set the flag of end of cycle to OFF. When the next cycle is finished, the system will set the flag to ON.

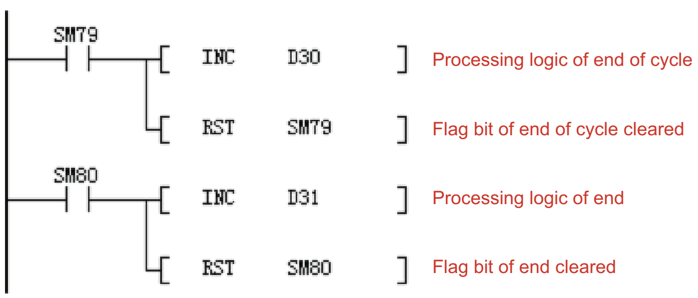
When execution of electronic cams or electronic gears ends, the system will automatically set to ON special SM elements for the flag of end. The flag of end is set to OFF by the system or user program when the cam is enabled.

The following table lists special SM elements for flag of end of cycle and flag of end.

	X Axis	Y Axis	Z Axis	Description
Flag of End of Cycle	SM79	SM179	SM279	It is set to ON when a cam cycle is finished.
Flag of End	SM80	SM180	SM280	It is set to ON when execution of electronic cams or electronic gears ends.

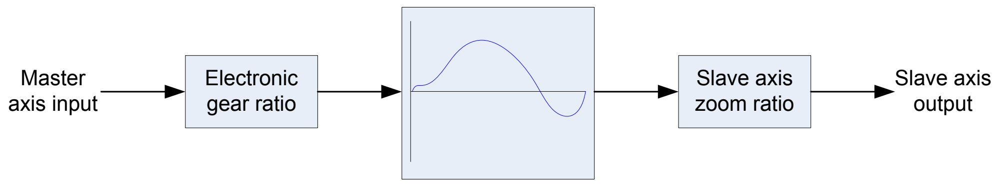


Example:

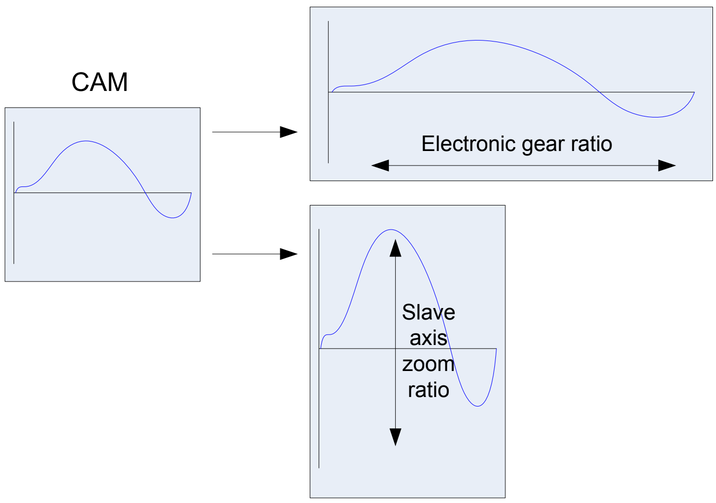


8.6 Scaling

Cam tables can be scaled.



Master and slave axes can be scaled by setting special SD elements for electronic gear ratio and slave axis zoom ratio.



The following table lists special elements for electronic gear ratio and slave axis zoom ratio.

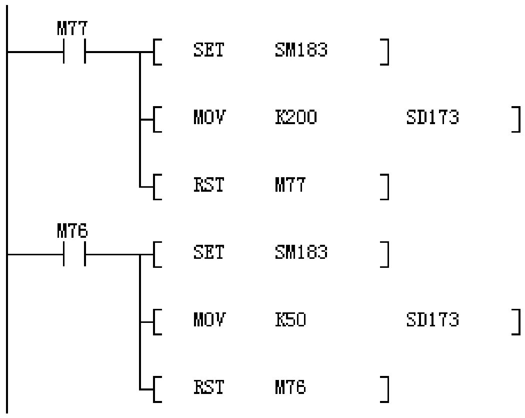
Electronic Gear Ratio/Master Axis Zoom Ratio			Slave Axis Zoom Ratio		
X axis	Y axis	Z axis	X axis	Y axis	Z axis
SD44/SD45	SD144/SD145	SD244/SD245	SD73/100	SD173/100	SD173/100
Master axis zoom ratio			When SD elements are enabled, the slave axis zoom ratio is 100% (namely, 1) by default.		

8

By default, changed electronic gear ratio and slave axis zoom ratio take effect the next time cams are started. To make them take effect currently, you need to set the cam table to modify special SM elements. In this way, the ratios will take effect in the next cam cycle. Then the modified SM elements will be automatically reset.

	X Axis	Y Axis	Z Axis	Description
Cam Table Data Modification	SM83	SM183	SM283	When the modified data takes effect, the elements will be automatically reset to OFF.

Example:



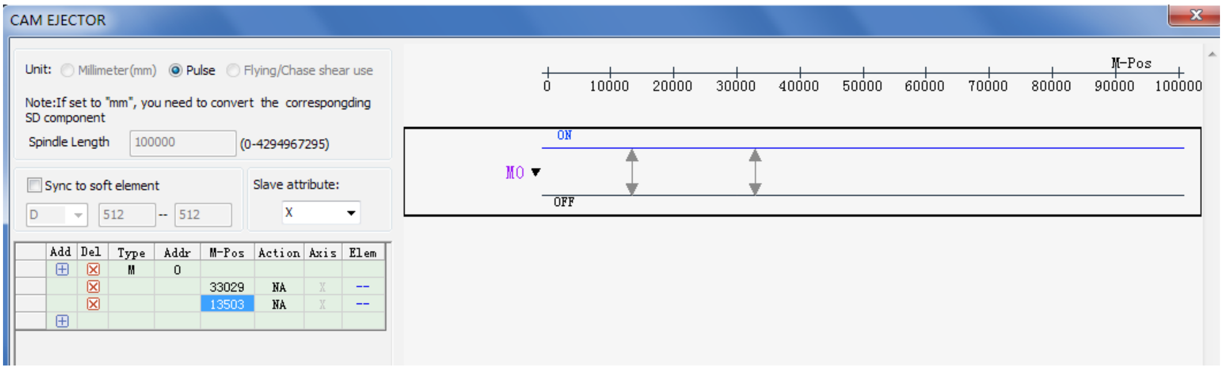
When M77 is set to ON, the slave axis zoom ratio is set to 2 (SD173/100 = 2) and will take effect in the next cam cycle.

When M76 is set to ON, the slave axis zoom ratio is set to 0.5 (SD173/100 = 0.5) and will take effect in the next cam cycle.

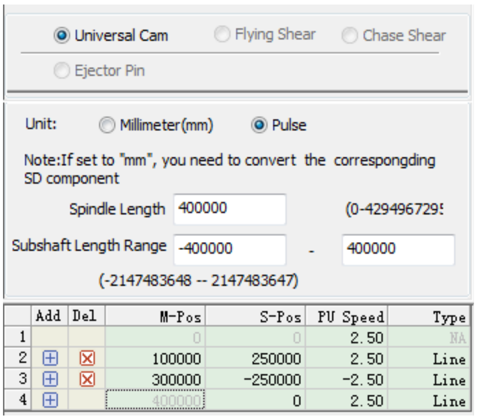
8.7 Ejector Pin

The ejector pin can turn on and off bit elements (M and Y elements) based on the master axis position. The following describes how to set the ejector pin.

1. On the cam curve editing page, click **Ejector Pin**.



2. Click the **Add** or **Del** icon to add or delete data respectively.



3. Configure the ejector pin in the table:

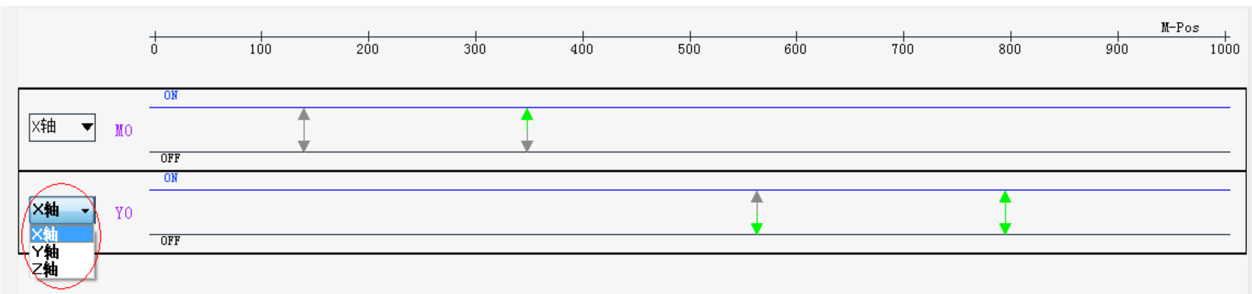
M-Pos: Sets the master axis position for an electronic cam.

Type: Sets the type of a bit element (M or Y).

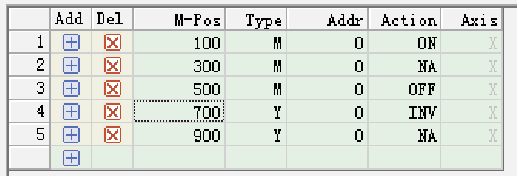
Addr: Sets the element number.

Action: Indicates the action when the master axis position matches the M-Pos value. NA indicates no action. ON indicates that the element is set to ON. OFF indicates that the element is set to OFF. INV indicates inversion operation.

Axis: Indicates the axis property. When the property of a slave axis matches the Axis value, the ejector pin data is valid. As shown in the following figure, you can select the axis property in the drop-down box.

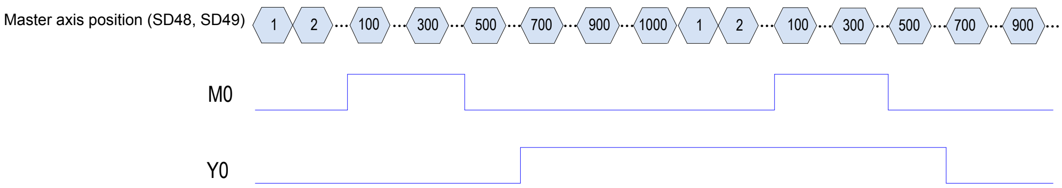


Example:



	Add	Del	M-Pos	Type	Addr	Action	Axis
1	⊞	☒	100	M	0	ON	X
2	⊞	☒	300	M	0	NA	X
3	⊞	☒	500	M	0	OFF	X
4	⊞	☒	700	Y	0	INV	X
5	⊞	☒	900	Y	0	NA	X
	⊞						

The following figure shows M0 and Y0 timing charts of an electronic cam when the X axis is selected as a slave axis.



8.8 Modifying Key Points for Electronic Cams

The cam curve established in the background can be read back, and the key points of the curve can be modified in program. The following table lists instruction formats.

Electronic Cam Instructions	"CAMWR: Writing electronic cam data"	Modifying key points in the unit of pulses
	"CAMRD: Reading electronic cam data"	Reading back key points in the unit of pulses
	"ECAMWR: Writing electronic cam floating-point decimal data"	Modifying key points in mechanical units
	"ECAMRD: Reading electronic cam floating-point decimal data"	Reading back key points in mechanical units

8.8.1 Writing Electronic Cam Data

8

CAMWR: Writing electronic cam data

◆ Overview

The CAMWR modifies electronic cam table data in the unit of pulses.

CAMWR m1 m2 D n			Writing electronic cam data	Applicable model: H3U	
m1	Cam table	Cam table to be modified			32-bit instruction (17 steps) DCAMWR: Continuous execution
m2	Starting point	Starting point			
D	Data	Data address			
n	Number of points	Number of key points			

◆ Operands

Operand	Bit Soft Element							Word Soft Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

The CAMWR instruction modifies electronic cam table data in the unit of pulses.

- [m1] specifies a cam table to be modified. When m1 = k11 to k13, the M-Pos value (SD48, SD49) of an X axis cam, the M-Pos value (SD148, SD149) of a Y axis cam, and the M-Pos value (SD248, SD249) of a Z axis cam are specified, respectively.
- [m2] sets the starting point of key points, m2 = k2 to k360.
- [D] indicates the head address. Multiple continuous address units starting with [D] are occupied. Each key point occupies two 32-bit registers that flag positions of master and slave axes respectively; that is, each key point occupies four address units.
- [n] indicates the number of key points. (m2 + n – 1) must be less than or equal to the number of downloaded key points.

◆ Notes:

1) **Only one CAMWR instruction can be executed at a time. If more than two CAMWR instructions are needed, the next instruction can be executed after the current instruction stops a scan cycle.**

2) **CAMWR instructions are executed in multiple cycles. An OFF-to-ON switch of the special register SM82 indicates that modification is finished.**

3) **When modification is finished, cam table data is changed. In this case, the data uploaded or read back with instructions is modified data. If an electronic cam is running, modified data cannot take effect immediately. To make the data take effect in the next cam cycle, you need to set special flag bits (SM83, SM183, and SM283). If the electronic cam is disabled, or if modified data does not take effect when the electronic cam is running, the data will automatically take effect when the electronic cam is started the next time.**

4) **The first point of an electronic cam table is the starting point and cannot be modified. Therefore, m2 must be greater than 1. (m2 + n – 1) must be less than or equal to the number of downloaded key points.**

5) **The M-Pos value must be greater than that of the previous point and less than that of the next point, or error 16268 will be returned.**

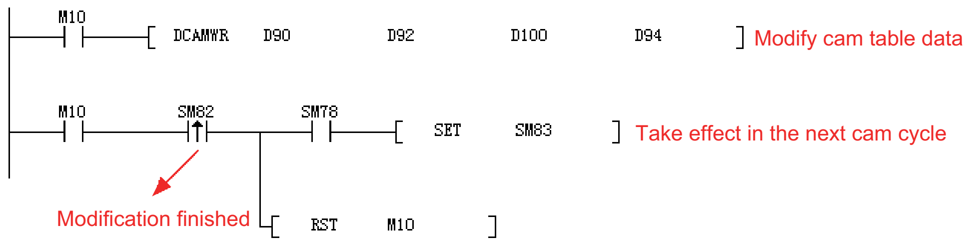
8

6) **Cam tables to be modified as specified by CAMWR instructions must exist in the PLC; that is, the cam tables have been downloaded to the PLC through AutoShop. Only cam table data in the unit of pulses can be modified.**

The following table lists special registers involved in modification.

SM82		An OFF-to-ON switch of SM82 indicates that modification is finished.
SM83	X axis	Flags indicating that modified data takes effect during running of an electronic cam: ON: Modified data takes effect in the next cam cycle, and the registers are automatically reset to OFF. OFF: Modified data takes effect the next time the cam is started.
SM183	Y axis	
SM283	Z axis	

◆ Example:



When M10 is set to ON, the PLC executes DCAMWR instructions to modify the cam table. When modification is finished, SM82 is switched from OFF to ON. If the X axis cam is running, you can set SM83 (X axis) to make the modified data take effect in the next cam cycle. Then SM83 is automatically reset.

Master and slave axes of each key point occupy two 32-bit registers; that is, each key point occupies four D elements. If five points are modified, from the second point, (D101, D100) indicates the master axis position, (D103, D102) indicates the slave axis position, and so on. Twenty D elements are occupied.

8.8.2 Writing Electronic Cam Floating-point Decimal Data

ECAMWR: Writing electronic cam floating-point decimal data

◆ Overview

The ECAMWR instruction modifies electronic cam floating-point decimal data in mechanical units.

ECAMWR m1 m2 D n			Writing electronic cam floating-point data	Applicable model: H3U
m1	Cam table	Cam table to be modified		32-bit instruction (17 steps) DECAMWR: Continuous execution
m2	Starting point	Starting point		
D	Data	Data address		
n	Number of points	Number of key points		

◆ Operands

Operand	Bit Soft Element							Word Soft Element																
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

The CAMWR instruction modifies electronic cam table data in the unit of pulses.

- [m1] specifies a cam table to be modified. When m1 = k11 to k13, the M-Pos value (SD48, SD49) of an X axis cam, the M-Pos value (SD148, SD149) of a Y axis cam, and the M-Pos value (SD248, SD249) of a Z axis cam are specified, respectively.
- [m2] sets the starting point of key points, m2 = k2 to k360.
- [D] indicates the starting address. Multiple continuous address units starting with [D] are occupied. Each key point occupies two 32-bit registers that flag positions of master and slave axes respectively; that is, each key point occupies four address units.
- [n] indicates the number of key points. (m2 + n – 1) must be less than or equal to the number of downloaded key points.

◆ Notes:

1) **ECAMWR instructions are used to modify floating-point decimal data.**

2) **Except for [D] data interpreted as floating-point decimal data, ECAMWR instructions are used in the same way as CAMWR instructions. For details, see** **“CAMWR: Writing electronic cam data” on page 576****.**

8

8.8.3 Reading Electronic Cam Data

CAMRD: Reading electronic cam data

◆ Overview

The CAMRD instruction reads electronic cam table data in the unit of pulses.

CAMRD m1 m2 D n		Electronic cam data reading	Applicable Model: H3U	
m1	Cam table	Cam table to be read		32-bit instruction (17 steps) DCAMRD: Continuous execution
m2	Starting point	Starting point		
D	Data	Data address		
n	Number of points	Number of key points		

◆ Operands

Operand	Bit Soft Element							Word Soft Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

8

The CAMRD instruction reads electronic cam table data in the unit of pulses.

- [m1] specifies a cam table to be read. When m1 = k11 to k13, the M-Pos value (SD48, SD49) of an X axis cam, the M-Pos value (SD148, SD149) of a Y axis cam, and the M-Pos value (SD248, SD249) of a Z axis cam are specified, respectively.
- [m2] sets the starting point, m2 = k1 to k360.
- [D] indicates the starting address. Multiple continuous address units starting with [D] are occupied. Each key point occupies two 32-bit registers that flag positions of master and slave axes respectively; that is, each key point occupies four address units.
- [n] indicates the number of key points. (m2 + n – 1) must be less than or equal to the number of downloaded key points.

◆ Notes:

1) **CAMRD instructions are used to read cam table data in the unit of pulses. Cam tables to be read must exist in the PLC; that is, the cam tables have been downloaded to the PLC through AutoShop.**

2) **(m2 + n – 1) must be less than or equal to the number of downloaded key points.**

◆ Example:

When M10 is set to ON, the PLC executes DCAMRD instructions to read the cam table. The read data is stored in D elements starting with D200.

Master and slave axes of each key point occupy two 32-bit registers; that is, each key point occupies four D elements. If five points are read, from the first point, (D201, D200) indicates the master axis position, (D203, D202) indicates the slave axis position, and so on. Twenty D elements are occupied.

8.8.4 Reading Electronic Cam Floating-point Decimal Data

ECAMRD: Reading electronic cam floating-point decimal data

◆ Overview

The ECAMRD instruction reads electronic cam floating-point decimal data in mechanical units.

CAMRD m1 m2 D n			Reading electronic cam floating-point decimal data	Applicable model: H3U	
m1	Cam table	Cam table to be read			32-bit instruction (17 steps) DECAMRD: Continuous execution
m2	Starting point	Starting point			
D	Data	Data address			
n	Number of points	Number of key points			

◆ Operands

Operand	Bit Soft Element							Word Soft Element																
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

The CAMRD instruction reads electronic cam table data in the unit of pulses.

- [m1] specifies a cam table to be read. When m1 = k11 to k13, the M-Pos value (SD48, SD49) of an X axis cam, the M-Pos value (SD148, SD149) of a Y axis cam, and the M-Pos value (SD248, SD249) of a Z axis cam are specified, respectively.
- [m2] sets the starting point, m2 = k1 to k360.
- [D] indicates the starting address. Multiple continuous address units starting with [D] are occupied. Each key point occupies two 32-bit registers that flag positions of master and slave axes respectively; that is, each key point occupies four address units.

8

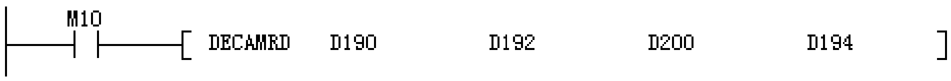
- [n] indicates the number of key points. (m2 + n – 1) must be less than or equal to the number of downloaded key points.

◆ Notes:

1) **CAMRD instructions are used to read cam table data in mechanical units, and the read data is interpreted as floating points. Cam tables to be read must exist in the PLC; that is, the cam tables have been downloaded to the PLC through AutoShop.**

2) **(m2 + n – 1) must be less than or equal to the number of downloaded key points.**

◆ Example:



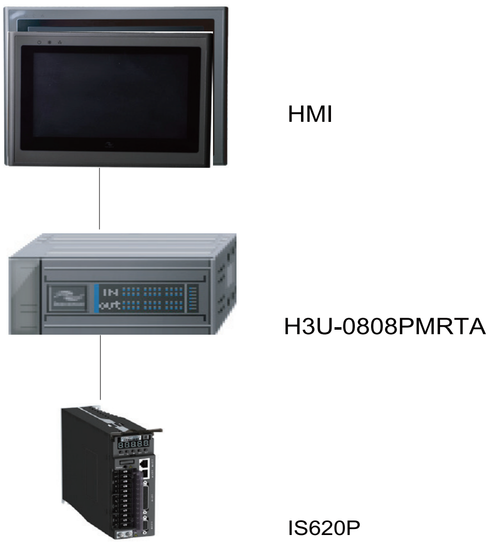
When M10 is set to ON, the PLC executes DECAMRD instructions to read the cam table. The read data is stored as floating points in D elements starting with D200.

Master and slave axes of each key point occupy two 32-bit registers; that is, each key point occupies four D elements. If five points are read, from the first point, (D201, D200) indicates the master axis position, (D203, D202) indicates the slave axis position, and so on. Twenty D elements are occupied.

8.9 Application Examples (Application of H3U-PM to Strapping Band Winding Machines)

◆ Overview

This solution describes the application of Inovance H3u-PM motion control PLCs to strapping band winding machines. Different from traditional wire ranging way, the H3u-PM+IS620P electronic cam solution is undemanding for the mechanical structure of devices, which makes it easy to change strapping band width and winding width.



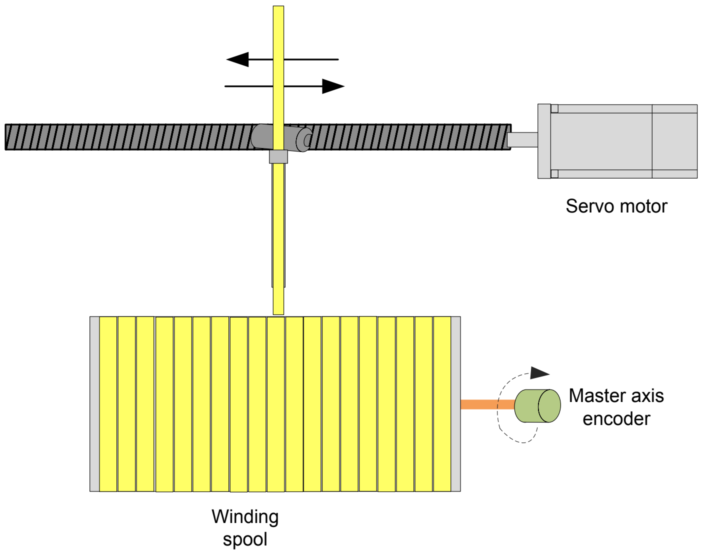
◆ Strapping Band Winding Machine

A Strapping band is made from polyethylene and polypropylene, squeezed out, and stretched unidirectionally. It can be used to seal up cartons and strap other packaging materials.

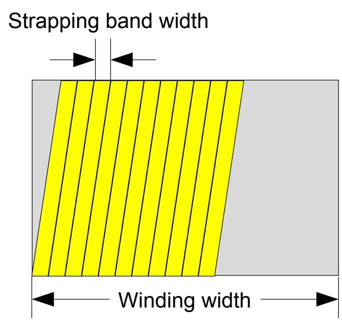
After a strapping band is squeezed out and stretched by upstream equipment, the strapping band winding machine coils it. This section details the winding control process.

◆ Winding Control Process

When the AC drive drives the winding spool (master axis), the wiring ranging servo (slave axis) moves with the master axis along the planned cam curve under control of the PLC based on pulse signals collected by the master axis encoder. In this way, the strapping band is coiled.



After the winding spool (master axis) finishes a circle, the distance the wiring ranging servo (slave axis) moves is the strapping band width, as shown in the following figure.



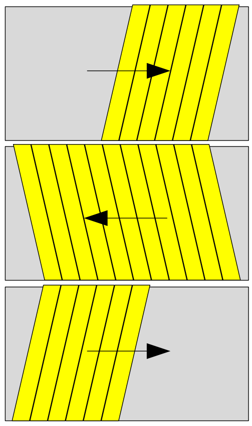
◆ Establishing an Electronic Cam

1) Master/Slave Axis Selection

The wiring ranging servo moves with the spool (master axis) as a slave axis. The rotary encoder of the spool collects pulse signals.

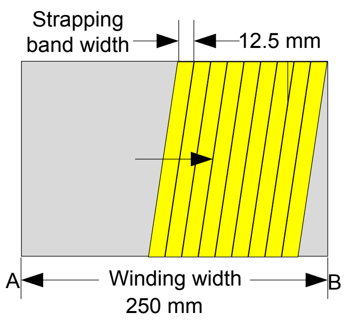
The strapping band moves from the middle of the spool to the other end, and then moves back to the starting point. This completes a cam cycle.

8



2) Master/Slave Axis Length

The number of pulses generated per circle in which the master axis encoder rotates is 2500 x 4 = 10,000 pp (a 2500-cable A/B phase encoder is used, and a multiplier 4 is used for high-speed counting). The wiring ranging servo motor is directly connected to the screw rod through the coupling. 10,000 pp of pulses are generated per circle in which the servo motor rotates, and the screw pitch is 5 mm.



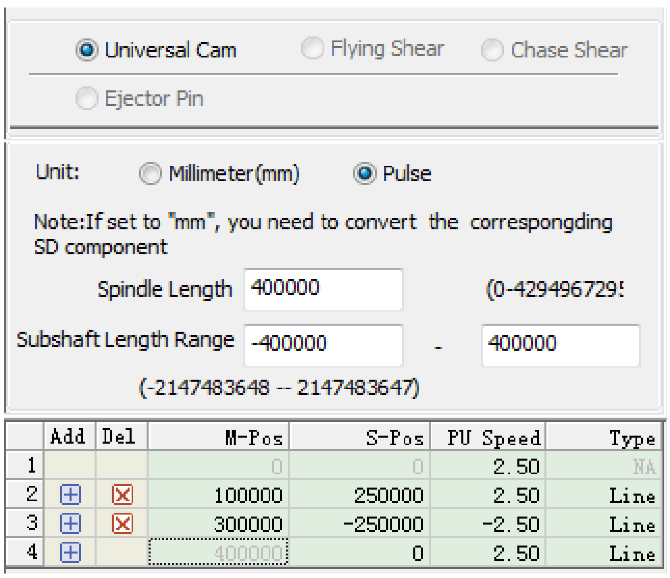
250/12.5 = 20 circles: Indicates the number of circles in which the strapping band is wound from A to B.

Master axis pulse: 20 x 10,000 x 2 = 400,000 pp

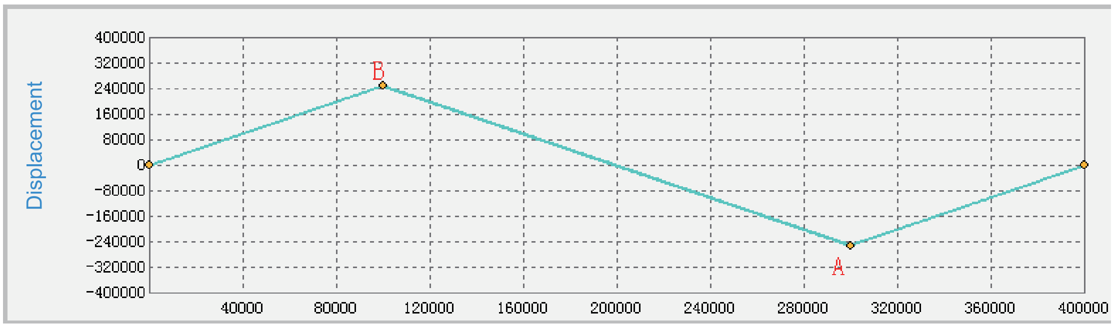
Slave axis pulse: 20 x 10,000 x 12.5/5 = 500,000 pp

8

3) Curve Planning

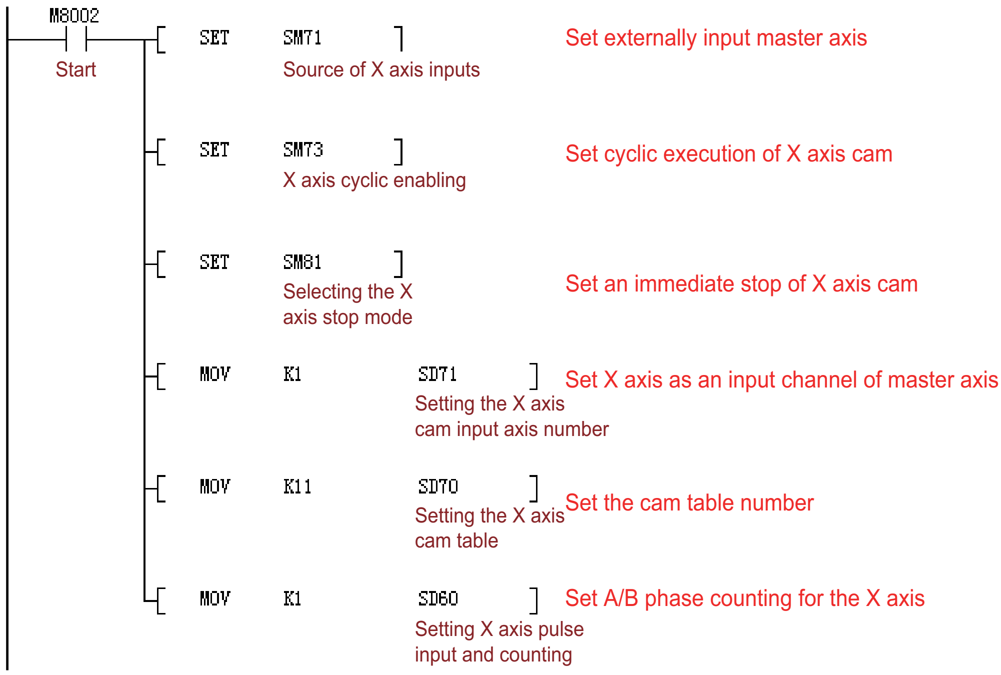


	Add	Del	M-Pos	S-Pos	PU Speed	Type
1			0	0	2.50	NA
2	⊞	☒	100000	250000	2.50	Line
3	⊞	☒	300000	-250000	-2.50	Line
4	⊞		400000	0	2.50	Line



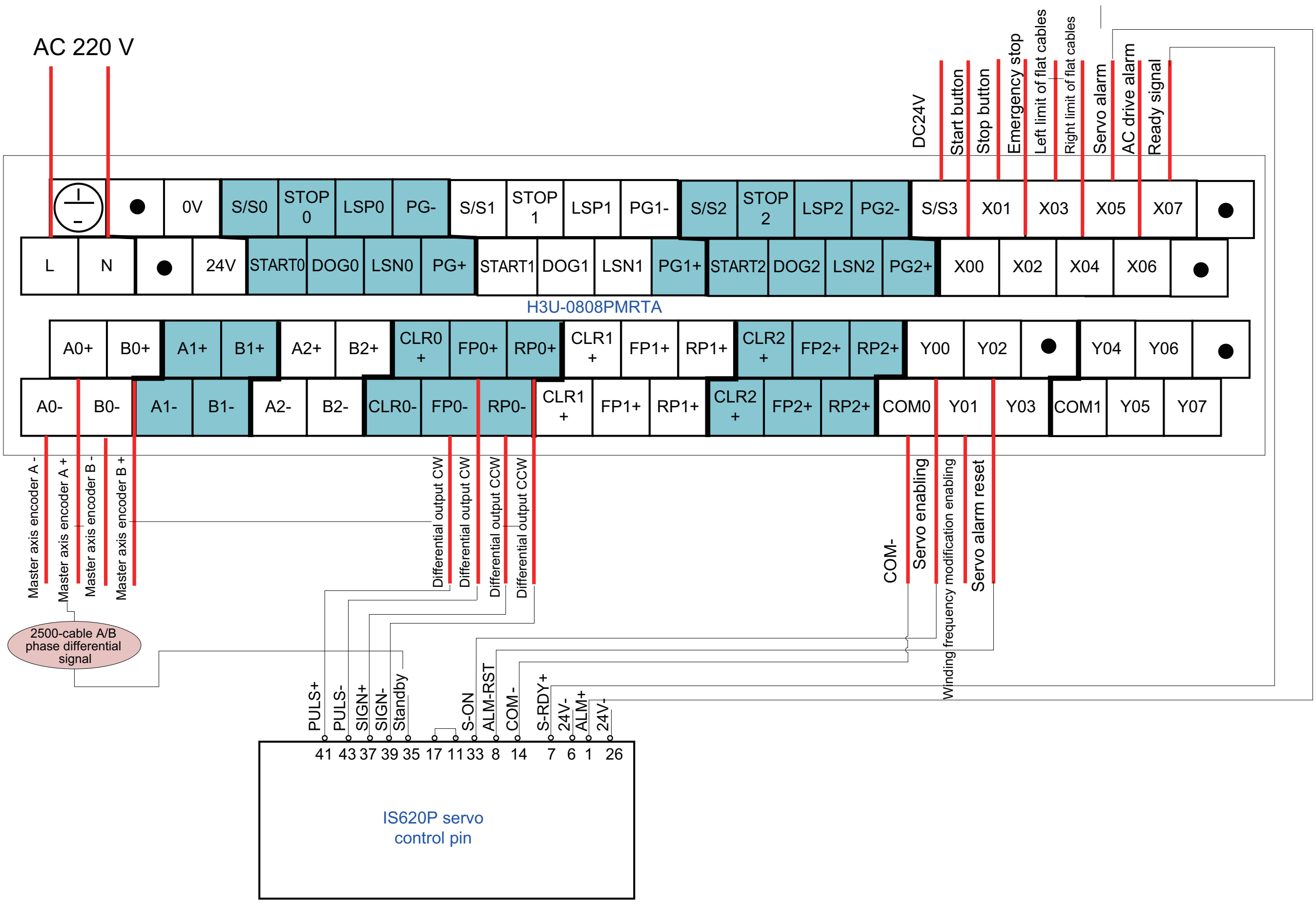
The cam moves from the middle of the spool to B and then to A, and returns to the middle of the spool. This completes a cam cycle.

4) Cam Programming



8

◆ Electrical Wiring



Wiring diagram of strapping band winding machine

8

Memo NO. ____________

Date / /

9 Communication

Chapter 9 Communication

9.1 Overview

The main module of an H3U PLC has Ethernet communication and CAN communication ports, and supports CANlink and CANopen protocols.

The module has two independent serial communication ports: COM0 and COM1. COM0 can be used for programming and monitoring. COM1 functions are user-defined.

The module has one MiniUSB communication port, which can be used for uploading, downloading, monitoring, and configuration.

The following figure shows position of each port and instructions.

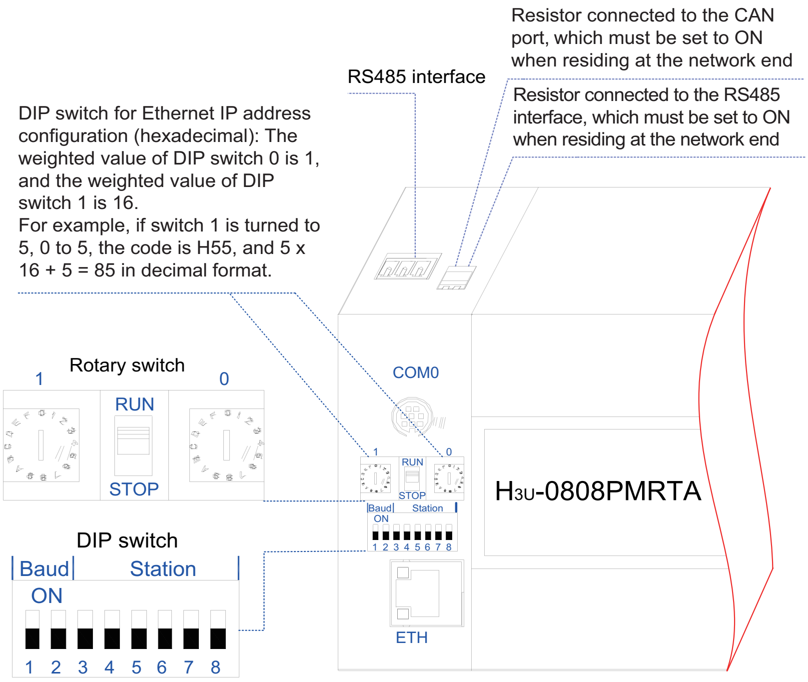


Diagram of H3U-3232MT/R and H3U-0808PMRTA communication interfaces

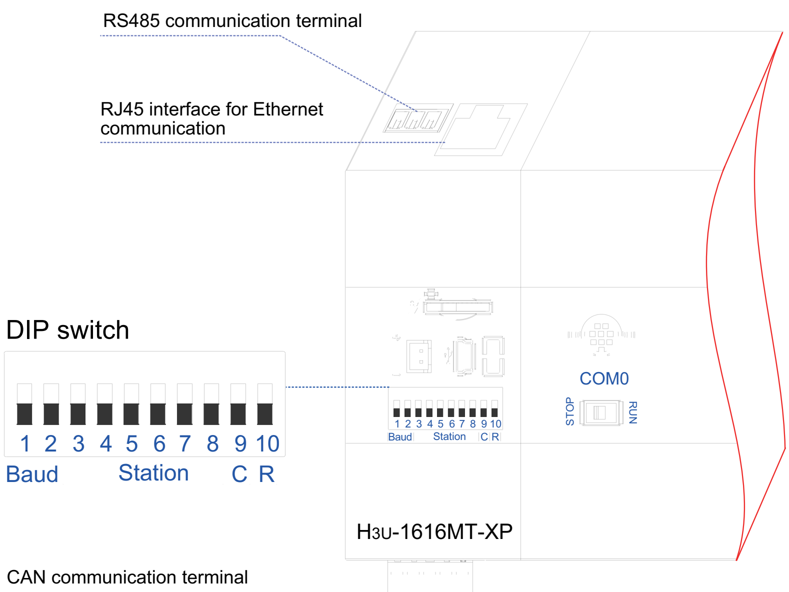


Diagram of H3U-1616MT/R-XP communication interface

9

9.2 Serial Port

9.2.1 Hardware and Communication Cable

COM0 is a standard RS-422 interface, and the interface terminal is an 8-hole mouse female connector. The USB communication port is a MiniUSB interface.

Interface definition

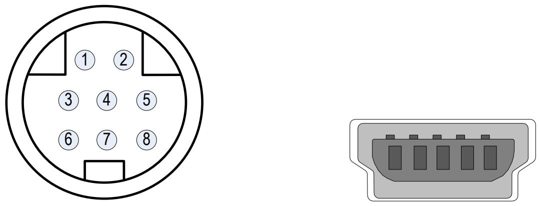


Diagram of COM0 and MiniUSB program downloading ports

Pin No.	Signal	Description
1	RXD-	Receives negative data.
2	RXD+	Receives positive data.
3	GND	Indicates a ground wire, 9 and 10 not electrically connected.
4	TXD-/RXD-	Externally sends negative data, or receives negative data if the port is an RS-485 interface.
5	+5V	Indicates external power supply +5 V, the same as the logical +5 V used internally.
6	CCS	Indicates a communication direction control cable, high level for "send" and low level for "receive". When the serial port is an RS-485 interface, the PLC controls sending or receiving data on pins 4 and 7. If the port is an RS-422 interface, the level keeps high, and pins 4 and 7 are used to send data.
7	TXD+/RXD+	Externally sends positive data, or receives positive data if the port is an RS-485 interface.
8	NC	Indicates no pin.

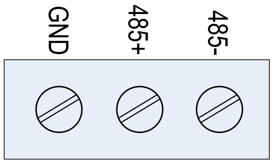
Through COM0, the PLC can be connected to a PC or touchscreen in two ways.

① Method 1: The PLC uses an RS-422 interface, and the PC uses a USB port. The PC is connected to the COM0 program downloading port through a special USB downloading cable.

② Method 2: The PLC uses an RS-422 interface, and the PC uses an RS-232 interface. The PC is connected to the COM0 program downloading port through a special serial port downloading cable.

H3U PLCs support USB-driven downloading. The USB driver is in the folder named **usb** in the background installation directory.

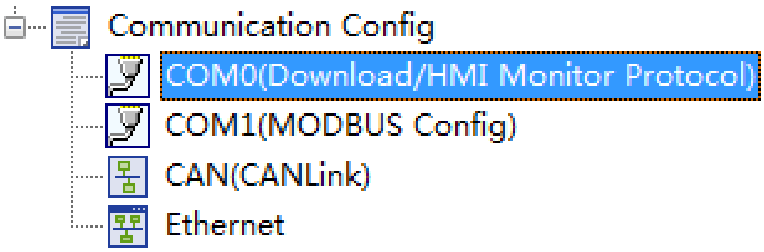
COM1 is an RS-485 interface, and the interface terminal is a wiring terminal. The following figure shows the interface definition.

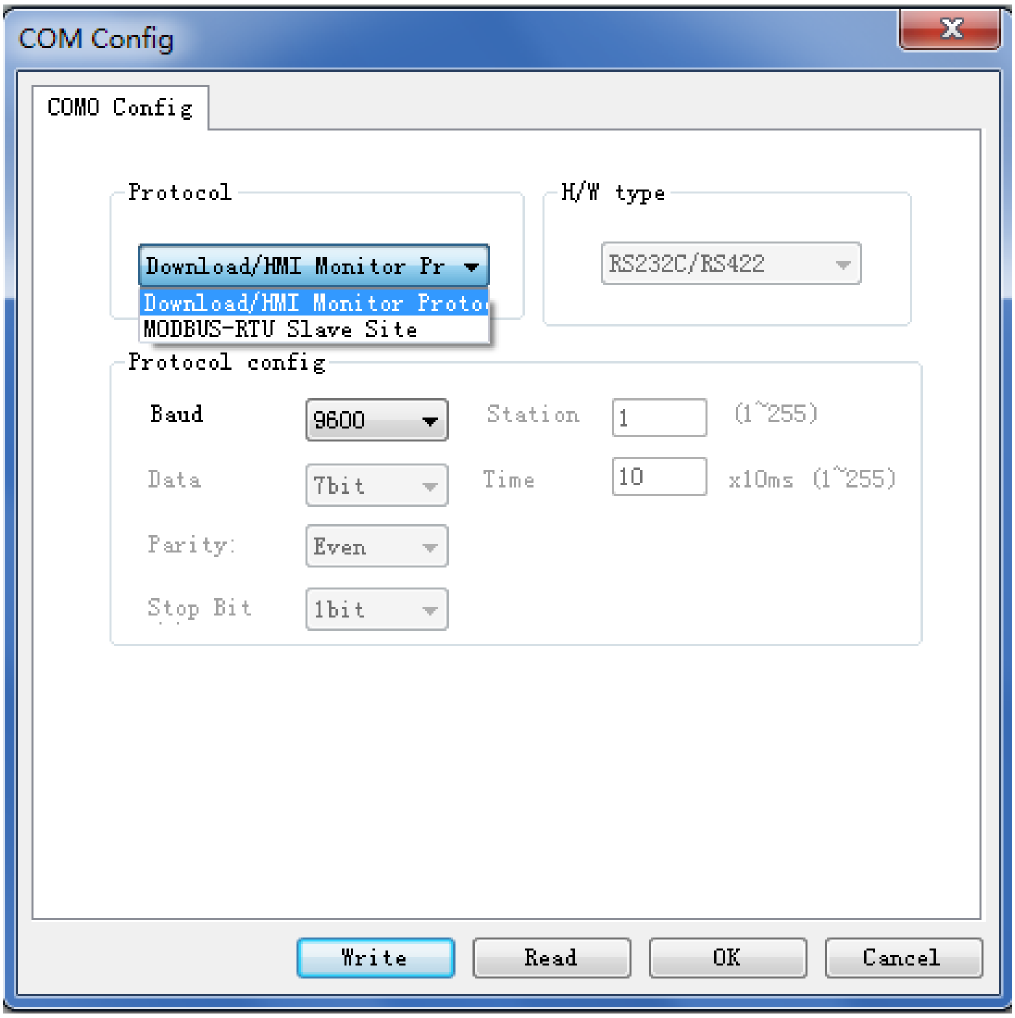


COM1 is connected to other devices through the wiring terminal.

9.2.2 Setting Communication Protocols

1) COM0 protocol configuration





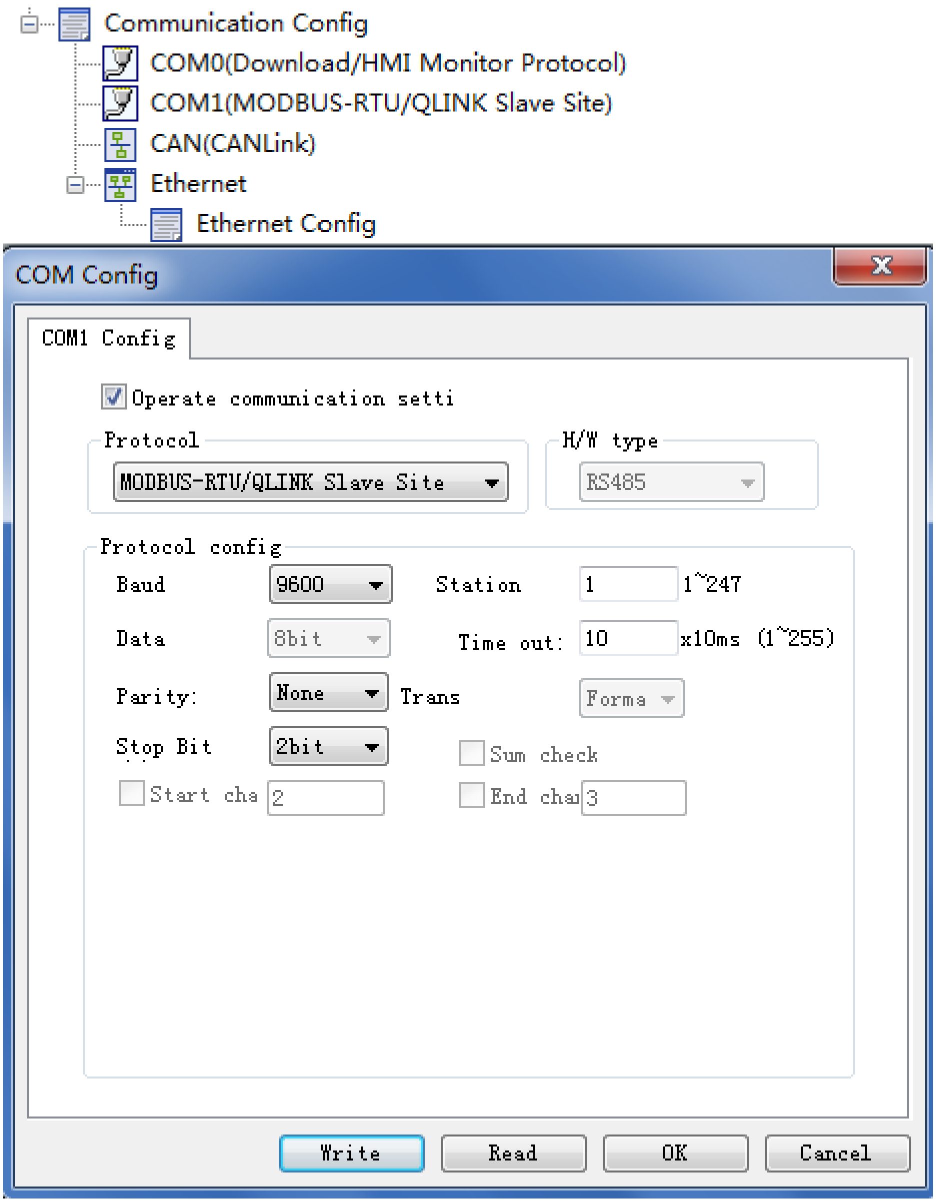
By default, the COM0 protocol is a download or Human Machine Interface (HMI) monitoring protocol.

9

Table of protocol configuration

COM0 Protocol	D8116	Half-duplex/full-duplex mode	COM0 communication format
Download/HMI Monitoring Protocol	Non-02h	Not supported	Fixed
Modbus-RTU Slave Station	02h	Half-duplex	Determined by COM0 configuration and displayed through D8110

2) COM1 protocol configuration

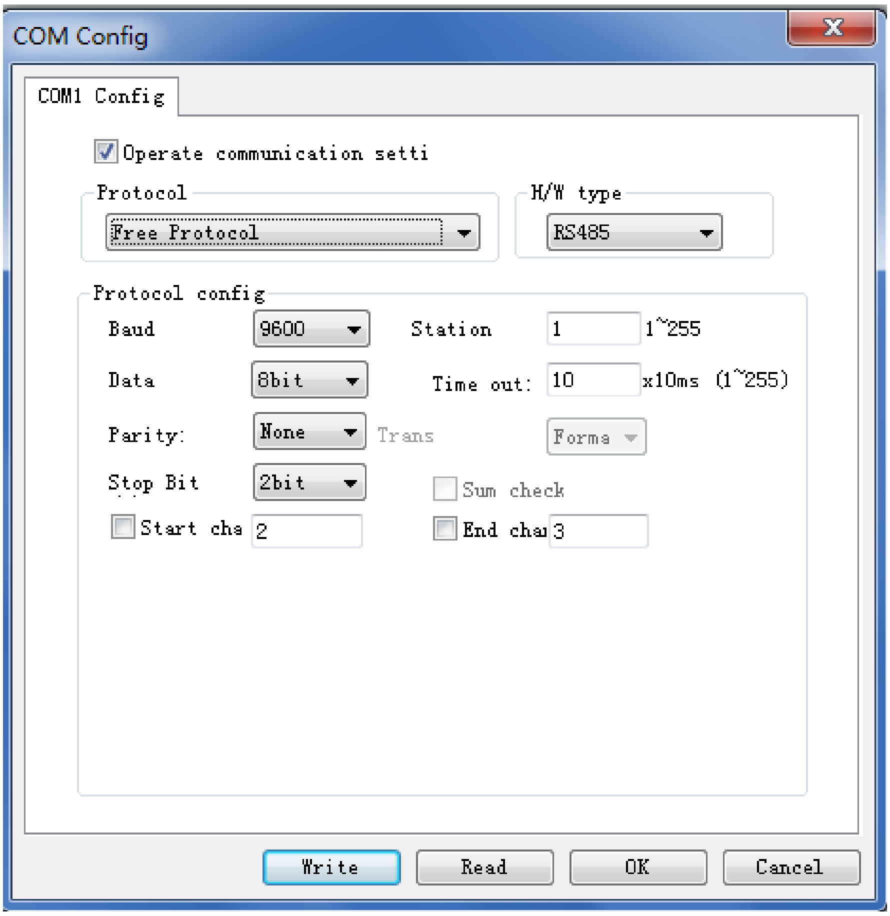


Conditions for protocol switchover: You can switch from one protocol to another on COM1 when the system is running or the user program is disabled. Protocols cannot be changed when the user program is enabled.

Table of protocol configuration

COM1 Protocol	D8126	Half-duplex/full-duplex mode	COM1 communication format
HMI Monitoring Protocol	01h	Half-duplex	Fixed
Modbus-RTU Slave Station/QLINK	02h	Half-duplex	Determined by COM1 configuration and displayed through D8120
Modbus-ASC Slave Station	03h	Half-duplex	
RS Instruction	10h	Half-duplex	
Modbus-RTU Instruction	20h	Half-duplex	
Modbus-ASC Instruction	30h	Half-duplex	
Modbus Configuration	60h	Half-duplex	

The following figure shows COM1 configuration.



NOTE

- The protocol cannot be changed when the PLC is running.

9

3) Serial port communication format

Comparison table of protocol and communication format

<table>
<tr><th>Protocol</th><th>Baud Rate</th><th>Data Bit</th><th>Check Bit</th><th>Stop Bit</th></tr>
<tr><td>HMI monitoring protocol</td><td>9600</td><td>7</td><td>Even parity check (E)</td><td>1</td></tr>
<tr><td>Modbus-RTU Slave Station</td><td rowspan="6">Serial port 0 is checked through D8110, and serial port 1 is checked through bits 4 to 7 of D8120.
0011b: 300 bits/s
0100b: 600 bits/s
0101b: 1200 bits/s
0110b: 2400 bits/s
0111b: 4800 bits/s
1000b: 9600 bits/s
1001b: 19200 bits/s
1010b: 38400 bits/s
1011b: 57600 bits/s
1100b: 115200 bits/s</td><td rowspan="6">Serial port 0 is checked through D8110, and serial port 1 is checked through bit 0 of D8120.
0b: 7 bits
1b: 8 bits
Note:
Modbus-RTU slave and master station protocols support only 8-bit data; otherwise, communication errors will occur.</td><td rowspan="6">Serial port 0 is checked through D8110, and serial port 1 is checked through bits 1 to 2 of D8120.
00b: No parity check (N)
01b: Odd parity check (O)
11b: Even parity check (E)</td><td rowspan="6">Serial port 0 is checked through D8110, and serial port 1 is checked through bit 3 of D8120.
0: 1 bit
1: 2 bits</td></tr>
<tr><td>Modbus-ASC Slave Station</td></tr>
<tr><td>Free RS protocol</td></tr>
<tr><td>Modbus-RTU master station</td></tr>
<tr><td>Modbus-ASC master station</td></tr>
<tr><td>Modbus configuration</td></tr>
</table>

4) List of soft elements in serial port communication format

- COM0

M8110	Reserved	D8110	Communication format
M8111	Reserved	D8111	Communication station number
M8112	Modbus - communication execution state	D8112	Downloading and HMI monitoring protocols - communication format
M8113	Modbus - communication error flag	D8113	Reserved
M8114	Reserved	D8114	Reserved
M8115	Reserved	D8115	Reserved
M8116	Reserved	D8116	Communication protocol
M8117	Reserved	D8117	Reserved
M8118	Reserved	D8118	Modbus - number of station with communication errors
M8119	Timeout criterion	D8119	Communication timeout period

- COM1

M8120	Reserved	D8120	Communication format
M8121	RS instruction - sending	D8121	Communication station number
M8122	Modbus - communication execution state RS instruction - flag of sending	D8122	Downloading and HMI monitoring protocols - communication format RS instruction - volume of residual data transmitted
M8123	Modbus - communication error flag RS instruction - flag of receipt	D8123	RS instruction - volume of received data
M8124	RS instruction - receiving	D8124	RS instruction - Start of Text (STX)
M8125	Reserved	D8125	RS instruction - End of Text (ETX)
M8126	Reserved	D8126	Communication protocol
M8127	Reserved	D8127	PC link protocol - starting data address required
M8128	Reserved	D8128	Modbus - number of station with communication errors PC link protocol - volume of sent data required
M8129	Timeout criterion	D8129	Communication timeout duration

5) List of communication error codes

Parallel Connection Communication Error M8063 (D8063) Continuation	0000	Normal	
	6301	Odd/even check error, overflow error, and frame error	Check whether PLCs are powered on, whether the adapter is correctly connected to the controller, and whether adapters are correctly connected to each other.
	6302	Incorrect communication character	
	6303	Inconsistent communication data sum	
	6304	Incorrect data format	
	6305	Incorrect instruction	
	6306	Monitor timer timeout	
	6307 to 6311	None	
	6312	Incorrect parallel character	
	6313	Incorrect parallel sum	
	6314	Incorrect parallel format	
	6330	Incorrect Modbus slave station address	If errors occur during COM0 communication, check whether the COM0 communication cable is correctly connected.
	6331	Incorrect data frame length	
	6332	Incorrect address	
	6333	CRC error	
	6334	Unsupported instruction code	
	6335	Receiving timeout	
	6336	Data error	
	6337	Buffer overflow	
	6338	Frame error	
	6339	Serial protocol error	
	6340	Incorrect Modbus slave station address	If errors occur during COM1 communication, check whether the COM1 communication cable is correctly connected. Check whether communication formats are matched.
	6341	Incorrect data frame length	
	6342	Incorrect address	
	6343	CRC error	
	6344	Unsupported instruction code	
	6345	Receiving timeout	
	6346	Data error	
	6347	Buffer overflow	
	6348	Frame error	
	6349	Serial protocol error	

NOTE

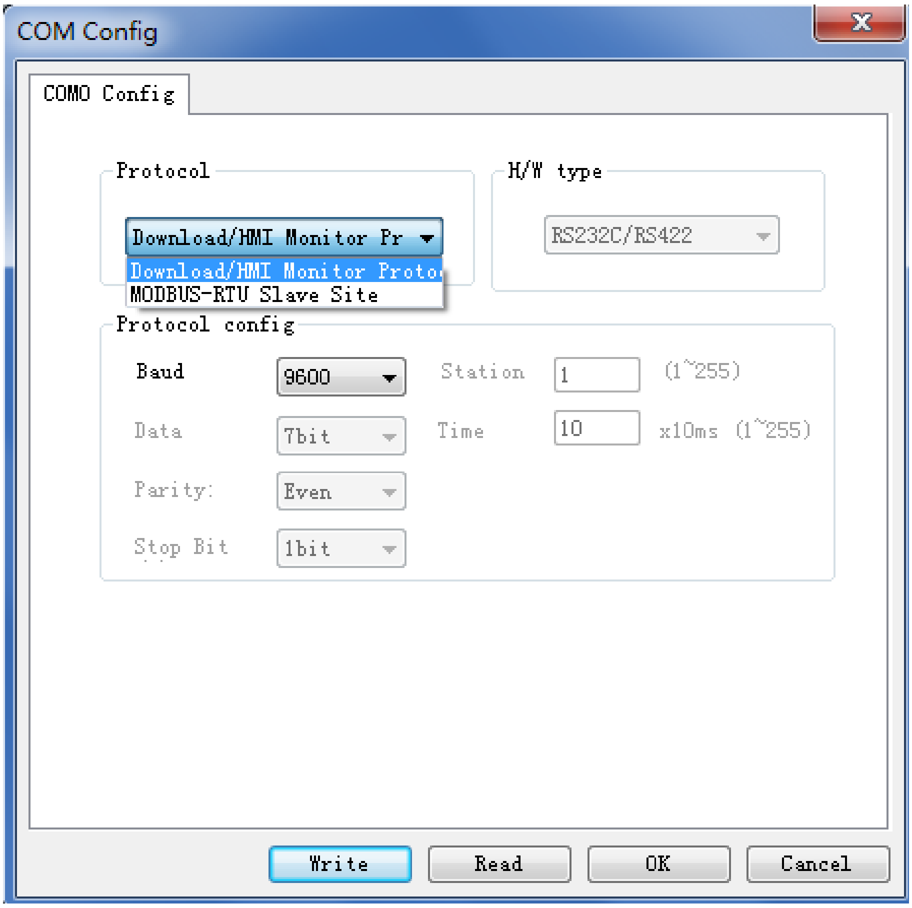
- M8063 and D8063 persist after troubleshooting until they are cleared manually.

9.3 HMI Monitoring Protocol

1) Hardware connection

During communication through COM0, devices are connected through RS-422. Only the downloading port can be used. During communication through COM1, devices are connected through RS-485. The wiring terminal is used.

2) Software configuration



3) Protocol description

The HMI monitoring protocol is an internal PLC protocol. It is used for communication between AutoShop and PLC. Through this protocol, AutoShop can erase, read, and download user programs, and remotely supervise, adjust, and control the PLC. AutoShop can supervise and change any PLC elements, start, and stop the PLC.

9.4 Modbus Protocol

RS-485 signals underlie Modbus communication. Twisted pairs are used for connection. Therefore, the transmission distance is long and up to 1,000 meters. With high interference immunity and low cost, Modbus communication is popular among industrial control devices. Many AC drive and controller manufacturers use Modbus protocols.

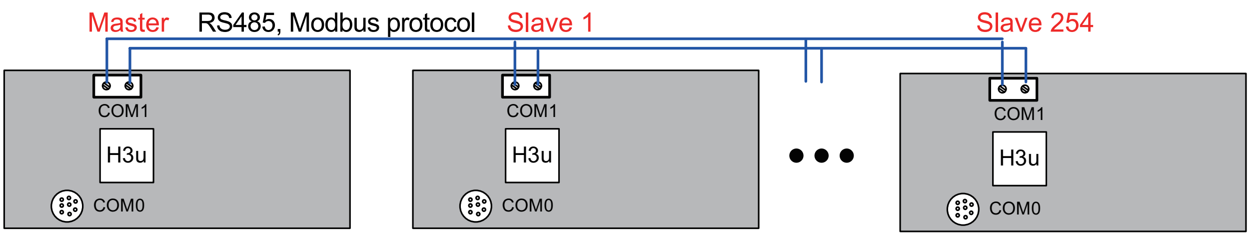
Data is transmitted in hexadecimal (HEX) and ASCII modes through Modbus-RTU and Modbus-ASC, respectively. In hexadecimal mode, data can be directly transmitted; in ASCII mode, data must be converted into ASCII codes first. Therefore, communication through Modbus-RTU is more efficient, less complicated, and more popular.

Modbus communication is in "single-master multi-slave" mode, in which the master station initiates sessions and the slave stations respond passively. Therefore, controlled devices (for example, AC drives) have slave station protocols, while controlling devices (for example, PLCs) have master and slave station protocols.

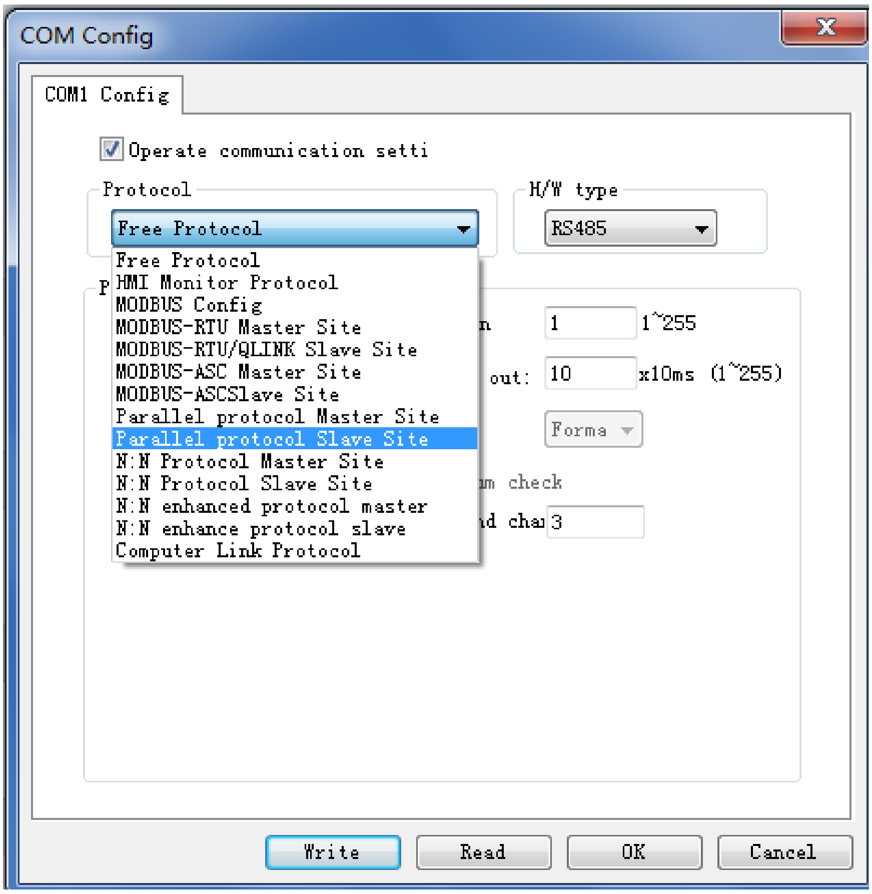
9.4.1 Modbus Master Station Communication

Modbus-RTU and Modbus-ASC instructions can be executed on COM1.

1) Hardware connection



2) Software configuration



3) Protocol description

Modbus instructions are valid on COM1. You can use Modbus instructions to program, and use the PLC as a master station to communicate with the Modbus slave station.

Multiple Modbus instructions can be available and driven at a time. The instructions are executed in sequence. According to Modbus protocols, the slave station must respond whether data is read or written (except for broadcasts). It usually takes multiple scan cycles to execute one Modbus instruction. In one scan cycle, the instruction is driven but not necessarily executed.

If multiple Modbus instructions exist, they are executed in the following order: the system scans the first driven Modbus instruction, records its parameters, and executes the instruction in the background. Then the system returns to the user program, scans the next driven Modbus instruction in the position where the previous instruction was executed, and executes the instruction.

- Instruction formats: Modbus (Addr&Cmd, RegAddr, RegLen, and DataBuf)

① Addr&Cmd indicates slave address and Modbus parameter number. Upper eight bits indicate the slave address, namely the address of the target device. Lower eight bits indicate the Modbus parameter number. Defined by a standard Modbus protocol, the bits support 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0f, and 0x10. For definitions, see standard Modbus protocols or Modbus protocols for target devices.

② RegAddr indicates the address of the slave coil (1-bit) or register (16-bit) to be read or written. Depending on Modbus slave protocols, the value can be indicated by an element or a constant.

③ RegLen indicates the number of slave coils or registers to be read or written. The value can be indicated by an element or a constant.

④ DataBuf can be used for D elements only. It indicates the starting register for data storage, namely the data buffer. The buffer length is correlated to RegLen. The value should be not less than 1. If the Modbus instruction is to read data, after the instruction is executed, the system will read slave data to the buffer. If the Modbus instruction is to write data, the system will send the buffer to the slave. During programming, you need to calculate the buffer length to reserve enough registers as a buffer.

- State flags

① M8122: Indicates the state of instruction execution. OFF indicates that the instruction has been executed; ON indicates that the instruction is being executed. If M8122 is OFF and the instruction flow is active in a scan cycle, the system will set M8122 to ON, record parameters, and execute the instruction in the background. After communication is finished, when the system returns to the instruction, whether the instruction flow is active or not, the system will reset M8122 to OFF, immediately scan the next instruction whose flow is active, record parameters, and execute the instruction in the background.

② M8123: Indicates the state of communication. ON indicates abnormal communication, and OFF indicates normal communication.

③ M8063: Indicates instruction errors. Error codes are stored in D8063.

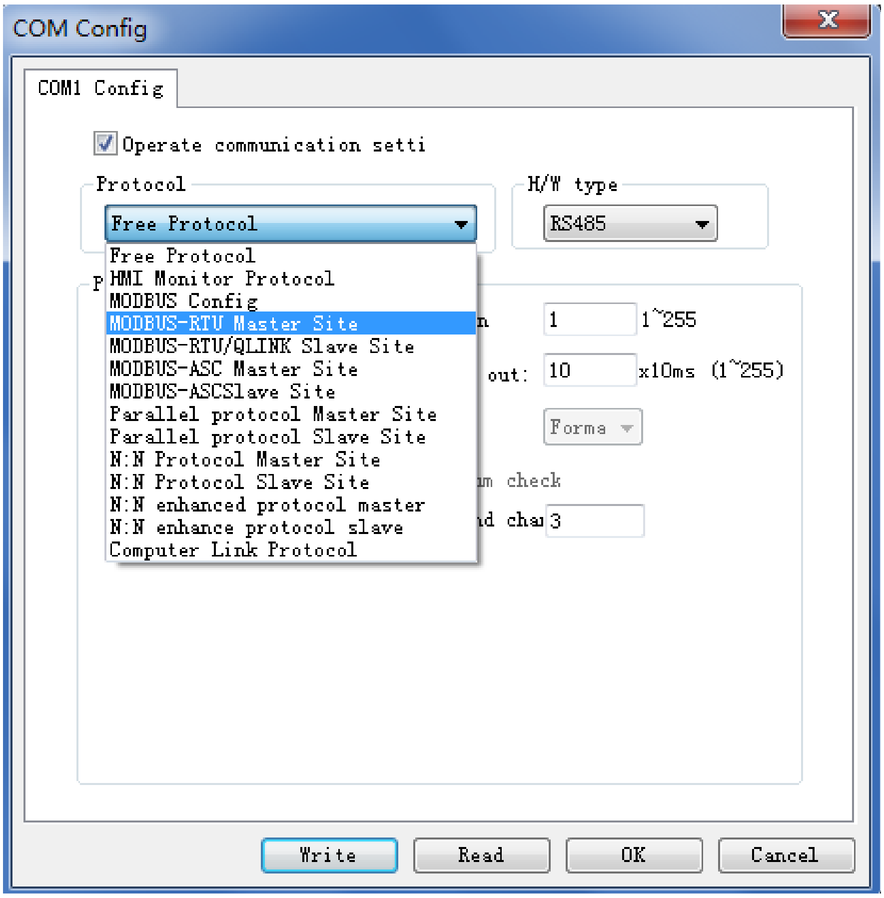
④ D8063: Indicates error codes. See the list of communication error codes.

NOTE

- In compliance with Modbus protocols, the H3U PLC can read a maximum of 125 registers and write a maximum of 123 registers at a time. If the number of registers to be read or written exceeds the limit, a parameter error will be returned.

In compliance with Modbus protocols, the H3U PLC can read and write a maximum of **255** coils at a time. If the number of registers to be read or written exceeds the limit, a parameter error will be returned.

4) Example 1: The PLC continuously reads registers whose slave address is 100. Data is stored in D10.



D0 = H0103 (Addr&Cmd): Indicates that the slave address is 01 and the Modbus instruction code is 03. The instruction is to read registers.

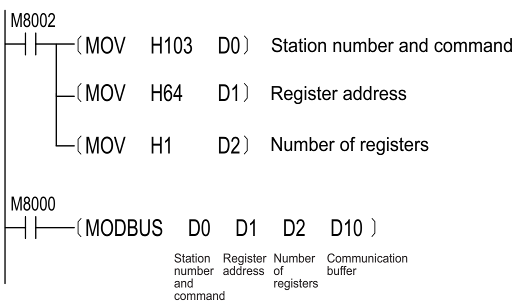
D1 = H0064 (RegAddr): Indicates the slave register address.

D2 = H0001 (RegLen): Indicates the number of registers.

D10 (Buf): Indicates the data buffer. After the instruction is executed, data is stored in D10.

The ladder chart is as follows.

9



Result: The PLC continuously reads registers whose slave address is 100, and sends the frame (hexadecimal) 01 03 00 64 00 01 C5 D5 through COM1.

01 indicates the slave address, namely the upper eight bits of D0.

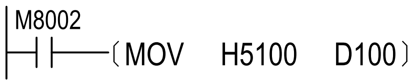
03 indicates the Modbus instruction code, namely the lower eight bits of D0. It means to read slave registers.

00 64 indicates the register address. It is the value of D1.

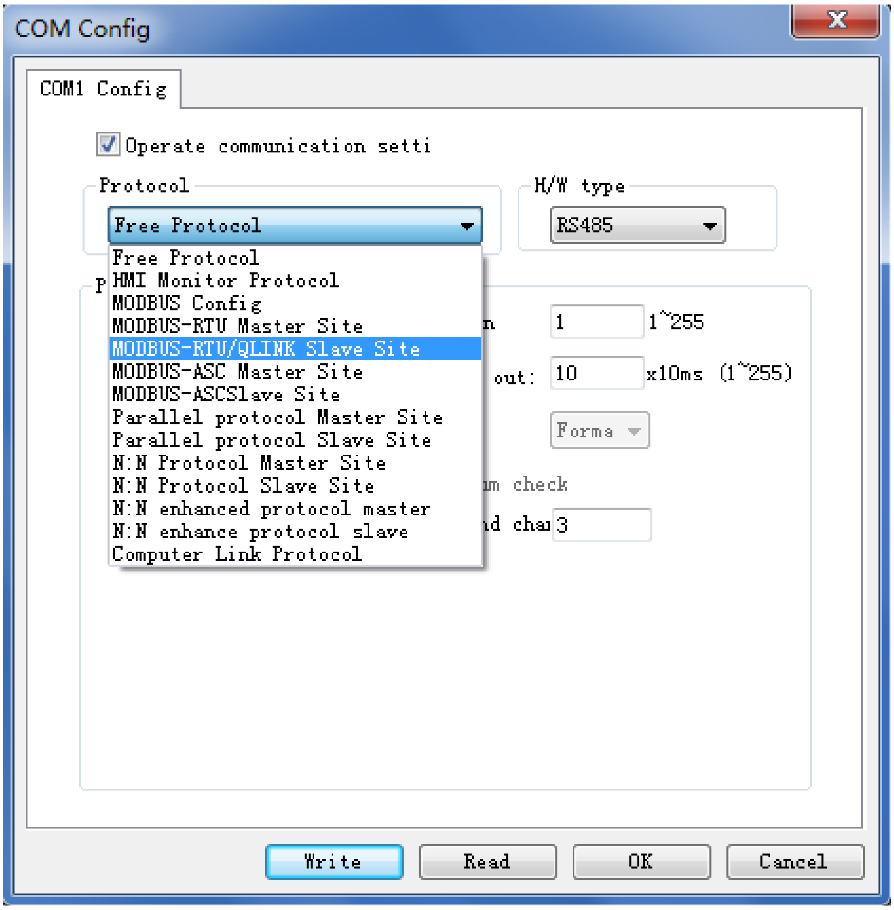
00 01 indicates the number of registers. It is the value of D2.

C5 D5 indicates the CRC code.

If the slave is an H3U PLC, select the Modbus-RTU slave station protocol. The ladder chart is as follows.



The following figure shows configuration of a slave station.



The slave correctly returns the frame (hexadecimal) 01 03 02 51 00 85 D4.

The slave sends the value of D100 (the register address is H0064) to the master.

01 indicates the slave address.

03 indicates the Modbus instruction code.

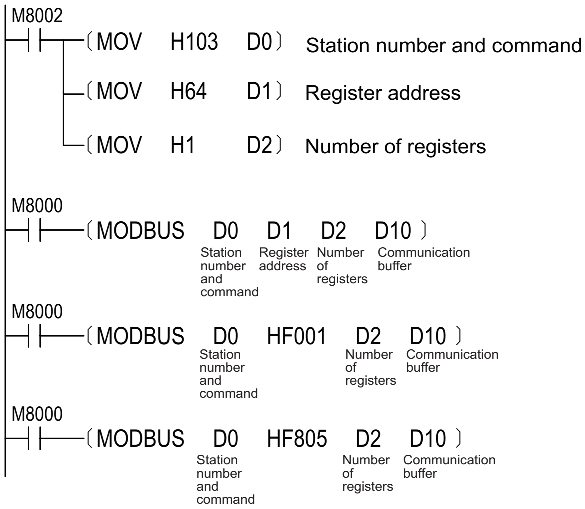
02 indicates 2-byte valid data returned.

51 00 indicates register data. It is the value of D100.

85 D4 indicates the CRC code.

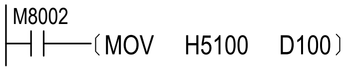
5) Example 2: The PLC executes three Modbus instructions to read registers whose slave addresses are H0064, F001, and F805, respectively. Data is stored in D10, D20, and D30.

The ladder chart is as follows.



Result: The PLC sends xx (hexadecimal) repeatedly and successively through COM1: 01 03 00 64 00 01 C5 D5, 01 03 F0 01 00 01 E6 CA, and 01 03 F8 05 00 01 A5 6B.

Response: If the slave is an H3U PLC, select the Modbus-RTU slave station protocol. The ladder chart is as follows.



Slave response

① For the first frame, the slave returns the frame (hexadecimal) 01 03 02 51 00 85 D4.

It means that the slave sends H5100, the value of D100 (the register address is H0064), to the master.

② For the second frame, the slave returns the frame (hexadecimal) 01 03 02 00 00 B8 44.

It means that the slave sends H0000, the value of T1 (the register address is F001), to the master. See "2) Variable addressing" on page 609.

③ For the third frame, the slave returns the frame (hexadecimal) 01 83 02 C0 F1.

It means Read error.

01 indicates the slave address.

83 indicates Read error.

02 indicates the error code. The address is incorrect because the register whose address is HF805 does not exist.

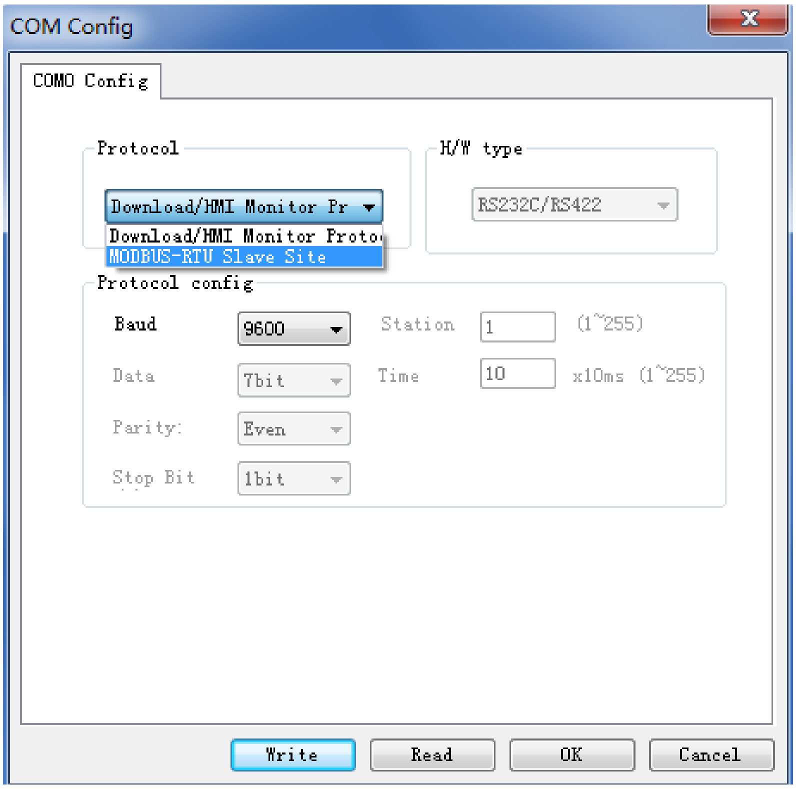
C0 F1 indicates the CRC code.

9.4.2 Modbus Slave Station Communication

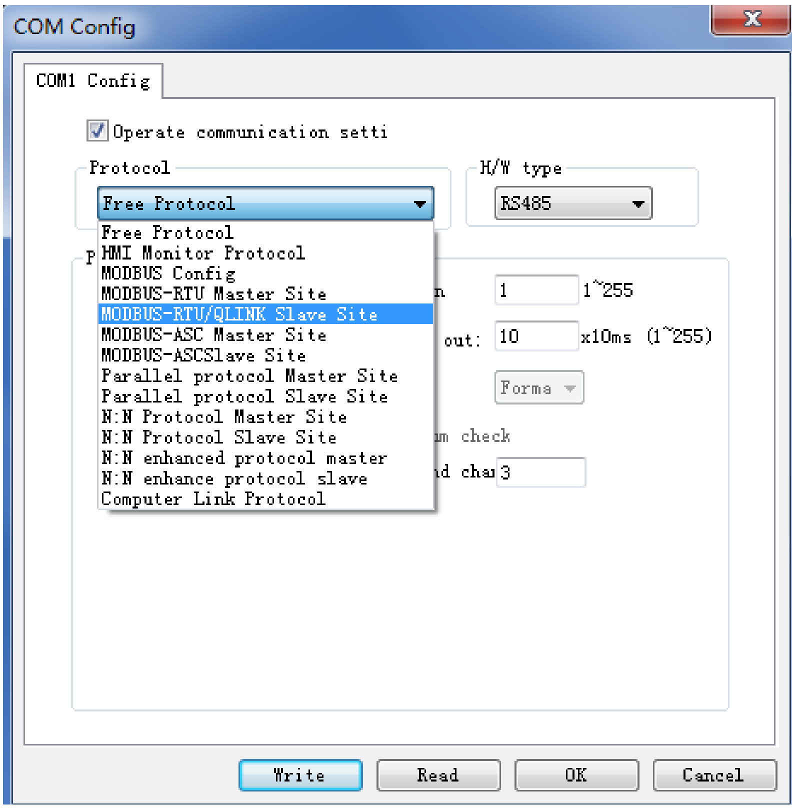
In some scenarios, as a part of the industrial automation system, the PLC must be monitored over the automatic control network. Typical host computers (for example, DCS and industrial PCs for operation configuration software) serve as monitoring hosts and communicate with the PLC through a Modbus master station protocol. In this case, the PLC needs to communicate with host computers through a Modbus slave station protocol. An H3U PLC has a Modbus-RTU slave station protocol. The protocol can run on both COM0 and COM1.

1) Software configuration for slave station

a) COM0 configuration page



b) COM1 configuration page



9

3) Protocol description

Modbus slave station protocols include the Modbus-RTU protocol (the RTU protocol) and the Modbus-ASC protocol (the ASC protocol). The data link used through the RTU protocol is different from that used through the ASC protocol. The data transmitted through the RTU protocol is original data, while the data transmitted through the ASC protocol has been converted into ASCII codes. In addition, the two protocols differ in the frame structure. In the RTU protocol, frames are differentiated by time: if no data is received in 3.5 bytes of time, the system determines that all data has been transmitted. In the ASC protocol, ":" is the starting character and "\CR\LF(0D0Ah)" is the end character of a frame. The efficiency of communication through the RTU approximately doubles that of communication through the ASC protocol. For details, see documents about standard Modbus protocols, which can be downloaded online.

9.4.3 Modbus Parameter Numbers and Data Addressing

When the H3U PLC serves as a Modbus slave station, it supports parameter numbers of Modbus protocols: 0x01, 0x03, 0x05, 0x06, 0x0f, and 0x10. With these parameter numbers, read-write coils are indicated by M, S, T, C, X (read-only), and Y variables; registers are indicated by D, T, and C variables.

When accessing (reading or modifying) internal variables of the slave PLC, the Modbus master must comply with the following definitions of communication command frames and methods of variable addressing.

1) Modbus frame format (Modbus-RTU)

a) Parameter number 0x01 (01): Reads coils.

Request frame format: slave address + 0x01 + head address of coils + number of coils + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x01 (parameter number)	1	The instruction is to read coils.
3	Head address of coils	2	Upper bits are followed by lower bits. See "Coil addressing".
4	Number of coils	2	Upper bits are followed by lower bits (N).
5	CRC code	2	Upper bits are followed by lower bits.

9

Response frame format: slave address + 0x01 + number of bytes + state of coils + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x01 (parameter number)	1	The instruction is to read coils.
3	Number of bytes	1	The value is (N + 7)/8.
4	State of coils	(N + 7)/8	Eight coils are indicated by one byte. If the last byte has less than eight bits, enter 0 for undefined bits. The first eight coils are indicated by the first byte, and the coil with the smallest address is indicated by the least significant bit.
5	CRC code	2	Upper bits are followed by lower bits.

Incorrect response: See "g) Incorrect response frame".

NOTE

- N: The H3U PLC can read a maximum of 255 coils at a time.

b) Parameter number 0x03 (03): Reads registers.

Request frame format: slave address + 0x03 + head address of registers + number of registers + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x03 (parameter number)	1	The instruction is to read registers.
3	Head address of registers	2	Upper bits are followed by lower bits. See "Register addressing".
4	Number of registers	2	Upper bits are followed by lower bits (N).
5	CRC code	2	Upper bits are followed by lower bits.

Response frame format: slave address + 0x03 + number of bytes + register value + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x03 (parameter number)	1	The instruction is to read registers.
3	Number of bytes	1	The value is N x 2
4	Register value	N x 2	One register value is indicated by two bytes. Upper bits are followed by lower bits. The register with a smaller address is indicated by the first byte.
5	CRC code	2	Upper bits are followed by lower bits.

Incorrect response: See "g) Incorrect response frame".

NOTE

- N: The H3U PLC can read a maximum of 125 registers at a time.

c) Parameter number 0x05 (05): Writes a single coil.

Request frame format: slave address + 0x05 + address of the coil + state of the coil + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x05 (parameter number)	1	The instruction is to write a single coil.
3	Address of the coil	2	Upper bits are followed by lower bits. See "Coil addressing".
4	State of the coil	2	Upper bits are followed by lower bits. For example, FF00 is valid.
5	CRC code	2	Upper bits are followed by lower bits.

Response frame format: slave address + 0x05 + address of the coil + state of the coil + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x05 (parameter number)	1	The instruction is to write a single coil.
3	Address of the coil	2	Upper bits are followed by lower bits. See "Coil addressing".
4	State of the coil	2	Upper bits are followed by lower bits. For example, FF00 is valid.
5	CRC code	2	Upper bits are followed by lower bits.

Incorrect response: See "g) Incorrect response frame".

d) Parameter number 0x06 (06): Writes a single register.

Request frame format: slave address + 0x06 + address of the register + register value + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x06 (parameter number)	1	The instruction is to write a single register.
3	Address of the register	2	Upper bits are followed by lower bits. See "Register addressing".
4	Register value	2	Upper bits are followed by lower bits. A non-zero value. For example, is valid.
5	CRC code	2	Upper bits are followed by lower bits.

9

Response frame format: slave address + 0x06 + head address of the register + register value + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x06 (parameter number)	1	The instruction is to write a single register.
3	Address of the register	2	Upper bits are followed by lower bits. See "Register addressing".
4	Register value	2	Upper bits are followed by lower bits. A non-zero value. For example, is valid.
5	CRC code	2	Upper bits are followed by lower bits.

Incorrect response: See "g) Incorrect response frame".

e) Parameter number 0x0f (15): Writes multiple coils.

Request frame format: slave address + 0x0f + head address of coils + number of coils + number of bytes + state of coils + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x0f (parameter number)	1	The instruction is to write multiple coils.
3	Head address of coils	2	Upper bits are followed by lower bits. See "Coil addressing".
4	Number of coils	2	Upper bits are followed by lower bits. The maximum of N is 1968.
5	Number of bytes	1	The value is (N + 7)/8.
6	State of the coil	(N + 7)/8	Eight coils are indicated by one byte. If the last byte has less than eight bits, enter 0 for undefined bits. The first eight coils are indicated by the first byte, and the coil with the smallest address is indicated by the least significant bit.
7	CRC code	2	Upper bits are followed by lower bits.

Response frame format: slave address + 0x0f + head address of coils + number of coils + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x0f (parameter number)	1	The instruction is to write multiple coils.
3	Head address of coils	2	Upper bits are followed by lower bits. See "Coil addressing".
4	Number of coils	2	Upper bits are followed by lower bits.
5	CRC code	2	Upper bits are followed by lower bits.

Incorrect response: See "g) Incorrect response frame".

NOTE

- N: The H3U PLC can write a maximum of 255 coils at a time.

f) Parameter number 0x10 (16): Writes multiple registers.

Request frame format: slave address + 0x10 + head address of registers + number of registers + number of bytes + register value + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x10 (parameter number)	1	The instruction is to write multiple registers.
3	Head address of registers	2	Upper bits are followed by lower bits. See "Register addressing".
4	Number of registers	2	Upper bits are followed by lower bits. The maximum of N is 120.
5	Number of bytes	1	The value is N x 2
6	Register value	N x 2 (N x 4)	
7	CRC code	2	Upper bits are followed by lower bits.

Response frame format: slave address + 0x10 + head address of registers + number of registers + CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	0x10 (parameter number)	1	The instruction is to write multiple registers.
3	Head address of registers	2	Upper bits are followed by lower bits. See "Register addressing".
4	Number of registers	2	Upper bits are followed by lower bits. The maximum of N is 120.
5	CRC code	2	Upper bits are followed by lower bits.

Incorrect response: See "g) Incorrect response frame".

NOTE

- N: The H3U PLC can write a maximum of 123 registers at a time.

g) Incorrect response frame

Incorrect response: slave address + (parameter number + 0 x 80) + error code +CRC code

No.	Definition of Data (Byte)	Number of Bytes	Description
1	Slave address	1	The value ranges from 1 to 247, set by D8121.
2	Parameter number + 0x80	1	It indicates an incorrect parameter number.
3	Error code	1	The value ranges from 1 to 4.
4	CRC code	2	Upper bits are followed by lower bits.

2) Variable addressing

a) Coil addressing

Coil: Bit variables, indicated by 0 or 1. M, S, T, C, X, and Y variables are included.

Variable	Head Address	Number of Coils
M0 to M7679	0x0000 (0)	7680
M8000 to M8511	0x1F40 (8000)	512
SM0 to SM1023	0x2400 (9216)	1024
S0 to S4095	0xE000 (57344)	4096
T0 to T511	0xF000 (61440)	512
C0 to C255	0xF400 (62464)	256
X0 to X377	0xF800 (63488)	256
Y0 to Y377	0xFC00 (64512)	256

b) Register addressing

Register: 16-bit or 32-bit variables. D, T, and C0 to C199 are 16-bit variables; C200 to C255 are 32-bit variables.

Variable	Head Address	Number of registers	Description
D0 to D8511	0 (0)	8512	16-bit register
SD0 to SD1023	0x2400	1024	16-bit register
R0 to R32767	0x3000	32768	16-bit register
T0 to T511	0xF000 (61440)	512	16-bit register
C0 to C199	0xF400 (62464)	200	16-bit register
C200 to C255	0xF700 (63232)	56	32-bit register

◆ **Note:**

When the system accesses 32-bit registers of C200 to C255 through Modbus protocols, one register should be regarded as two because the space occupied by a 32-bit register doubles that occupied by a 16-bit register. For example, if registers of C205 to C208 will be read or written, the Modbus address is 0xF70A (0xF700 + 10), and the number of registers is 8 (4 x 2).

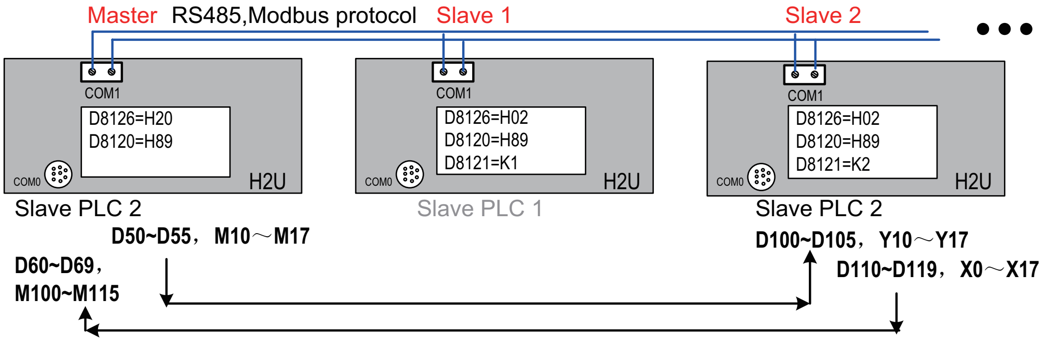
The parameter number (0x06) to write a single register is inapplicable to 32-bit registers.

9.4.4 Programming Method for Communication Among Multiple PLCs Through Modbus Protocols

For a system where two or more PLCs communicate with each other and run in parallel, programming through Modbus protocols is easy and flexible. For combinations of multiple devices (for example, PLC+MDI), programming through Modbus protocols is convenient.

Modbus communication is in "one master+multiple slaves" mode. The master station initiates data exchanges, and all slave stations passively receive and respond to requests. Communication programs are mainly written in the master station. During communication programming for slave stations, you only need to configure communication protocol, communication format, and station number, and process communication data.

Example I: As shown in the following figure, the master PLC sends (D50 to D55, M10 to M17) to slave PLC 2, and PLC 2 returns (D100 to D105, Y10 to Y17) to the master PLC; PLC 2 sends (D110 to D119, X0 to X17) to the master PLC, and the master PLC returns (D60 to D69, M100 to M115) to PLC 2.



◆ **Programming method**

When configuring COM1 of the master PLC, select a Modbus master station protocol, set the baud rate to 9600 bps, and select the 8N2 format. The master PLC exchanges (reads and writes) data.

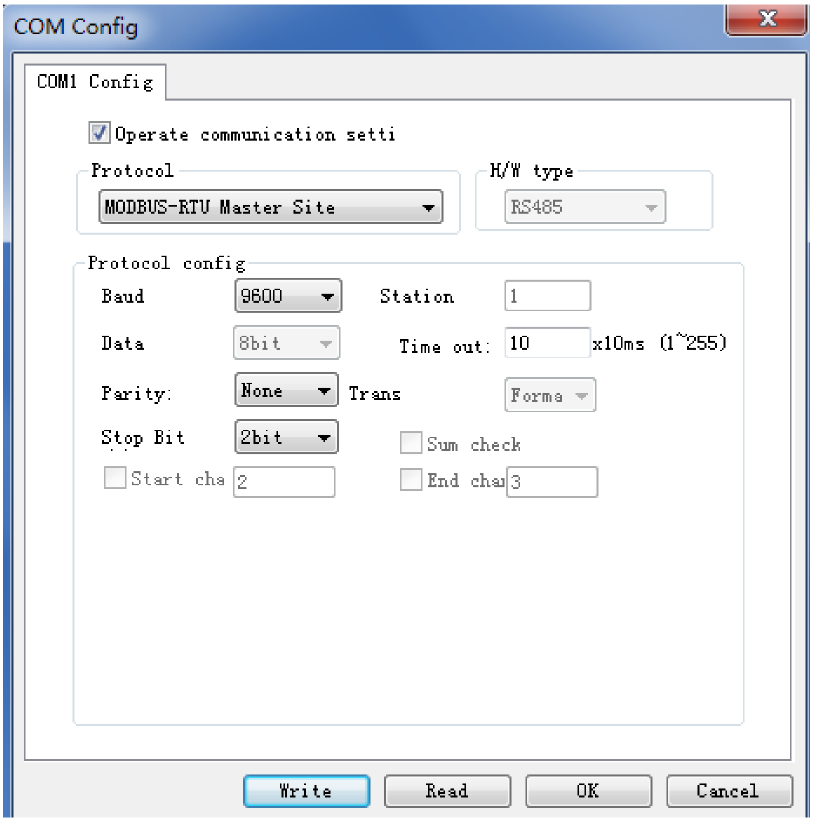
X, Y, and M bit variables are integrated into D variables, and then exchanged in batches within a range of consecutive D variables. Master and slave PLCs combine and parse variables. In this way, data is exchanged efficiently and programming is easy.

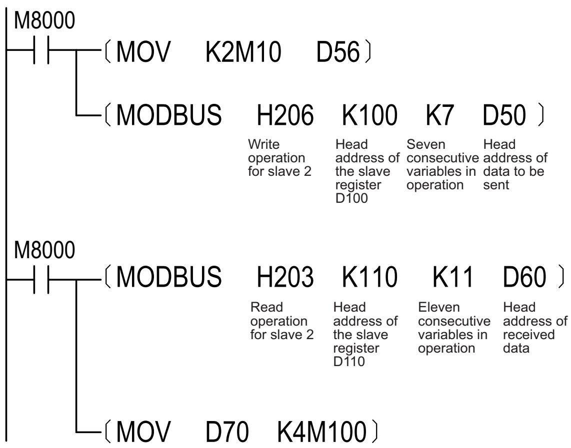
When variables M10 to M17 of the master station are integrated into D56, the master station will send seven D variables: D100 to D106. When variables X0 to X17 of a slave station are integrated into D120, the master station will read 11 D variables: D110 to D120.

During programming for the master station with Modbus instructions, data is exchanged efficiently, and less registers are occupied.

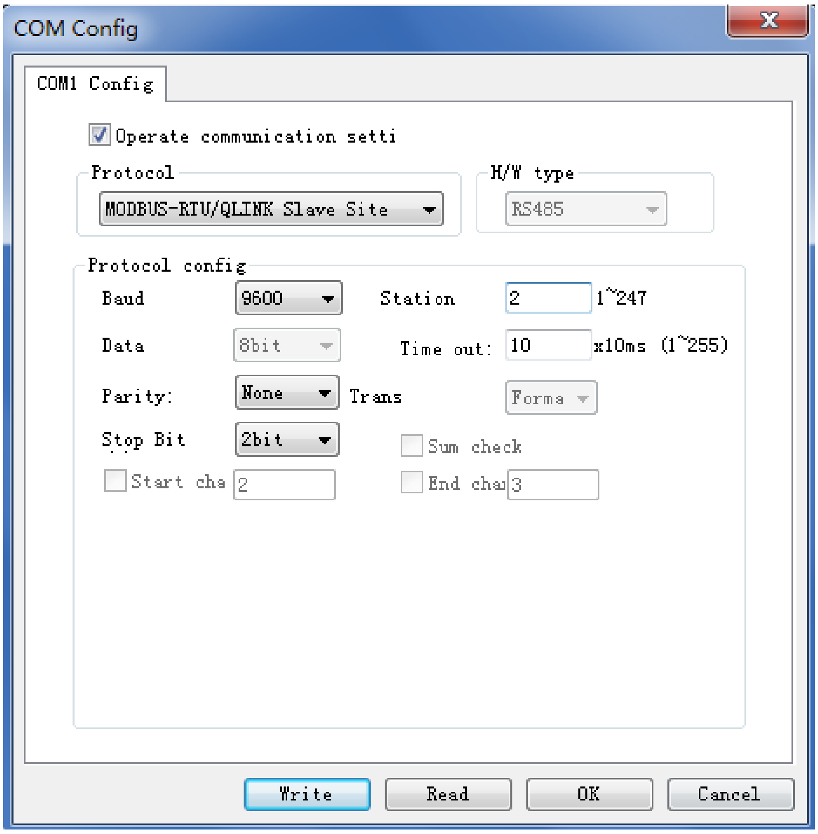
Configure COM1 through AutoShop. Select the Modbus-RTU master station protocol, set baud rate to 9600 bps, set the data length to 8-bit, set the odd/even parity check to none, and set the number of stop bits to 2.

9





When configuring COM1 of PLC 2, select the Modbus-RTU slave station protocol, set the baud rate to 9600 bps, select the 8N2 format, and set the station number to 2. Refresh registers to be read by the master station, and maintain the data written in the master station through communication.



This programming method is applicable to other slave stations.
Note: A station number cannot be reused.

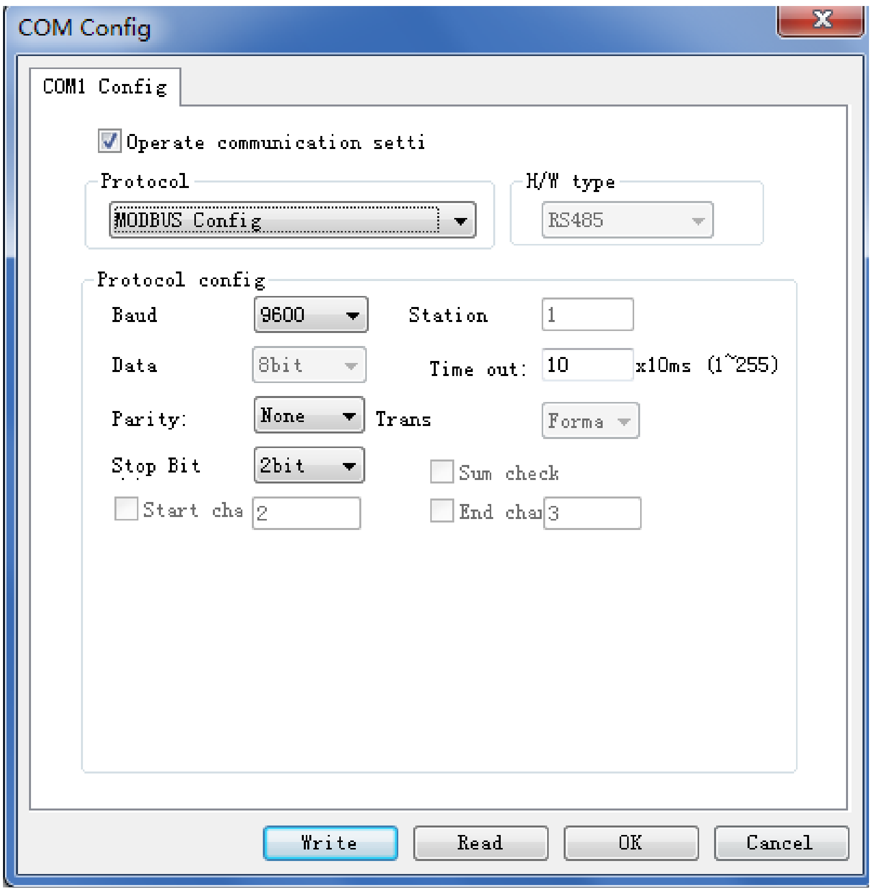
9.5 Modbus Configuration and Usage

Programming with Modbus instructions is flexible, and user programs are easy to understand. However, if a slave station is disconnected during communication, the program scan duration will be affected, which will affect control and may even cause scan timeout alarms. Configuration tables solve this problem.

Define communication content and data storage units in a Modbus configuration table, and download the table together with user programs to the PLC. When the PLC executes user programs, system software will automatically conduct Modbus master station communication. The following describes operations during programming.

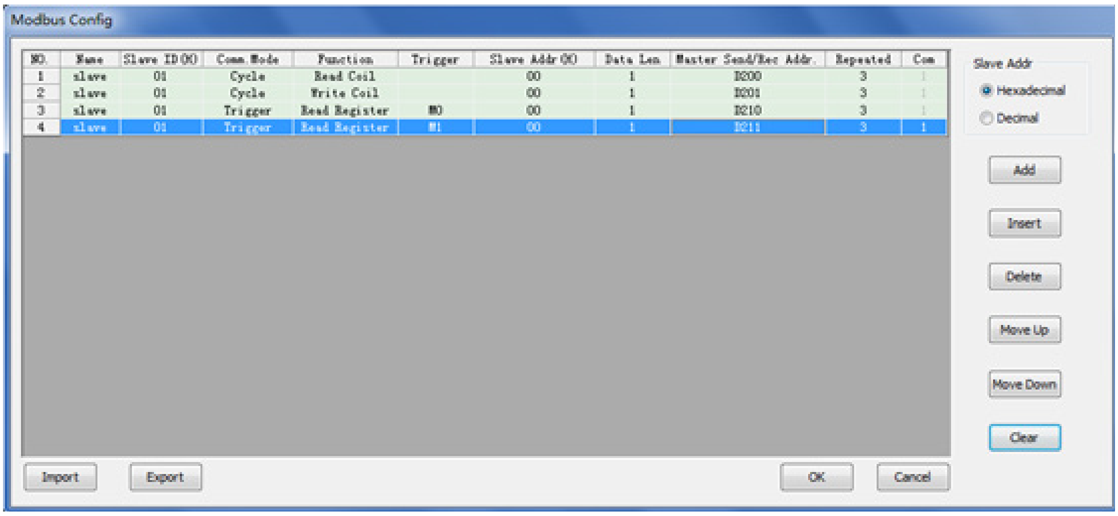
Configure the specified communication port by selecting a Modbus master station protocol and setting the communication format. Complete the configuration table based on data frequency, address, and trigger condition. Refresh D variable, trigger M flags, and use the received D variables for calculation. The master PLC regularly checks the state of each Modbus slave station, determines impact of communication faults on the system, and gives alarms or shuts down the system.

9.5.1 Protocol Setting Method for Modbus Configuration Table



9.5.2 Setting the Modbus Configuration Table

In the **Project Manager** window, double-click **Modbus Configuration**.



You can click **Add** to add communication configuration items. The data in the table can be edited. All items are operands required by instructions shown in the Modbus ladder chart, which must be entered based on desired operations and D variables. Click **OK**. When downloading user programs, download Modbus configurations.

1) Tips and suggestions for completing the Modbus configuration table

Slave station number (H) and register address (H) are hexadecimal.

Example 1: If the register address is 18, the station number is 12.

Example 2: To access the AC drive parameter number F0-24, enter F018 as the slave register address.

Two communication modes are available: cyclic and triggered modes. It is recommended that you classify data based on the usage frequency.

a) Cyclic communication

To repeatedly and quickly read and write variable data of slave stations (for example, operating frequency and state of AC drives, and state of input ports), or to change the operating frequency of AC drives and state of output ports in real time in process control systems, you can select cyclic communication. When the PLC executes user programs, it repeatedly scans and executes all items for cyclic communication in the configuration table.

b) Triggered communication

To regularly read or write stable data of slave stations (for example, output current and output power of AC drives, and active fault alarms), you can select triggered communication. When the trigger flag is set, the corresponding items in the configuration table will be triggered. You can regularly set the flag to read or write data.

2) Suggestions for setting the communication mode

Select the communication mode based on the data update frequency so that communication can be significantly improved. Do not select the cyclic mode for all communication operations to simplify programming; otherwise, data may not be exchanged timely because of too many cyclic operations, which will affect control. Select the triggered mode for unimportant access because triggering communication in order of priority can significantly improve timeliness.

As the baud rate of Modbus communication through the RS-485 interface is normally 9600 bps, the number of "cyclic" items is limited to 10, and the number of "triggered" items per second is limited to 10. In this case, data can be exchanged timely.

3) Suggestions for setting M variables

If you select the triggered mode, you need to enter the trigger condition in AutoShop. An M bit element can be used as a trigger condition. When the bit element is set to ON, communication is triggered. Then the system automatically clears the trigger flag. Therefore, the M flag can also be used to determine whether communication is triggered. Therefore, when setting the communication configuration table, do not use one M variable as the trigger flag for multiple communication operations; otherwise, other communication operations may be triggered because the system clears the M flag.

4) Types of Modbus communication operations

In the **Function** column, you can enter types of operations: reading registers, writing registers, reading coils, and writing coils. Registers are indicated by word variables (16-bit), and coils are indicated by bit variables (1-bit variables, indicated by 0 or 1). You need to enter commands based on the type of variables.

a) Entering the slave register address

Before accessing internal variables of slaves, you need to understand rules for defining slave register addresses. The following describes common address algorithms and precautions when PLCs, AC drives, or servos serve as slave stations.

b) Register address of a slave PLC

It is the register address of a slave PLC when multiple PLCs are connected through Modbus protocols.

c) Address of a PLC register indicated by word variables

Word variables are 16-bit (word) or 32-bit (double-word) variables. D, T, and C0 to C199 are 16-bit variables; C200 to C255 are 32-bit variables. The following table lists head addresses of registers (register address = head address + variable number).

Variable	Head Address	Number of Registers	Description
D0 to D8511	0x0000 (0)	8512	16-bit register
SD0 to SD1023	0x2400 (9216)	1024	16-bit register
R0 to R32767	0x3000 (12288)	32768	16-bit register
T0 to T255	0xF000 (61440)	256	16-bit register
C0 to C199	0xF400 (62464)	200	16-bit register
C200 to C255	0xF700 (63232)	56	32-bit register

NOTE

- Note: When the system accesses 32-bit registers of C200 to C255 through Modbus protocols, one register should be regarded as two because the space occupied by a 32-bit register doubles that occupied by a 16-bit register. For example, if registers of C205 to C208 will be read or written, the Modbus address is 0xF70A (0xF700 + 10), and the number of registers is 8 (4 x 2).

d) Address of a PLC (register) indicated by bit variables

Bit variables of a PLC are also called "coils", for example, M, S, T, C, X, and Y variables, which are indicated by 0 or 1. The following table lists head addresses of registers (register address = head address + variable number).

9

Variable	Head Address	Number of Coils
M0 to M7679	0 (0)	7680
M8000 to M8511	0x1F40 (8512)	512
SM0 to SM1023	0x2400 (9216)	1024
S0 to S4095	0xE000 (57344)	4096
T0 to T511	0xF000 (61440)	512
C0 to C255	0xF400 (62464)	256
X0 to X377	0xF800 (63488)	256
Y0 to Y377	0xFC00 (64512)	256

5) Register address of a slave AC drive and precautions

For details about register addresses of slave AC drives, see Appendix B of the **MD500 Series General-Purpose AC Drive User Manual** (visit www.inovance.cn to download the latest version), in which parameter number addresses, state, start/stop control, frequency instructions, and alarms are defined. Parameter numbers of AC drives are accessed through registers. The address of a parameter number corresponds to the parameter number group number. Take FX-yy as an example: "X" is hexadecimal, and "yy" is decimal. When calculating the register address, the system converts "yy" into hexadecimal "YY", so the corresponding address is hexadecimal FXYY. The uppermost bits of the hexadecimal address of a U parameter number are indicated by 7XYY, as listed in the following table.

Parameter Number Group	Address to Be Read (HEX)	Address to Be Permanently Changed (HEX)	Address to Be Temporarily Changed (HEX)
F0-00 to FE-29	F000 to FE1D	F000 to FE1D	0000 to 0E1D
A0-00 to AC-27	A000 to AC1B	A000 to AC1B	4000 to 4C1B
U0-00 to U0-65	7000 to 7041	Unchangeable	Unchangeable

Parameter numbers of AC drives and servos are stored in internal flash drive, and can be retained upon power failure. The flash drive does not limit the number of Read operations but limits that of Write operations to 100,000. If the limit is exceeded, hardware may be damaged. Therefore, modify parameter numbers of AC drives and servos in "triggered" mode rather than in "cyclic" mode.

For AC drive parameter numbers to be frequently modified, AC drives only allow addresses of parameter numbers in random access memory (RAM) to be modified. In this way, the PLC will temporarily modify parameter numbers in RAM, which is valid for AC drives and will not trigger modification of the flash drive.

9.6 CANlink Communication

H3U PLCs support CAN communication. The main PLC module supports CANlink and CANopen networks. You can switch from one protocol to another through M8280. When M8280 = OFF, CANlink3.0 is enabled; when M8280 = ON, CANopen is enabled. To switch from one protocol to another, you need to power off then on the PLC or switch it from STOP to RUN.

9.6.1 Principle of CANlink3.0 Communication

CANlink3.0 communication is implemented through CAN configuration rather than CAN communication instructions. When downloading user programs, you need to download CAN configurations to the PLC.

Understanding the principle of CANlink3.0 network configuration can help you complete the CAN configuration table.

On a CANlink3.0 network, one master station must exist, which can be an H3U, H2u-XP, or H1u-XP PLC.

On a CANlink3.0 network, one or more slave stations must exist, which can be H2U IO/AI/AO/AM/PT/TC remote extension modules, MD AC drives with CANlink3.0 interface cards, IS servo drivers with CANlink3.0 interface cards, H3U/H2U-XP/H1U-XP PLCs, or devices developed based on the CANlink3.0 protocol.

Master and slave stations on a CANlink3.0 network communicate with each other by automatically sending and writing data rather than in query-response mode.

Example:

- To send data to slave stations, the master station "writes" register data in slave registers based on CANlink communication configurations when trigger conditions are met.
- Slave stations automatically send data to the master station and "write" the data in the receiving unit of the master station based on CANlink communication configurations.
- Slave stations automatically send data to each other and "write" the data in receiving units of slave stations based on CANlink communication configurations.
- To send data to multiple stations, a station automatically sends "Write operation" data to itself (equivalent to broadcasts), while the other stations selectively receive the data and automatically store it in their receiving units.
- For efficient data exchange during network communication, master and slave stations save "heard" broadcast data sent by other stations. You need to click **Receiving Configuration** to set receiving slave station numbers and addresses. In this way, the stations configured as receiving stations will ignore the broadcast data from stations not configured as sending stations.

You do not need to configure CANlink3.0 slave stations because CANlink configurations can be transmitted to slave stations through an H3U, H2U-XP, or H1U-XP master PLC. Therefore, CANlink3.0 communication configuration items for slave stations are forwarded by the CANlink master station through configuration frames.

Upon startup, the master station sends configuration frames to CANlink slave stations and assigns the list of communication tasks. Slave stations automatically send data based on the list.

CANlink3.0 configuration items include address of the sent register, address of the target receiving slave station, number of data entries, address of the received register, interval for sending, and trigger condition, which are required by common communication instructions. Different from common communication operations, "communicate-write" operations do not need responses.

In communication scenarios where multiple slaves must synchronously act and respond (for example, servo-driven synchronous multi-axes control and position-controlled high-speed movement), you need to set **Synchronous Write** for the master station. The master station writes data from slave stations and then sends broadcast command frames to make slave stations run simultaneously.

9

9.6.2 CANlink Network

1) Hardware interface

An H3U host has a CAN hardware interface with the following pins.

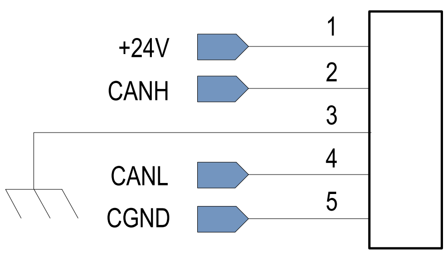
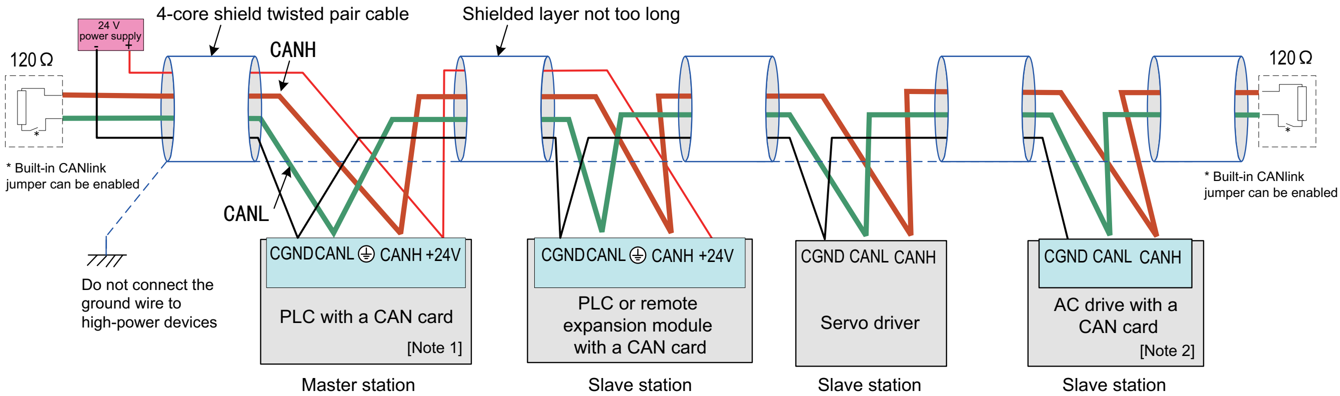


Diagram of CANlink interface definition

Definition of CANlink interface pins

Pin No.	Signal	Description
1	+24 V	External power supply +24 V DC
2	CANH	Positive CAN bus
3	PGND	Shield ground wire, connected to the shield layer of the communication cable
4	CANL	Negative CAN bus
5	CGND	External power supply -24 V DC

The five wires of each device must be interconnected to form a CAN. An external 24 V AC power supply must be provided between pin 1 (＋24 V) and pin 5 (CGND). 120 Ω resistors must be provided at both sides of the CAN bus. The CAN bus wiring diagram is as follows.



[Note 1] Select a CAN card based on the PLC type.

[Note 2] Select a CAN card based on the AC drive type.

Wiring diagram of a CAN network formed by multiple devices

NOTE

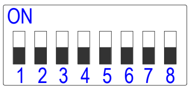
- Recommended CAN communication cable manufacturer: Shenzhen LianJiaXiang Science & Technology Co., Ltd. Model: RVVP 2 × 2 × 0.5

2) DIP switch

DIP switches of H3U-3232MT/R and H3U-0808PMRTA models differ from that of the H3U-1616MT/R-XP model in design and usage.

a) H3U-3232MT/R and H3U-0808PMRTA models

H3U-3232MT/R and H3U-0808PMRTA models have 8-digit DIP switches. The following describes the definition of an 8-bit DIP switch.



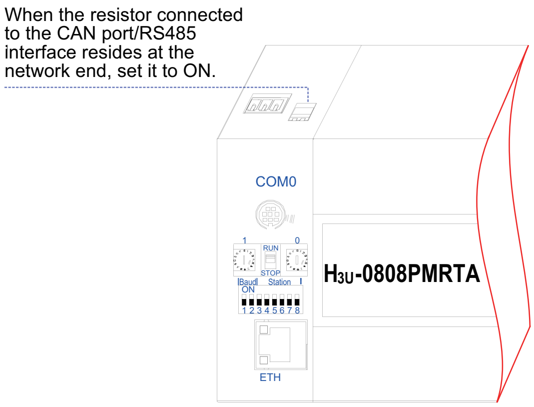
CAN DIP switch (H3U-3232MT/R and H3U-0808PMRTA models)

Definition of CAN DIP switch (H3U-3232MT/R and H3U-0808PMRTA models)

No.	Signal	Description	
1	Baud rate combination bit 1	bit 1 and bit 0	00 - 500 Kbps
			01 - 100 Kbps
2	Baud rate combination bit 0		10 - 1 Mbps
			11 - 50 Kbps
3	Address pin A5	The digits of the 6-bit DIP switch form a binary numeral in descending order, which indicates the station number (for the main PLC module, the station number can be set through D elements). ON indicates 1, and OFF indicates 0. A5 is the most significant bit, and A0 is the least significant bit: A5A4A3A2A1A0. For example, if A0 is ON and the others are OFF, the address is 000001 in binary format, K01 in decimal format, and h01 in hexadecimal format. If A3 and A4 are ON, and the others are OFF, the address is 011000 in binary format, K24 in decimal format, and h18 in hexadecimal format.	
4	Address pin A4		
5	Address pin A3		
6	Address pin A2		
7	Address pin A1		
8	Address pin A0		

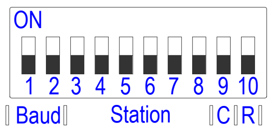
Note: If the DIP switch is changed, the baud rate and address will not take effect immediately. You need to power off then on the device or switch it from STOP to RUN.

H3U-3232MT/R and H3U-0808PMRTA models have CAN build-out resistors. The following figure shows DIP switches.



b) H3U-1616MT/R-XP model

The CAN address of the H3U-1616MT/R-XP model overlaps the fourth segment of the Ethernet IP address. The following describes how to use a DIP switch.



CAN DIP switch (H3U-1616MT/R-XP model)

9

CAN DIP switch definition (H3U-1616MT/R-XP model)

Baud Rate: 2 Bits			Station Number and IP Address: 6 Bits						CAN	RS485
Baud rate	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
No.	1	2	3	4	5	6	7	8	9	10
500 K	0	0	Binary: The station number overlaps the fourth segment of the IP address. For example, if bit 7 to bit 2 are 011011, the station number and the fourth segment of the IP address are both 27.						Terminal resistor switch: Bit 0: RS-485 communication Bit 1: CAN communication	
100 Kbps	0	1								
1 Mbps	1	0								
50 Kbps	1	1								

3) Relationship between distance and baud rate

Baud Rate	Distance (m)	Minimum Cable Diameter (mm^2)	Maximum Number of Access Points
1000 Kbps	30	0.3	18
500 Kbps	80	0.3	32
250 Kbps	150	0.3	63
125 Kbps	300	0.5	63
100 Kbps	500	0.5	63
50 Kbps	1000	0.7	63

4) CANlink communication software variables

Table of special CANlink communication variables

M Element	M Element Definition	D Element	D Element Definition
M8280	Protocol flag[1] 0: CANlink VER3.00 protocol 1: CANopen protocol	D8280	CANlink protocol version (300) CANopen protocol version (100)
M8281	Reserved	D8281	Reserved
M8282	Reserved	D8282	CANlink heartbeat
M8283	Valid address for CANlink online monitoring	D8283	Head address for CANlink online monitoring
M8284	0: CAN address set by the DIP switch and displayed by D8284 1: CAN address set by D8284	D8284	CAN address setting/display※1
M8285	0: Baud rate set by the DIP switch and displayed by D8285 1: Baud rate set by D8286※1	D8285	Valid Baud rate displayed
M8286	The CANlink Synchronous Write error can be cleared after setting. The element is automatically reset after the error is cleared.	D8286	10: 10 Kbps 20: 20 Kbps 50: 50 Kbps 100: 100 Kbps 125: 125 Kbps 250: 250 Kbps 500: 500 Kbps 800: 800 Kbps 1000: 1 Mbps

M Element	M Element Definition	D Element	D Element Definition
M8287	Reserved	D8287	Number of the station with CANopen configuration errors
M8288	Reserved	D8288	CANopen configuration error number
M8289	Reserved	D8289	CAN bus error count (upper eight bits indicate received errors and lower eight bits indicate sent errors)
M8290	CANlink start/stop element	D8290	Number of frames received by CAN per second
M8291	CANlink Synchronous Send trigger element	D8291	Total number of frames sent and received by CAN per second
M8292	Reserved	D8292	Reserved
M8293	Reserved	D8293	Reserved
M8294	Reserved	D8294	Reserved
M8295	Reserved	D8295	Reserved
M8296	Reserved	D8296	Reserved
M8297	Reserved	D8297	Reserved
M8298	Reserved	D8298	Reserved
M8299	Reserved	D8299	Reserved
M8300	Reserved	D8300	Reserved
M8301	Reserved	D8301	Reserved
M8302	Reserved	D8302	Reserved
M8303	Reserved	D8303	Reserved
M8304	Reserved	D8304	Reserved
M8305	Reserved	D8305	Reserved
M8306	Reserved	D8306	Reserved
M8307	Reserved	D8307	CANlink Synchronous Write error
M8308	Reserved	D8308	CANlink configuration error
M8309	Reserved	D8309	Reserved

9

NOTE

- [1]: You need to power off and then on the device or switch it from STOP to RUN so that the flag can be identified.

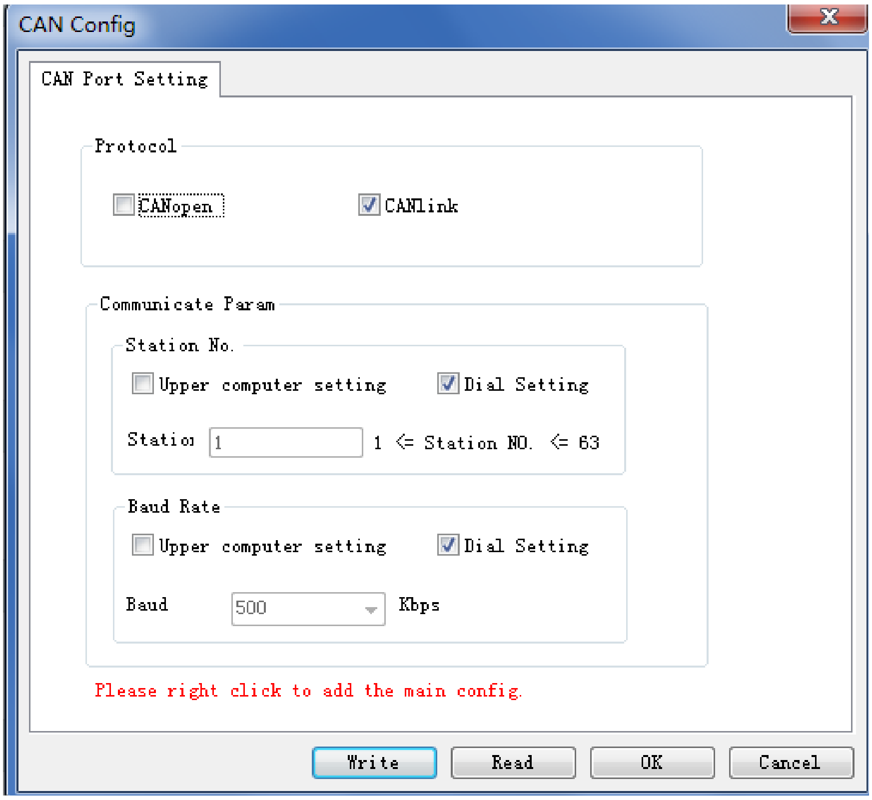
9.6.3 CANlink Software Configuration

Take the following steps to configure a CANlink network.

- Configure the CANlink network through AutoShop and define the data to be exchanged.
- Download configurations to the H3U PLC.
- Enable the CANlink network on the PLC and other programmable devices. Unless otherwise indicated, the CANlink network is enabled in the H3U extension module by default. In addition, on a PLC with CANlink configuration, configure the CANlink network through the user program.

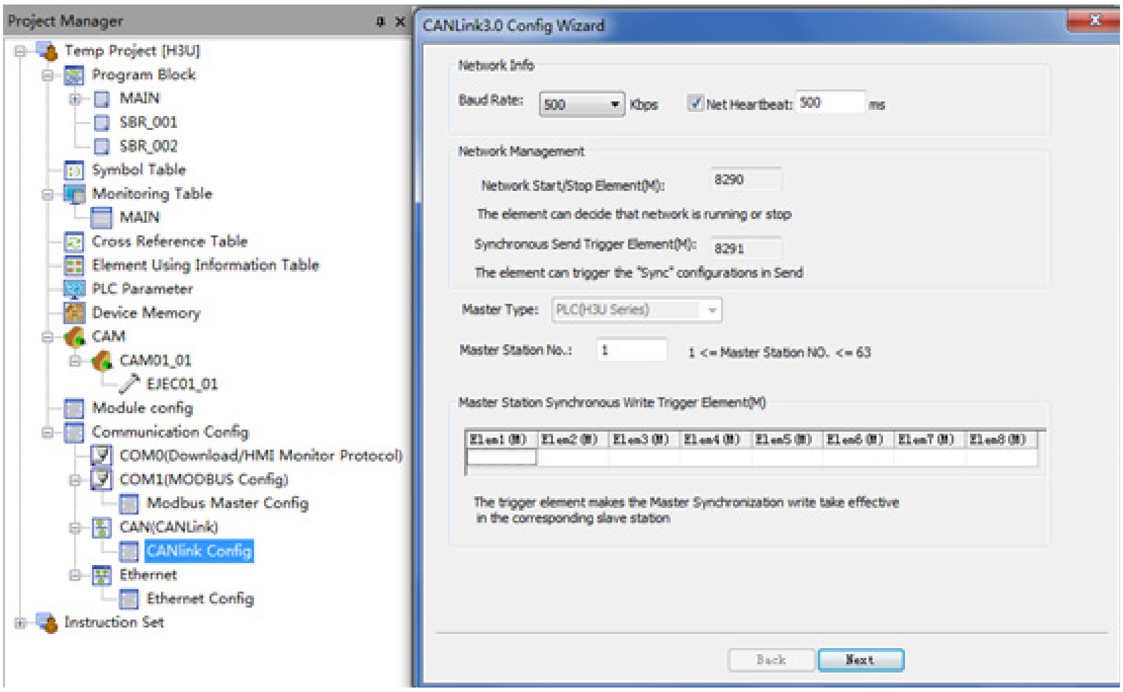
1) Master CANlink station configuration

a) Choose **Project Manager** > **Communication Config**. Double-click **CAN Port**. The following window is displayed.



Select **CANlink** and click **OK**.

b) The **CANlink Config** icon is displayed. Double-click **CANlink Config**. The **CANlink3.0 Config Wizard** page is displayed, as shown in the following figure.



- **Baud Rate** (required)

Eight options are available for different scenarios: 20 Kbps, 50 Kbps, 100 Kbps, 125 Kbps, 250 Kbps, 500 Kbps, 800 Kbps, and 1 Mbps. You can select the desired option in the drop-down box, and then download the configuration to the master station (this parameter is valid for the master station only, and needs to be manually modified on a slave station). You can select the baud rate based on the bus load and communication distance.

- **Net Heartbeat** (optional)

All slave stations send heartbeats to the master station at a specified interval. The master station monitors the state (online or offline) of each slave station through the heartbeat mechanism. Slave stations monitor the state of the master station through its heartbeats. (It is recommended that the interval be more than 200 ms.) If you deselect this parameter, the heartbeat function is disabled and the system cannot monitor the network.

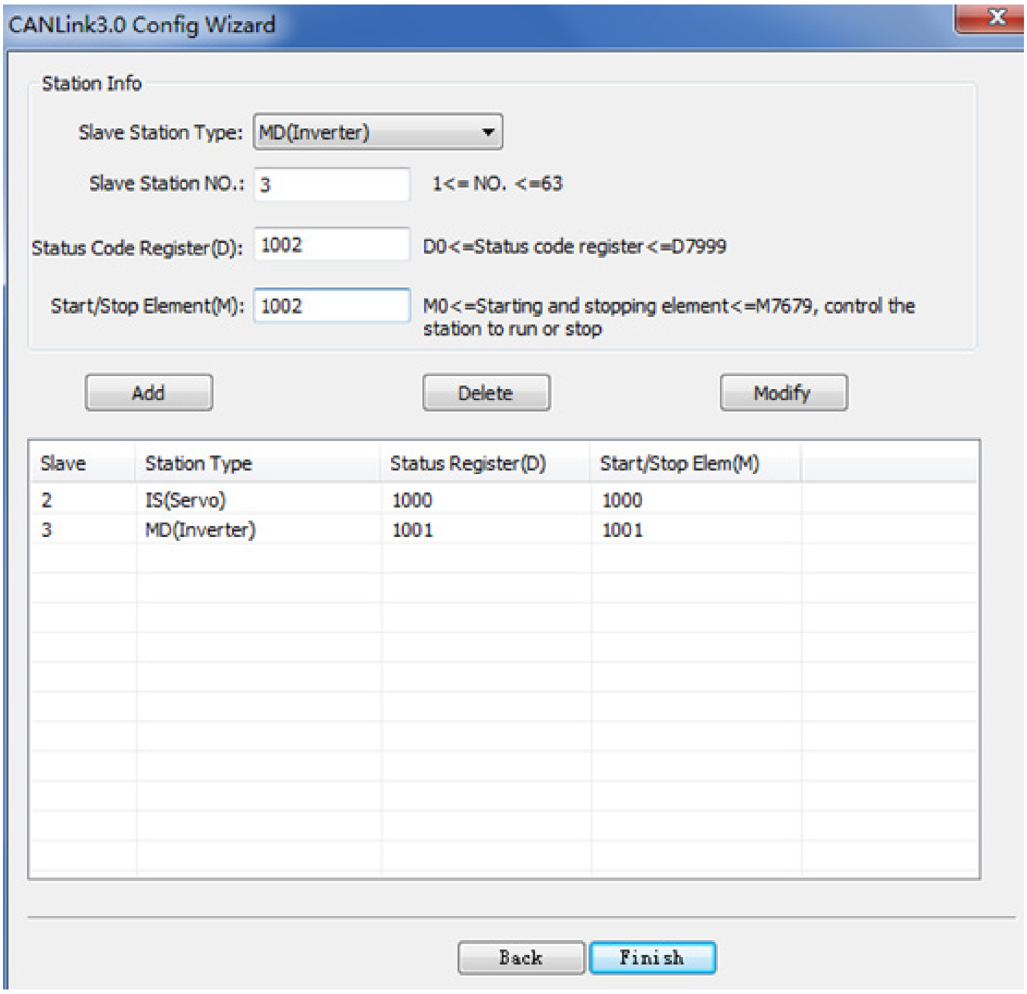
- **Master Station No.** (required)

In this example, the master station number is the number of the PLC that serves as a master station. The number cannot be changed. If the number entered is inconsistent with the actual number, the PLC will determine that the downloaded configuration is invalid. For example, if you enter 7, the configuration is valid only when downloaded to station 7. Station 7 then assigns the configuration to other stations. The CANlink network configuration is downloaded to the master station and then assigned to slave stations. In this way, the system can monitor and manage the entire network through the master station in the background.

- **Master Station Synchronous Write Trigger Element** (optional)

It is an element triggering Synchronous Write for the master station. When a trigger element (M) is set, the corresponding configuration takes effect. The element is automatically reset after data is sent.

2) Click Next. The following window is displayed.



- **Add**

After configuring a slave station, click **Add**. The station is added to the list.

- **Delete**

Select a station, click **Delete**, and then click **OK** (you can select multiple stations at a time).

- **Modify**

Select a station, modify parameters, and then click **Modify**. Do not modify the station type.

- **Slave Station No.**

Set the CANlink slave station number.

- **Status Code Register (D)**

It is used to save the status of a slave station fed back through heartbeat frames of the slave station.

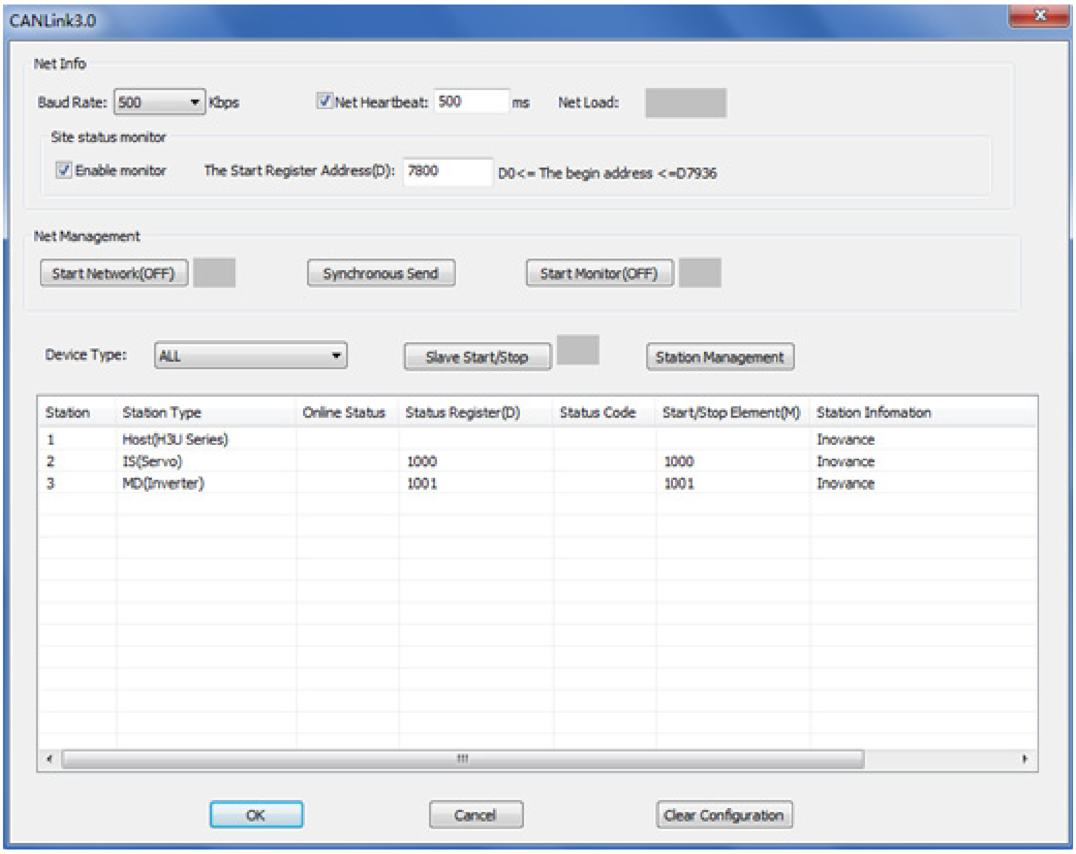
- **Start/Stop element (M)**

It is an M element used to start or stop communication. When M = ON, communication is started; when M = OFF, communication is stopped.

NOTE

- In the **CANlink3.0 Config Wizard** window, you can click **Finish** to save modifications and exit, or click **X** in the upper right corner to cancel modifications and exit.

3) The following window is displayed.



a) Net Info

Baud Rate: Indicates the baud rate of the master station.

Net Heartbeat: The heartbeat function is enabled by selecting this parameter.

Net Load: Computes the real-time load of the network (this parameter is displayed only when the network load is monitored during running of devices).

① Network load ≤ 50: Green (good)

② 50 < Network load ≤ 75: Yellow (warning)

③ 75 < Network load ≤ 90: Red (major warning)

④ Network load > 90: ERR, red background (error)

- Site status monitor

After **Enable monitor** is selected, the online state of the slave station will be updated to the corresponding D soft element (head address of the monitored register + station number). The default head address is D7800. For example, if station 2 is to be monitored, the state will be displayed in D7802.

Register Status	Definition
1	Configurations of the slave station are available.
2	The slave station is running.
5	The slave station is disconnected.

NOTE

- If the heartbeat function is not enabled, station state monitoring is disabled.

b) Net Management

① Start Network (OFF) (enabled when monitoring is enabled): Starts and stops network communication.

② Synchronous Send: Synchronization will be triggered. You can enable the function in the user program by setting M8291. M8291 will be automatically reset after synchronous data frames are sent.

③ Start Monitor (OFF): Starts and stops network monitoring.

c) Device Type: Filters displayed stations.

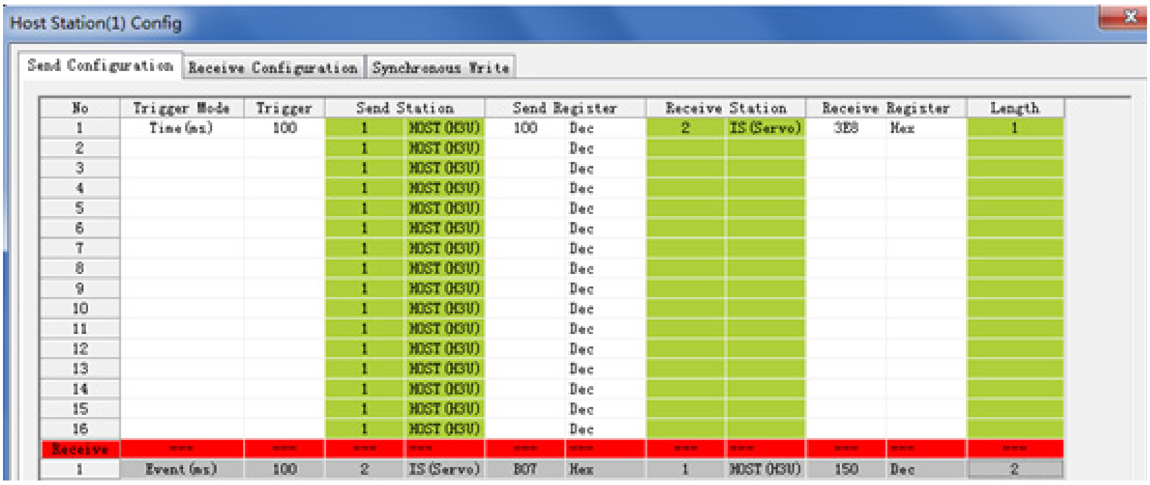
① Slave Start/Stop: Select a slave station and control start/stop of communication.

② Station Management: Click **Station Management**. The initialization wizard page is displayed. You can modify parameters of the master or slave station.

③ Station configuration:

Double-click a station. The communication configuration window is displayed. Communication configuration includes sending configuration, receiving configuration, and synchronization configuration (for the master station only).

- Sending configuration



No	Trigger Mode	Trigger	Send Station		Send Register		Receive Station		Receive Register		Length
1	Time(ms)	100	1	HOST(H3U)	100	Dec	2	IS(Servo)	3E8	Hex	1
2			1	HOST(H3U)		Dec					
3			1	HOST(H3U)		Dec					
4			1	HOST(H3U)		Dec					
5			1	HOST(H3U)		Dec					
6			1	HOST(H3U)		Dec					
7			1	HOST(H3U)		Dec					
8			1	HOST(H3U)		Dec					
9			1	HOST(H3U)		Dec					
10			1	HOST(H3U)		Dec					
11			1	HOST(H3U)		Dec					
12			1	HOST(H3U)		Dec					
13			1	HOST(H3U)		Dec					
14			1	HOST(H3U)		Dec					
15			1	HOST(H3U)		Dec					
16			1	HOST(H3U)		Dec					
Receive	---	---	---	---	---	---	---	---	---	---	---
1	Event(ms)	100	2	IS(Servo)	807	Hex	1	HOST(H3U)	150	Dec	2

- Trigger Mode

① Time (ms): It is applicable to all devices. The station applies the configuration at a fixed interval. The value ranges from 1 to 30,000.

② Event (M): It is applicable to the host and PLC. The station applies the configuration when the trigger condition (M) is set. An event can be triggered by the same M element. The element is automatically reset after data is sent. Edge trigger instructions must be used to operate M elements; otherwise, the network load will be excessive.

③ Synchronization (M): It is applicable to all devices. The master station applies the configuration when M8291 is set. The element is automatically reset after data is sent.

④ Event (ms): It is applicable to IS, MD, and remote extension module (TCM/NTCM). The station applies the configuration when it detects the changed value of sent register and the trigger condition (disabling time) is met.

NOTE

- The disabling time indicates the minimum interval for sending the same configuration.
- Maximum number of configuration items for one station: 256 for the host (master station), 16 for one slave station, and 256 for all slave stations.

9

- If you select a configuration item and press **Insert**, an empty configuration line will be added following the item. If you select a configuration item and press **Delete**, the item will be deleted. In addition, you can press shortcut keys or right-click an item to copy-paste or delete it, and insert or delete a line.

● Register

Host and PLC register values correspond to D elements. IS and MD register values correspond to parameter numbers. TCM/NTCM corresponds to BFM.

● Number of registers

It is the number of sent or received consecutive D elements or parameter numbers.

● Point-to-multipoint configuration

When a sending station is also a receiving station, the station applies the point-to-multipoint configuration, in which no receiving station is specified. If you enter the sending station number into the receiving configuration table, the configured station can receive data sent by the sending station. The received register is the D element or parameter number corresponding to the receiving station.

● Received data

The entries in the gray background indicate data received from other stations, including point-to-point and point-to-multipoint data. You can see which element or parameter number of which station will affect the configured station.

● Receiving configuration

Receiving configuration applies to receiving point-to-multipoint data from other stations. Each station can receive point-to-multipoint data from eight stations.

Example

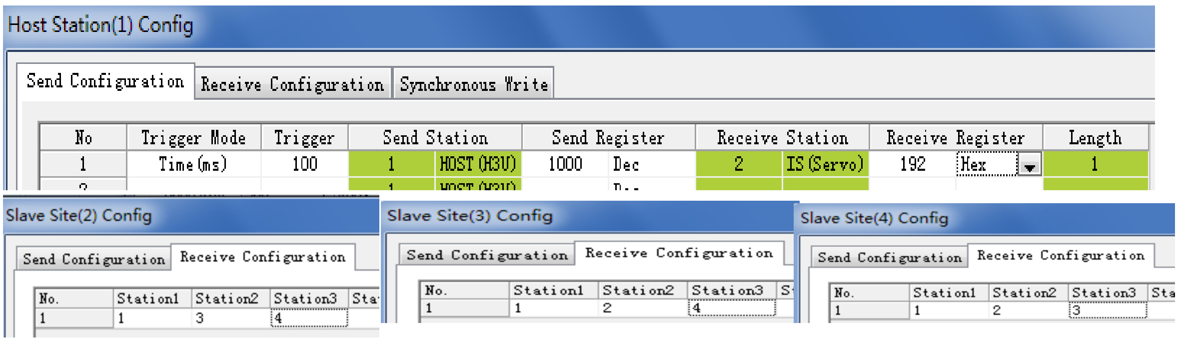


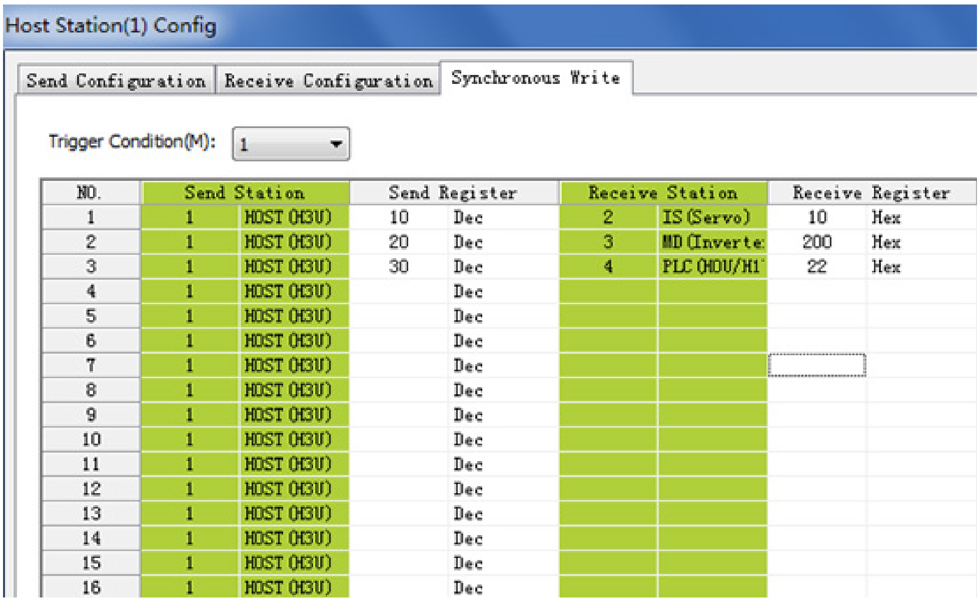
Diagram of receiving configuration

In this example, the master station (station 1) sends the D1000 value in the form of a point-to-multipoint frame to the receiving station D192 at an interval of 100 ms. Based on the receiving configuration of slave stations 10, 20, and 30, slave stations 10 and 20 will receive and write the frame in D192, while station 30 will ignore the frame.

NOTE

- Point-to-multipoint configuration enables data to take effect simultaneously. It is equivalent to master station synchronization configuration; however, the master station is not the only sending station. Each station can receive point-to-multipoint data from a maximum of eight stations. However, the number of stations receiving point-to-multipoint data sent by each station is not limited. That is, all stations other than the sending station can receive data if the sending station number is entered into the receiving configuration table.

- Synchronous Write configuration for the master station



When the trigger condition (M) is set, the Synchronous Send configuration for the master station takes effect. You can select different trigger conditions (M) to display, add, modify, or delete synchronization configurations. Synchronization configuration is applicable to scenarios in which an operation needs to be initiated synchronously.

As shown in the figure, when M1 = 1, the master station sends the three configuration items successively. Upon receipt of the items, slave stations store them in the buffer. After the last data entry is sent, the master station automatically sends a configuration application command. Upon receipt of the command, all slave stations automatically write the data in the buffer in corresponding elements or parameter numbers. As shown in the figure, PLC 10 writes the D10 value in D10, servo 20 writes the D20 value in H200, and AC drive 30 writes the D30 value in HF003. All these values are synchronously written when slave stations receive the configuration application command. After the command is sent, the master station automatically resets the trigger element M1. Edge trigger instructions must be used to operate M elements; otherwise, the network load will be excessive.

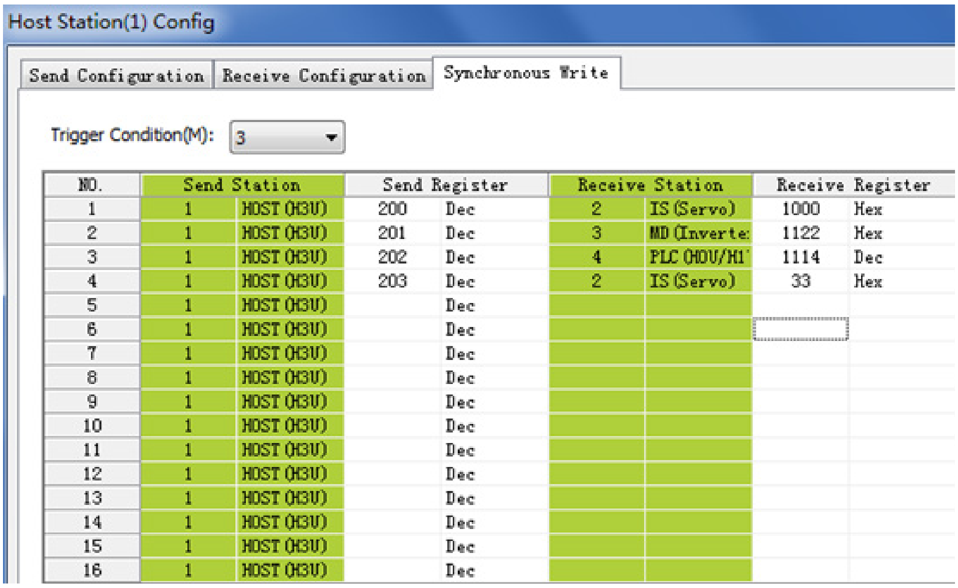
NOTE

- Trigger condition (M): Each trigger condition associates a maximum of 16 configuration items. It determines whether the associated synchronization configuration is valid. A maximum of eight trigger conditions (M) are allowed. You can select a trigger condition in the drop-down box. During synchronization configuration of a 32-bit servo register, data must correspond to upper 16 bits and lower 16 bits respectively for the same trigger element. That is, two data entries must be written for one trigger element, one corresponding to upper address bits of the 32-bit parameter number, and the other corresponding to lower address bits. If only one entry is written or two entries are written for two trigger elements respectively, the servo will return an error, and the configuration cannot continue.

- Example of 32-bit servo register synchronization configuration

As shown in the following figure, H1112 is a 32-bit parameter number of the servo. During configuration of the parameter number, two data entries must be written, corresponding to upper and lower address bits respectively. When M3 is set, the master station writes D201 and D202 values in H1112. When all of the five data entries are sent, the master station sends a command to enable the slave stations and apply the configurations. Then M3 is automatically reset.

9



NO.	Send Station		Send Register		Receive Station		Receive Register	
1	1	HOST (H3U)	200	Dec	2	IS (Servo)	1000	Hex
2	1	HOST (H3U)	201	Dec	3	MD (Inverte	1122	Hex
3	1	HOST (H3U)	202	Dec	4	PLC (H0U/H1'	1114	Dec
4	1	HOST (H3U)	203	Dec	2	IS (Servo)	33	Hex
5	1	HOST (H3U)		Dec				
6	1	HOST (H3U)		Dec				
7	1	HOST (H3U)		Dec				
8	1	HOST (H3U)		Dec				
9	1	HOST (H3U)		Dec				
10	1	HOST (H3U)		Dec				
11	1	HOST (H3U)		Dec				
12	1	HOST (H3U)		Dec				
13	1	HOST (H3U)		Dec				
14	1	HOST (H3U)		Dec				
15	1	HOST (H3U)		Dec				
16	1	HOST (H3U)		Dec				

If only one address is processed for one trigger element, the servo will return an error so that synchronization cannot continue. The error will be recorded in D8307 of the master station. Error codes are listed in Section 9.11.6.

- Device type

This item can be used to filter displayed stations.

- Master station error codes and processing

① The following table lists configuration errors and causes. The register address is D8303.

Table of configuration errors

Error Code※	Cause	Solution
XX00	Reserved	None
XX01	Incorrect code	Check whether the internal definition is correct.
XX02	Incorrect index	Check whether the device type is correct.
XX03	Incorrect information	Check whether the address is valid and check the read-write attribute.
XX04	Reserved	Reserved
XX05	Incorrect data length	Check whether the data length is beyond the limit.
XX06	Configuration frames failing to respond within a specified time	Check whether the connection is normal.

② The following table lists abnormality codes and causes. The register address is D8307.

Table of abnormalities

Error Code※	Cause	Solution
XX00	Reserved	Reserved
XX01	Invalid command code	Check whether the internal definition is correct.
XX02	Abnormal address	Check whether the address is normal or whether the address can be accessed.
XX03	Abnormal data	Check whether the data is within a specified range.
XX04	Invalid operation	Check whether the operation is authorized.
XX05	Invalid length	Check whether the data length is beyond the limit.
XX06	Responding timeout	Check whether the connection is normal.

NOTE

- The codes are displayed in decimal format. "XX" indicates the station number, which means that an error occurs when station **XX** is configured or when a command is sent to station **XX**.
- Different from error codes for the master station, an error code for a slave PLC does not include the station number.

9.6.4 Examples of Slave Station (Servo and AC Drive) Access

Currently, Inovance PLCs (H3U, H2U-XP, and H1U-XP), Inovance AC drives (MD380 and MD500), Inovance servos (IS620P), and remote extension modules support CANlink3.0.

1) Servo driver access

Parameter numbers for servo CANlink communication

Parameter Number	Name	Setting Range	Unit	Default	Effective Time	Type	Mode
H0C-00	Servo axis address	1 to 247	1	1	Upon power-on again	Running setting	PST
H0C-08	Setting of baud rate for CAN communication	0: 20 Kbps 1: 50 Kbps 2: 100 Kbps 3: 125 Kbps 4: 250 Kbps 5: 500 Kbps 6: 1 Mbps 7: 1 Mbps	1	5	Upon power-on again	Running setting	PST
H0C-13	Writing the updated parameter number in the EEPROM	0: No 1: Yes	1	1	Immediately	Running setting	PST

9

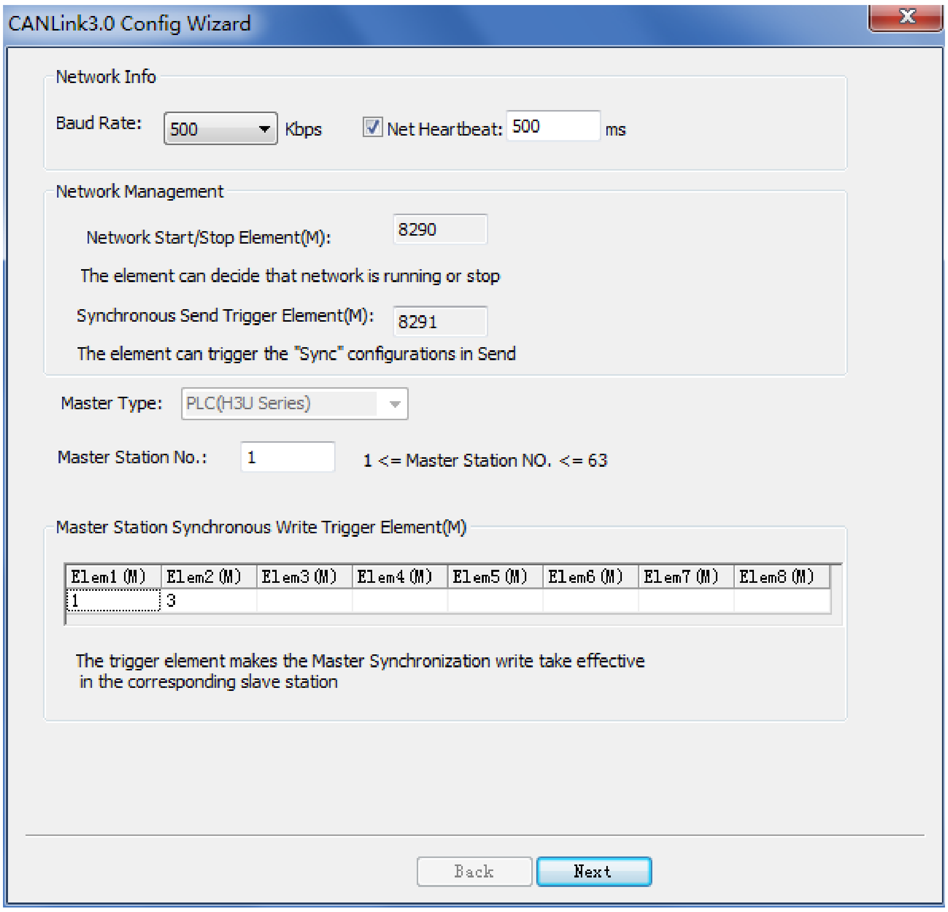
Parameter Number	Name	Setting Range	Unit	Default	Effective Time	Type	Mode
H0C-15	CAN communication protocol selection	0: CANLink protocol 1: Reserved (CANopen protocol)	1	0	Upon power-on again	Stop setting	PST

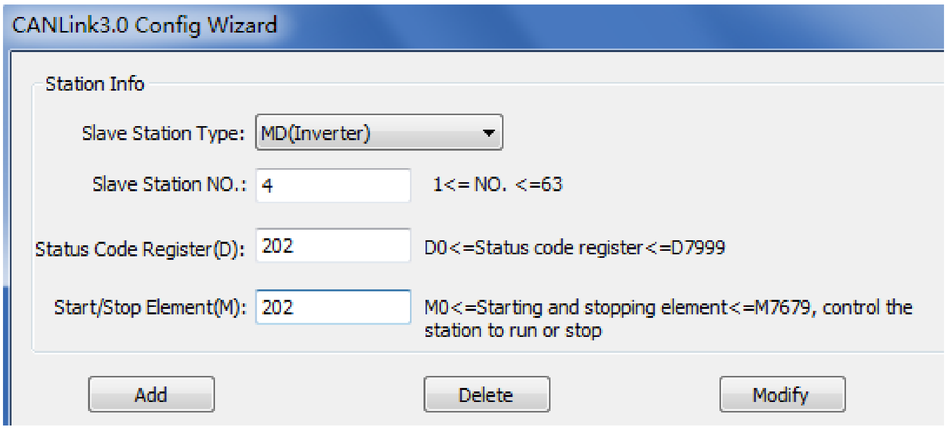
NOTE

- The group number (the first two digits) of a servo parameter number does not change, and the parameter number (the last two digits) needs to be converted from decimal to hexadecimal format. For example, H08.22 ("22" is decimal) is converted into H0816 ("16" is hexadecimal) in PLC programs. You need to configure the servo address (H0C.00) and baud rate (H0C.08), and select a CAN protocol (H0C.15). Some servo parameter numbers take effect only after you power off and then on the device.

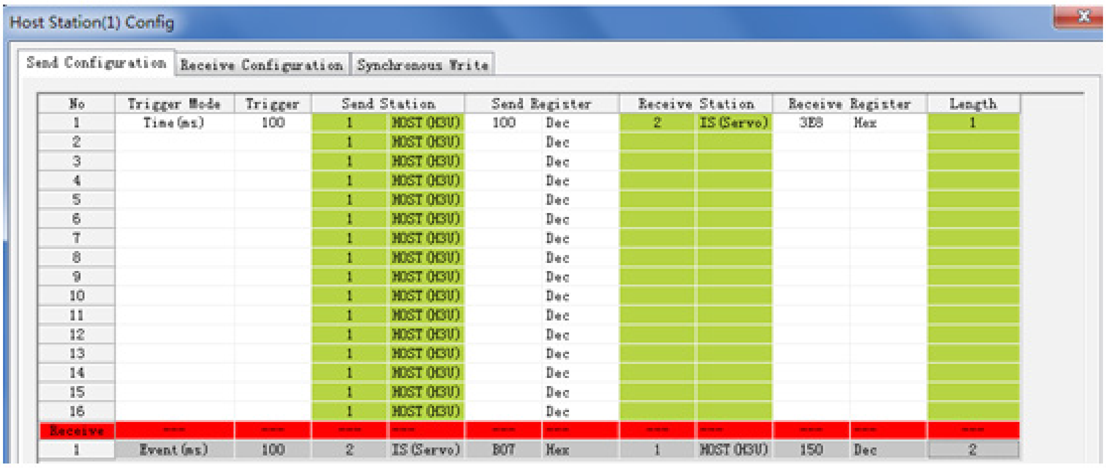
2) AC drive access

One H3U PLC, one MD380 AC drive, and one H2U-4DAR extension module form a CANlink network. The AC drive runs under control of the PLC for 20 seconds, and then stops for 20 seconds. The process repeats. The H2U-4DAR PLC provides the control voltage. H1U-XP station number is 1. The H2U-4DAR station number is 2, and the MD380 station number is 3. As only a few stations exist and the communication distance is only 10 m, set the baud rate to 500 Kbps and the network heartbeat to 500 ms (default). Create a project "CANlink3.0 Example". Double-click **CAN(CANLink)**. The following figure is displayed.

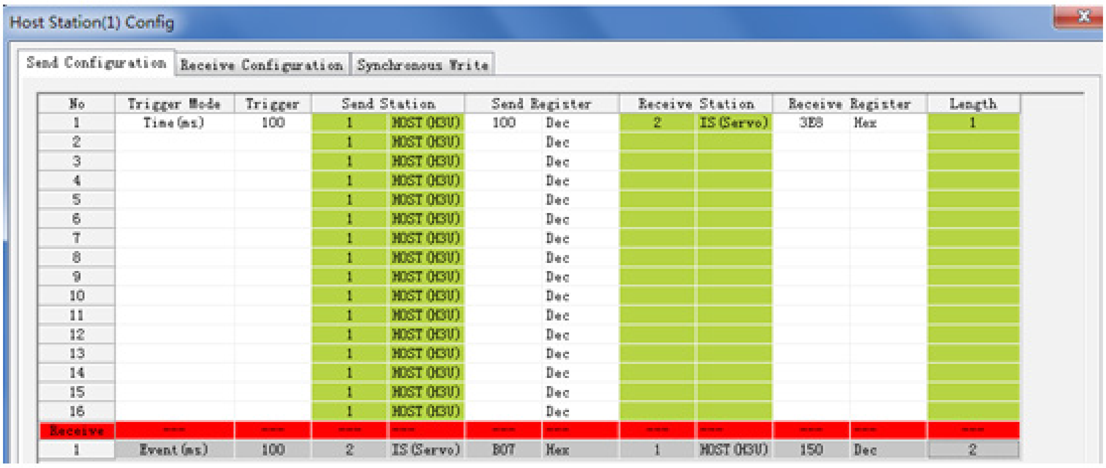




Configure the AC drive first. Fd-02 is 3 (station number), the number at the thousands place of Fd-00 is 5 (baud rate), F0-02 is 2 (communication command channel), and F0-03 is 3 (the master frequency source X is AI2). The following figure shows how the master station controls the MD380 AC drive.

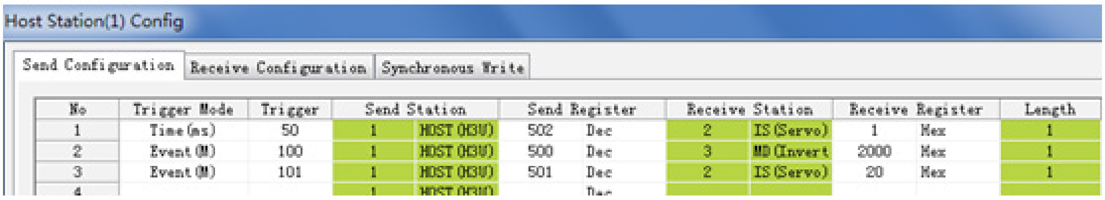


The AC drive returns to the current state, as shown in the following figure.



The AC drive returns the H3000 value to the master station. The value is stored in D2000. When the H3000 value changes, it will be returned. The minimum interval for sending the value is 100 ms.

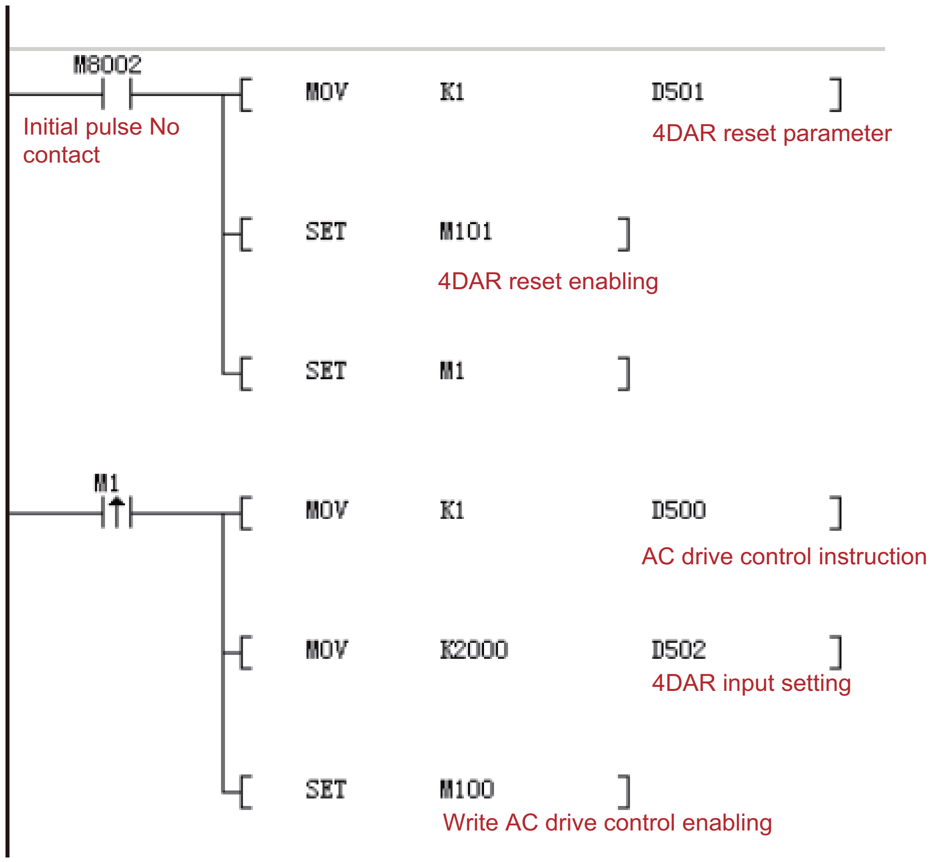
Remote module configuration: When M101 is set, BFM#20 of the H2U-4DAR PLC is written (module reset register). Outputs are sent every 50 ms.

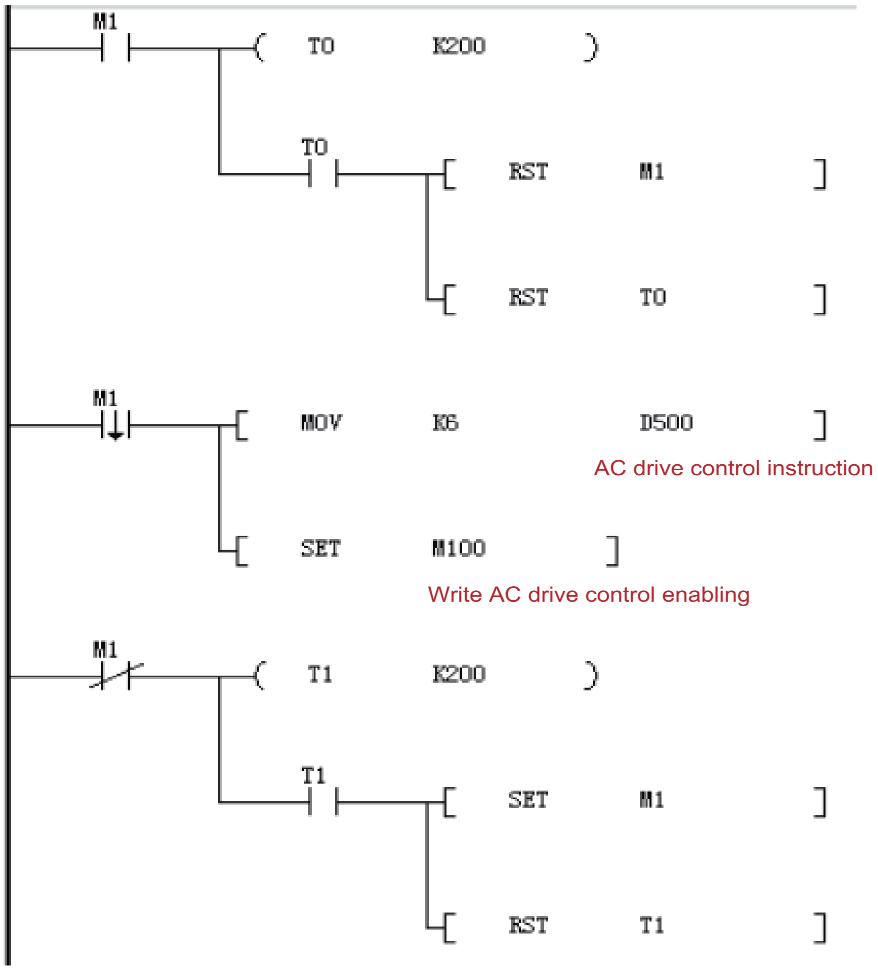

Host Station(1) Config

Send Configuration | Receive Configuration | Synchronous Write

No	Trigger Mode	Trigger	Send Station		Send Register		Receive Station		Receive Register		Length
1	Time (ms)	50	1	HOST (H3U)	502	Dec	2	IS (Servo)	1	Hex	1
2	Event (M)	100	1	HOST (H3U)	500	Dec	3	MD (Invert	2000	Hex	1
3	Event (M)	101	1	HOST (H3U)	501	Dec	2	IS (Servo)	20	Hex	1
4			1	HOST (H3U)		Dec					

The following figure shows the application program.





9.6.5 Troubleshooting for CANlink Communication

Indicator	State	Indication
Communication (green)	Off	CANlink bus not connected or disconnected
	On	CANlink bus connected (remote frames received on the node)
	Blinking (≤ 3 Hz)	During CANlink communication, one blink per frame of bus data sent or received
	Blinking (5Hz)	Flag monitor
Fault (red)	Off	No fault
	On	Monitor timeout (node), no node (monitor)
	Blinking (0.5Hz)	CANlink configuration error (for the configurator)
	Blinking (1Hz)	Node lost or crash (for the monitor)
	Blinking (5Hz)	CANlink address conflict

1) Checking whether CANlink3.0 is supported

Device	Check
PLC	Check the D8280 value. If D8280 = 300, CANlink3.0 is supported; otherwise, CANlink3.0 is not supported.
AC drive/servo	Check the software version. For details, see the user manual.

2) Checking the build-out resistor

Power off all devices. Use a multimeter to measure the resistance between CANH and CANL. The resistance should be about 60 Ω. If the resistance is too small, there are build-out resistors incorrectly connected at other locations. In this case, disconnect these build-out resistors. If only one resistor is available, the resistance is about 120 Ω, and the network connection is bad. If no resistor is available, communication fails. Provide build-out resistors between the stations at both ends of the network.

3) Checking the baud rate

Check whether the Baud rate is normal. Power off and then on the device or switch it from STOP to RUN so that the baud rate can take effect.

4) Checking wiring

The CAN communication port and extension module of the PLC are powered by an external 24 V power supply. The AC drive and servo are self-powered. Interconnect CGND pins of all CAN devices to ensure that all devices share one power supply CGND port.

Check whether the communication cable, shielded cable, and power supply are short-circuited.

5) Others

In case of strong interference, reduce the baud rate.

9.7 CANopen Communication

For details about CANopen hardware port connection, CAN build-out resistor connection, and soft elements, see “9.6 CANlink Communication” on page 616.

9.7.1 CANopen Protocol Selection

Set M8280 to 1. Power off and then on the device or switch it from STOP to RUN. When D8280 = 100, switch to the CANopen protocol.

H3U series supports CANopen DS301.

Software Function Module	Slave Station	Master Station
Supported protocol	DS301 V4.02	DS301 V4.02
Maximum number of TPDOs	8	64
Maximum number of RPDOs	8	64
Number of slave station nodes	/	30

Software Function Module	Slave Station	Master Station
Baud rate and communication distance	1 Mbps/25 m 800 Kbps/50 m 500 Kbps/100 m 250 Kbps/250 m 125 Kbps/500 m 50 Kbps/1000 m 20 Kbps/2500 m 100 Kbps 10 Kbps	1 Mbps/25 m 800 Kbps/50 m 500 Kbps/100 m 250 Kbps/250 m 125 Kbps/500 m 50 Kbps/1000 m 20 Kbps/2500 m 100 Kbps 10 Kbps
Soft element for data exchange	SD300 to SD363	D0 to D7999 (configurable)

9.7.2 CANopen Indicators

LED Indicator	CAN RUN (Green)	CAN ERR (Red)
Off	None	No error
On	Operational	Bus disconnected
Blinking slowly (0.8-second cycle)	Pre-operational	Pre-operational
Blinking once quickly (1.2-second cycle)	Stopped	At least one error counter hitting or exceeding the threshold (too many error frames)
Blinking twice quickly (1.6-second cycle)	None	Incorrect control (node protection or heartbeat timeout)

9.7.3 Definitions of CANopen Acronyms

9

NMT: Network Management

Network management includes management of application layer, network state, and node ID allocation. It is implemented in master-slave communication mode. That is, on a CAN, only one NMT master station exists with one or more slave stations. The service is used to control the slave station state.

SDO: Server Data Object

An SDO can access the data in the slave station object dictionary (OD) through index and subindex. SDOs are used for slave station configuration. Each frame of an SDO request must be answered.

PDO: Process Data Object

PDOs are used to transmit real-time data. The data length ranges from one to eight bytes. Data can be transmitted in synchronous and asynchronous modes. PDO frames are primary data exchange frames after slave stations are started.

Sync: Synchronous

Synchronization is implemented in master-slave communication mode. The master sync node regularly sends sync objects, and the sync slave node synchronously executes tasks upon receipt of the objects.

Sync frames are used for synchronous transmission through PDOs.

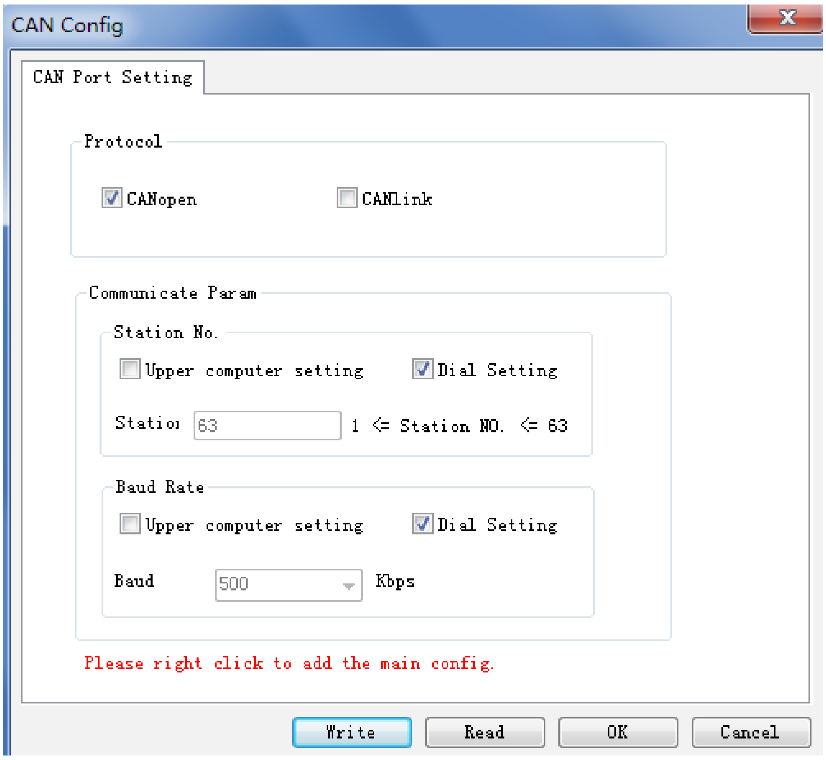
COB-ID: Communication Object Identifier

Each CANopen frame starts with a COB-ID. A COB-ID is not the slave station number. However, it is associated with the slave station number by default.

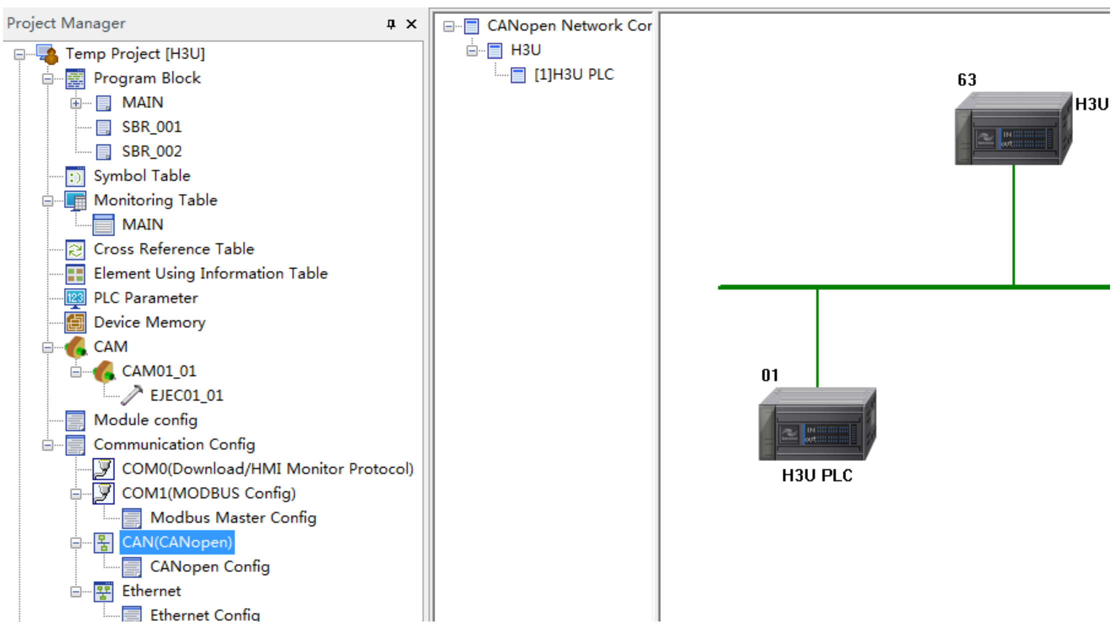
9.7.4 CANopen Configuration

1) Configuring the master station

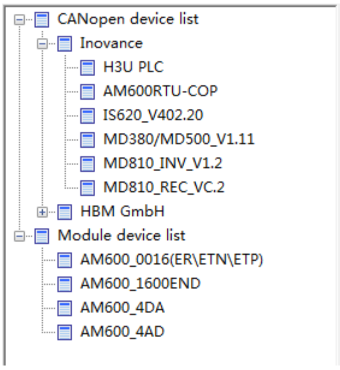
Open AutoShop. On the **Project Manager** page, double-click **CAN**, and select **CANopen**.



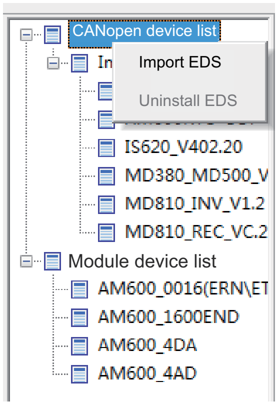
Double-click **CANopen Config**. The following page is displayed.



Double-click a CANopen slave station in the list of CANopen devices or drag a CANopen slave station to the list.

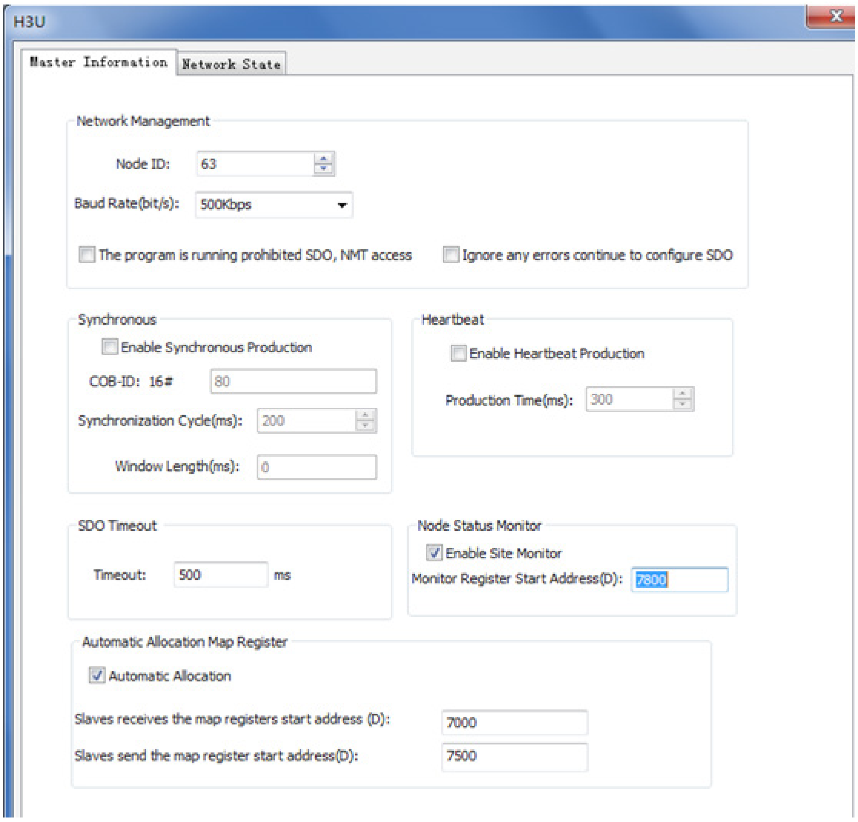


If no slave station is available, right-click the list icon to import the EDS file. The file can be obtained from the manufacturer.



a) Master station information

Double-click the icon of H3U master station. The following window is displayed.



9

① Network Management

Node ID: Indicates the master station number. If the station number is identical to the PLC number, the PLC will be initialized as the CANopen master station. If the station number is different from the PLC number, the PLC will be initialized as a CANopen slave station.

Baud Rate: Indicates the communication baud rate valid for the master station.

The program is running prohibited SDO, NMT access: If this option is selected, online debugging is disabled during running of the program. The function only applies to background software.

Ignore any errors continue to configure SDO: After this option is selected, if SDO configuration errors occur, configuration will continue. The function is valid for all slave stations. If the option is not selected, when SDO errors occur, the master station will reset slave stations through broadcasts.

② Synchronous

Enable Synchronous Production: If this option is selected, the configured station will send a sync frame repeatedly in the set synchronization cycle.

COB-ID: Indicates the ID for sync frame sending. The default value is 0x80. The parameter cannot be configured.

Synchronous Cycle (ms): Indicates the cycle for sync frame sending. The default value is 200, in the unit of milliseconds.

Window Length (ms): The value is 0 by default. The parameter cannot be configured.

NOTE

- On a network, only one sync frame can be sent.

③ Heartbeat

Enable Heartbeat Production: If this option is selected, the configured station will send heartbeat frames repeatedly in the set cycle.

Production Time (ms): Indicates the cycle for heartbeat sending. The default value is 300, in the unit of milliseconds.

NOTE

- By default, the heartbeat monitoring consumption time is 2.5 times the heartbeat generation time. (The heartbeat monitoring timeout is 2.5 times the heartbeat generation time.)

④ SDO Timeout

Timeout: Indicates the SDO waiting time. The default value is 500, in the unit of milliseconds. SDO frames are used for network configuration. If the SDO fails to receive return frames after the third try, the master station determines that configuration times out. The waiting time for each frame is called SDO timeout.

⑤ Node Status Monitor

Enable Site Monitor: If this option is selected, the master station will write the slave station state in the corresponding register. By default, the option is selected.

Monitor Register Start Address: The default value is 7800. That is, D7800 is the head address. D7800 indicates the master station state, and D (7800 + slave station number) indicates the slave station state. The following table lists definitions of state values.

Value	State
0	Initializing
4	Stopped
5	Operational
127	Pre-operational
255	Offline

NOTE

- If no slave station exists, the register will not be updated. For example, if station 3 does not exist, data on D7803 will not be updated.
- To enable this function, you need to configure heartbeat or node protection for the slave station because the node state is fed back by heartbeats or node protection frames.

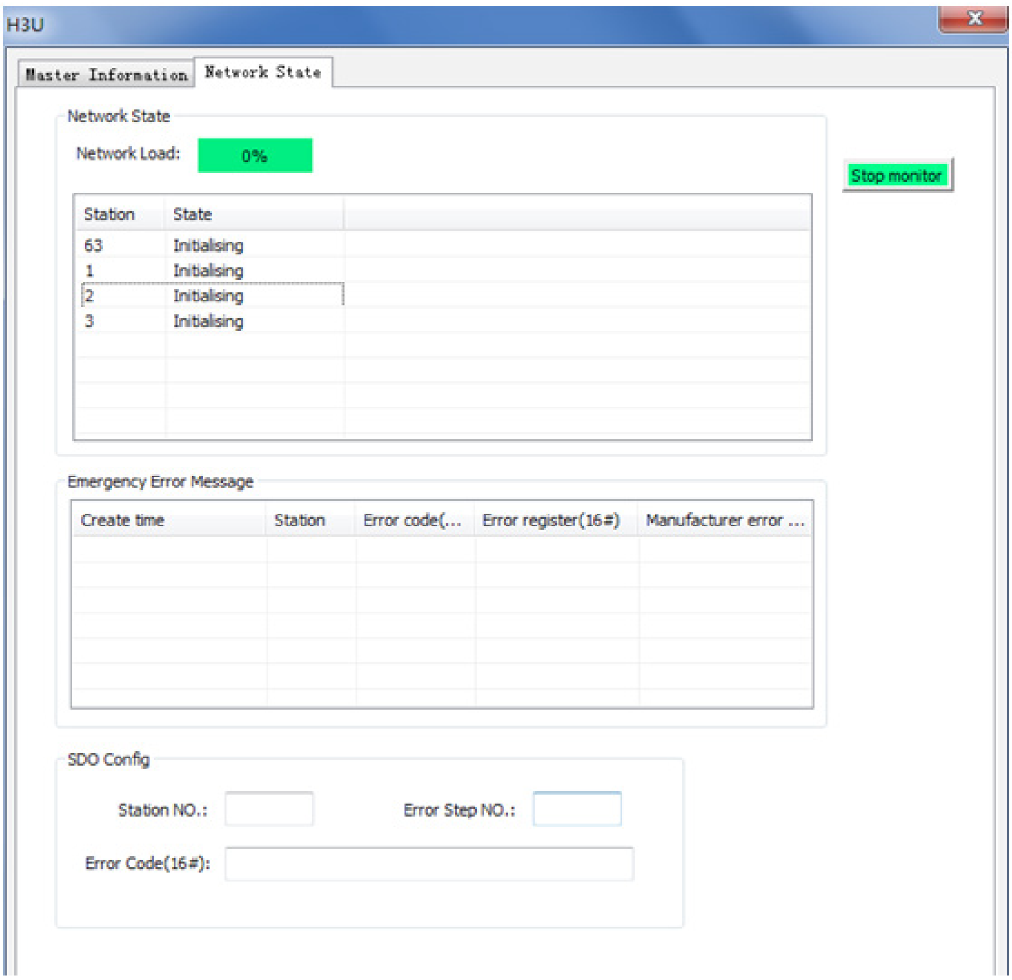
⑥ Automatic Allocation Map Register

Automatic Allocation: If this option is selected, the system will automatically assign the address of the register for master-slave data exchange. If the option is not selected, you need to configure the head address for data exchange (by configuring the head address of each PDO). The option is selected by default.

Slaves receive the map registers start address: Indicates the automatically assigned head address of data sent by the master station (Automatic Allocation must be selected).

Slaves send the map registers start address: Indicates the automatically assigned head address of data received by the master station (Automatic Allocation must be selected).

b) Network state



9

① Network State

Start Monitor: Information monitoring is enabled by clicking this option. Monitoring is disabled by clicking the option again.

Network Load: Monitors the network load in real time.

Network state table: Displays the station state. The table is applicable only to the master station. The state value is from the node state monitoring register.

② Emergency Error Message

The table lists emergency error messages on the network. It is applicable only to the master station. The master PLC only caches the latest error message. If background programs are not shut down, a maximum of five messages will be cached in the background.

SDO Config (see “9.7.4 CANopen Configuration” on page 637)

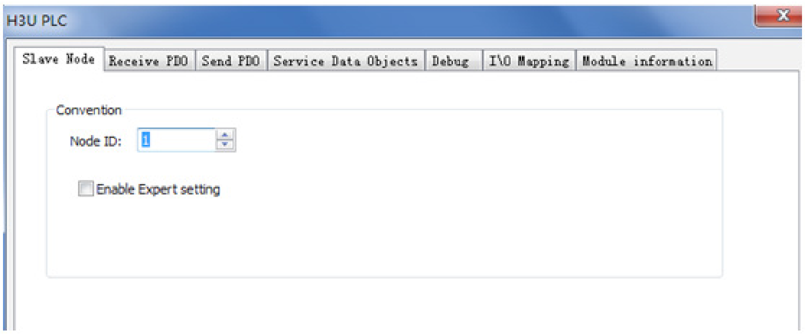
Station No.: Indicates the number of the station with SDO configuration errors.

Error Step No.: Indicates the SDO error number. To check numbers of slave stations with parameter errors, click the SDO tab.

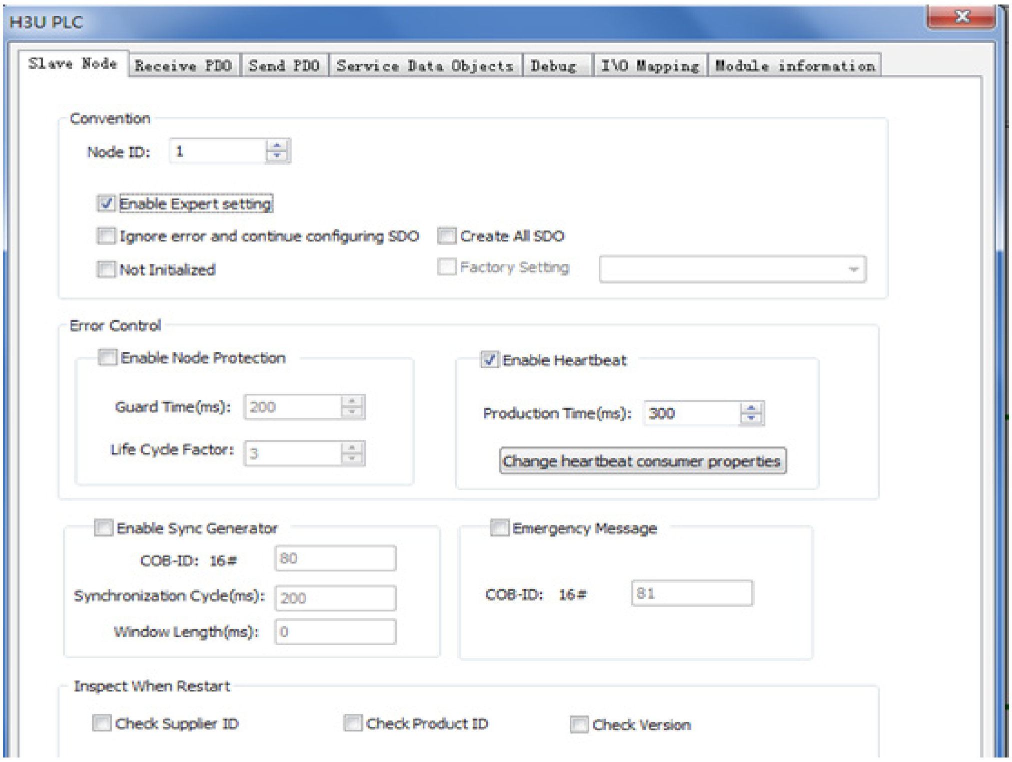
Error Code: Indicates the SDO error code (standard CANopen error code).

2) Configuring a slave station (H3U slave station)

Double-click the icon of a slave station. The following window is displayed.



Select Enable Expert setting. The following window is displayed. (By default, this option is not selected.)



a) Convention

Node ID: Indicates the ID of a node.

Enable Expert setting: When this option is selected, detailed configurations are displayed. By default, the option is not selected.

Ignore error and continue configuration SDO

① Valid: When a configuration error (other than a check type error) occurs, configuration continues.

② Invalid: When a configuration error occurs, configuration cannot continue, and the entire network is disconnected. By default, this option is not selected.

Create All SDO: If this option is selected, all writable ODs in the EDS will be added and initialized. By default, the option is not selected.

Not Initialized: If this option is selected, the slave station will not be initialized (the option can be selected only when the station applies the default configuration). By default, the option is not selected.

9

Factory Setting: If this option is selected, you can select options in the drop-down box. By default, the option is not selected.

b) Error Control

① Node protection attributes

Enable Node Protection: If this option is selected, node protection will be enabled. By default, the option is not selected.

Node protection timeout = Protection time x Life cycle factor

Node protection provides a network evaluation platform on which master station and slave station monitor each other with return frames. Either the heartbeat or node protection function can be selected.

Guard Time (ms): Indicates the node protection time, which is 200 ms by default.

Life Cycle Factor: Indicates the node protection factor, which is 3 by default.

② Heartbeat attributes

Enable Heartbeat: If this option is selected, heartbeats will be generated. By default, the option is selected. When the option is selected, the master station will monitor the heartbeat state by default.

Production Time (ms): Indicates the cycle for heartbeat sending.

Change heartbeat consumer properties: It is used to set heartbeats of other stations to be monitored by the configured station. This function is disabled by default. The function can be enabled only when the slave station supports heartbeat monitoring.

③ Synchronous (if supported)

Enable Sync Generator: If this option is selected, the configured station will send a sync frame repeatedly in the set synchronization cycle.

COB-ID: Indicates the ID for sync frame sending. The default value is 0x80. The parameter cannot be configured.

Synchronous Cycle (ms): Indicates the cycle for sync frame sending. The default value is 200, in the unit of milliseconds.

Window Length (ms): The value is 0 by default. The parameter cannot be configured.

NOTE

- On a network, only one sync frame can be sent.

C) Emergency message

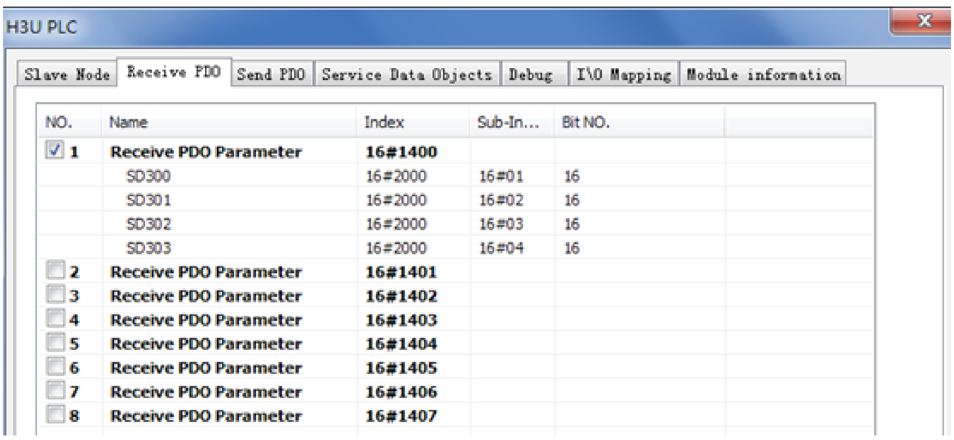
If this option is selected, you can set the COB-ID of an emergency message. By default, the option is not selected.

d) Inspect When Restart

If **Check Supplier ID**, **Check Product ID**, or **Check Version** is selected, corresponding data will be checked before configuration of the slave station. If the check fails, the network cannot be connected.

3) Receiving/sending PDO parameters

Click **Receive PDO** or **Send PDO**. The following page is displayed.



① Receive PDO Parameter: Indicates the data sent by the master station to a slave station.

② Send PDO Parameter: Indicates the data sent by a slave station to the master station.

- PDO enabling

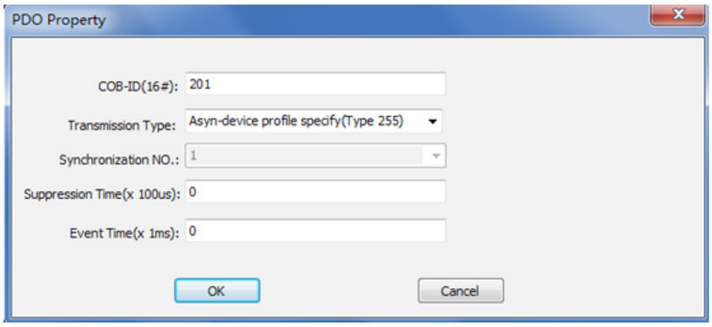
You can check the box in front of the number to enable a PDO. The PDO in the EDS file that takes effect by default should be checked first.

- Editing PDO mapping

You can click **Add PDO mapping**, **Edit**, or **Delete** to edit PDO mapping.

- Setting PDO properties

Double-click a PDO. The following page is displayed.



COB-ID: Indicates the ID for sending a PDO parameter. Based on the CANopen DS301 protocol, default COB-IDs are available for the first four PDO parameters. COB-IDs must be different from each other, ranging from 0x180 to 0x57F.

Transmission type:

Type	Condition for Data Sending	Condition for Valid Data
Loop-synchronization (type 0)	Data is changed, and a sync frame is received.	Data does not take effect immediately but takes effect after a sync frame is received.
Loop-synchronization (types 1 to 240)	Data is sent after the corresponding "number of synchronizations" frame is received.	Data does not take effect immediately but takes effect after a sync frame is received.
Asynchronization-only RTR (type 252)	Not supported	Not supported
Asynchronization-only RTR (type 253)	Not supported	Not supported
Asynchronization-specified by manufacturers (type 254)	Manufacturer-defined	Manufacturer-defined
Asynchronization-specified by the configuration file (type 255)	Data is changed or the event time is correct, and the change cycle is shorter than the suppression time.	Immediately

NOTE

- When enabling a type of synchronization, you need to enable synchronous production on a station, usually the master station.

The number of synchronizations takes effect after "loop-synchronization (types 1 to 240)" is selected.

The suppression time can be set after "asynchronization-specified by the configuration file (type 255)" is selected. If the value is 0, the function is disabled. If the value is not 0, the suppression time is the minimum interval for frame sending.

The event time can be set after "asynchronization-specified by the configuration file (type 255)" is selected. If the value is 0, the function is disabled. If the value is not 0, the event time is the cycle for data sending. (Data sending is limited by the suppression time.)

The following figure shows the example of loop-synchronization (type 2).

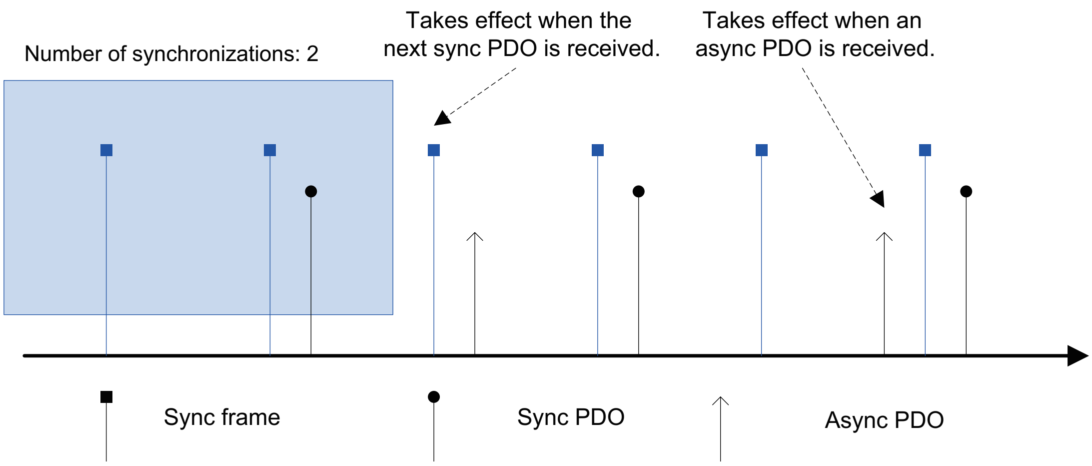
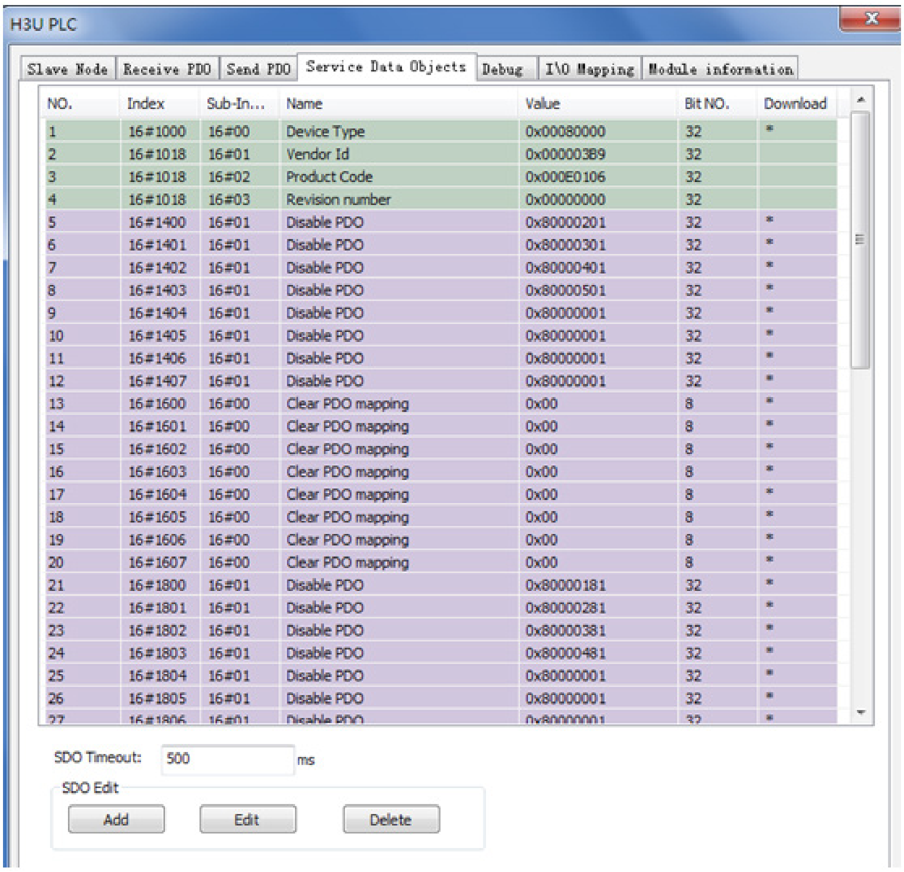


Diagram of PDO sending

③ SDO

Click the **SDO** tab. The following page is displayed.



NO.	Index	Sub-In...	Name	Value	Bit NO.	Download
1	16#1000	16#00	Device Type	0x00080000	32	*
2	16#1018	16#01	Vendor Id	0x000003B9	32	
3	16#1018	16#02	Product Code	0x000E0106	32	
4	16#1018	16#03	Revision number	0x00000000	32	
5	16#1400	16#01	Disable PDO	0x80000201	32	*
6	16#1401	16#01	Disable PDO	0x80000301	32	*
7	16#1402	16#01	Disable PDO	0x80000401	32	*
8	16#1403	16#01	Disable PDO	0x80000501	32	*
9	16#1404	16#01	Disable PDO	0x80000001	32	*
10	16#1405	16#01	Disable PDO	0x80000001	32	*
11	16#1406	16#01	Disable PDO	0x80000001	32	*
12	16#1407	16#01	Disable PDO	0x80000001	32	*
13	16#1600	16#00	Clear PDO mapping	0x00	8	*
14	16#1601	16#00	Clear PDO mapping	0x00	8	*
15	16#1602	16#00	Clear PDO mapping	0x00	8	*
16	16#1603	16#00	Clear PDO mapping	0x00	8	*
17	16#1604	16#00	Clear PDO mapping	0x00	8	*
18	16#1605	16#00	Clear PDO mapping	0x00	8	*
19	16#1606	16#00	Clear PDO mapping	0x00	8	*
20	16#1607	16#00	Clear PDO mapping	0x00	8	*
21	16#1800	16#01	Disable PDO	0x80000181	32	*
22	16#1801	16#01	Disable PDO	0x80000281	32	*
23	16#1802	16#01	Disable PDO	0x80000381	32	*
24	16#1803	16#01	Disable PDO	0x80000481	32	*
25	16#1804	16#01	Disable PDO	0x80000001	32	*
26	16#1805	16#01	Disable PDO	0x80000001	32	*
27	16#1806	16#01	Disable PDO	0x80000001	32	*

The table lists SDO configurations automatically generated based on user settings.

- SDO editing

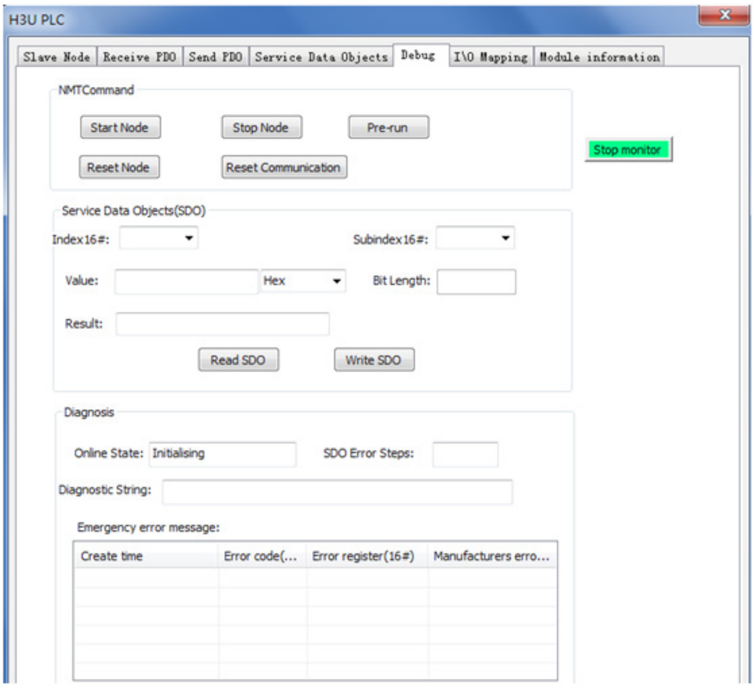
Add: Adds configurations. It is used to assign initial values to ODs of a slave station.

Edit: Edits configurations.

Delete: Deletes configurations.

④ Online debugging

Click the **Debug** tab. The following page is displayed.



NOTE

- If **The program is running prohibited SDO, NMT access** is selected, the function is disabled.

- NMT Command

Start Node: Sends a command to the slave station to start a node.

Stop Node: Sends a command to the slave station to stop a node.

Pre-run: Sends a command to the node to pre-run.

Reset Node: Sends a command to the node to reset.

Reset Communication: Sends a command to the node to reset communication.

- Sercice Data Objects

You can only select ODs in the EDS as indexes or subindexes.

Value indicates sent or returned data.

Bit Length is automatically generated based on an OD in the EDS. It must not be modified.

Result indicates abnormality information.

Read SDO and Write SDO: Reads and writes ODs.

- Diagnosis (See “9.7.6 List of Fault Codes” on page 648.)

Online State: Indicates the state of the slave station (fed back based on heartbeat or node protection).

9

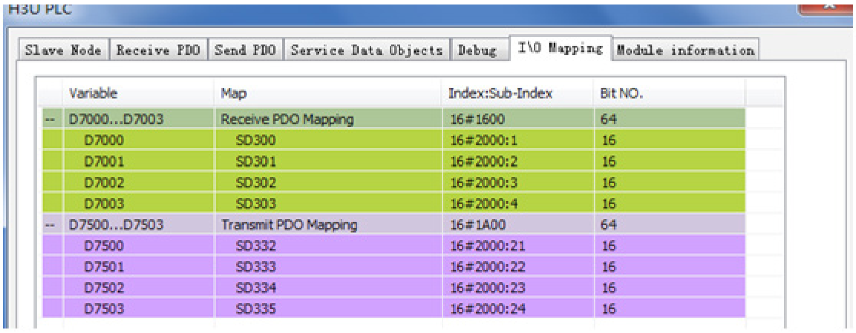
SDO Error Steps: Indicates the SDO error number. This number corresponds to the **Service Data Objects** tab.

Diagnosis String: Indicates the error message (SDO error).

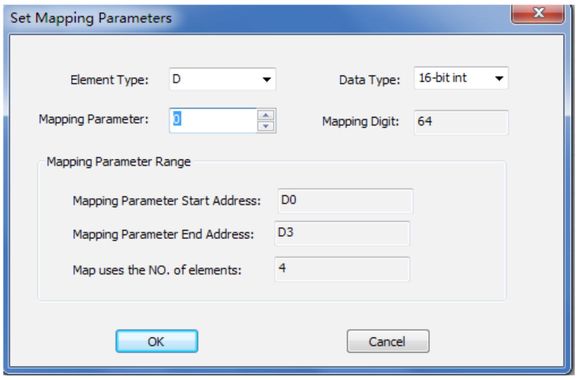
Emergency error message: Indicates an emergency error frame (the system monitors real-time errors and caches five error messages in the background; the PLC only caches the latest error message) (emergency error).

⑤ I/O mapping

Click the **I/O Mapping** tab. The following page is displayed.



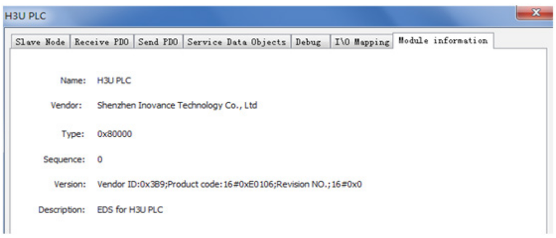
This tab is used to set the communication relationship between master and slave PDOs. If **Automatic Allocation** is not selected, when you double-click an item, the following page is displayed.



You can configure the head register address for the master station corresponding to a slave PDO.

⑥ Module information

Click the **Module information** tab. The following page is displayed.



Device information can be obtained from the EDS file.

9.7.5 Troubleshooting for CANopen Communication

1) Checking whether CANopen is supported

Device	Check
PLC	Check the D8280 value. If D8280 = 100, CANopen is supported; otherwise, CANopen is not supported.
AC drive/servo	Check the software version. For details, see the user manual.

2) Checking the build-out resistor

Power off all devices. Use a multimeter to measure the resistance between CANH and CANL. The resistance should be about 60 Ω. If the resistance is too small, there are build-out resistors incorrectly connected at other locations. In this case, disconnect these build-out resistors. If only one resistor is available, the resistance is about 120 Ω, and the network connection is bad. If no resistor is available, communication fails. Provide build-out resistors between the stations at both ends of the network.

3) Checking the baud rate

Check whether the baud rate is normal. Power off and then on the device or switch it from STOP to RUN so that the baud rate can take effect.

For the relationship between distance and baud rate, see "3) Relationship between distance and baud rate" on page 620.

4) Checking wiring

The CAN communication port and extension module of the PLC are powered by an external 24 V power supply. The AC drive and servo are self-powered. Interconnect CGND pins of all CAN devices to ensure that all devices share one power supply CGND port.

Check whether the communication cable, shielded cable, and power supply are short-circuited.

5) Others

In case of strong interference, reduce the baud rate.

9.7.6 List of Fault Codes

1) SDO error codes

Abort Code	Function
0503 0000	Trigger bits not alternated
0504 0000	SDO protocol timeout
0504 0001	Invalid or unknown client/server command word
0504 0002	Invalid block size (for the block transfer mode only)
0504 0003	Invalid serial number (for the block transfer mode only)
0503 0004	CRC error (for the block transfer mode only)
0503 0005	Memory overflow
0601 0000	Inaccessible object
0601 0001	Attempt to read a write-only object
0601 0002	Attempt to write a read-only object
0602 0000	Object unavailable in the OD
0604 0041	Object unable to be mapped to the PDO

Abort Code	Function
0503 0000	Trigger bits not alternated
0504 0000	SDO protocol timeout
0504 0001	Invalid or unknown client/server command word
0504 0002	Invalid block size (for the block transfer mode only)
0504 0003	Invalid serial number (for the block transfer mode only)
0503 0004	CRC error (for the block transfer mode only)
0503 0005	Memory overflow
0601 0000	Inaccessible object
0601 0001	Attempt to read a write-only object
0601 0002	Attempt to write a read-only object
0602 0000	Object unavailable in the OD
0604 0041	Object unable to be mapped to the PDO
0604 0042	Number and length of objects to be mapped exceeding those of PDOs
0604 0043	General parameter incompatibility
0604 0047	General internal incompatibility
0606 0000	Failure to access objects because of hardware errors
0606 0010	Incorrect data type: incorrect service parameter length
0606 0012	Incorrect data type: service parameter too long
0606 0013	Incorrect data type: service parameter too short
0609 0011	Subindex unavailable
0609 0030	Beyond the value range (for write access)
0609 0031	Written parameter value too large
0609 0032	Written parameter value too small
0609 0036	Maximum less than minimum
0800 0000	General error
0800 0020	Data unable to be transmitted or saved to the application
0800 0021	Data unable to be transmitted or saved to the application because of local control
0800 0022	Data unable to be transmitted or saved to the application because of the device state
0800 0023	OD error or OD unavailable (For example, an OD is generated through a file, but an error occurs because the file is corrupted.)

2) Main table 1 of emergency error codes (hexadecimal)

Emergency error code	Description
00xx	No error
10xx	General error
20xx	Current
21xx	Current at input end
22xx	Internal current
23xx	Current at output end
30xx	Voltage
31xx	Power supply voltage
32xx	Internal voltage
33xx	Output Voltage
40xx	Temperature
41xx	Operation temperature

9

Emergency error code	Description
42xx	Device temperature
50xx	Hardware
60xx	Software
61xx	Internal software
62xx	User software
63xx	Setting
70xx	Extra module
80xx	Monitoring
81xx	Communication
82xx	Protocol error
90**	External error
F0**	Extra function
FF**	Special device

3) Table 2 of emergency error codes (hexadecimal)

Emergency error code	Description
0000	Incorrect reset or no error
1000	General error
2000	Current error
2100	Input current
2200	Internal current
2300	Output current
3000	Voltage error
3100	Power supply voltage
3200	Internal voltage
3300	Output voltage
4000	Temperature error
4100	Operation temperature
4200	Device temperature
5000	Hardware error
6000	Software error
6100	Internal software
6200	User software
6300	Setting
7000	Extra module error
8000	Monitoring error
8100	General communication error
8110	CAN communication overload
8120	Incorrect CAN passive method
8130	Node protection or heartbeat error
8140	Bus disconnection

Emergency error code	Description
8150	CAN-ID impulse
8200	Protocol error
8210	Incorrect PDO length
8220	Excessive PDO length
8240	Unidentifiable sync data length
8250	RPDO timeout
9000	External error
F000	Extra function error
FF00	Special device error

9.8 Ethernet Communication

The main H3U module has Ethernet communication interfaces, with support for adaptive 10 Mbps/100 Mbps rate and Modbus TCP.

An H3U standard model supports 16 connections (connections sharing the same IP address and port number are regarded as one connection; an H3U-PM motion control model supports eight connections). A master or slave H3U PLC can exchange data with a maximum of 16 stations (an H3U-PM motion control model can exchange data with a maximum of eight stations). A station can serve as a master station and a slave station at a time.

Sent and received Ethernet frames are processed in each user program scan cycle, so the read-write speed is affected by the user program scan cycle.

9.8.1 Hardware Interface and IP Settings

H3U-3232MT/R and H3U-0808PMRTA models differ from the H3U-1616MT/R-XP model in Ethernet communication interface design, as shown in the following figure.

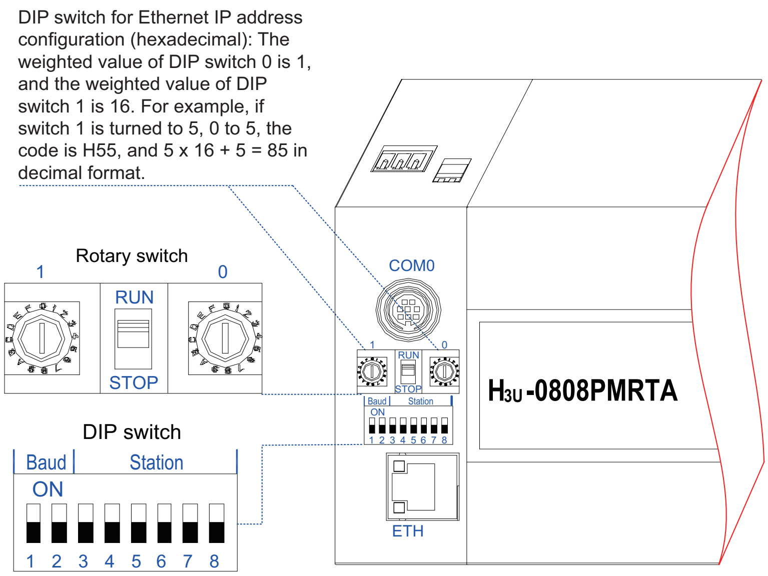


Diagram of H3U-3232MT/R and H3U-0808PMRTA Ethernet communication interfaces

9

Silk Screen	Terminal	Function
RJ45	RJ45 interface	Ethernet communication interface
0	DIP switch board	Ones place of the last segment of the IP address (0 to F)
1	DIP switch board	Tens place of the last segment of the IP address (range of tens place + ones place: 0 to FF)
ETH	Communication indicator (green)	Blinking: Data being transmitted Off: No data transmitted

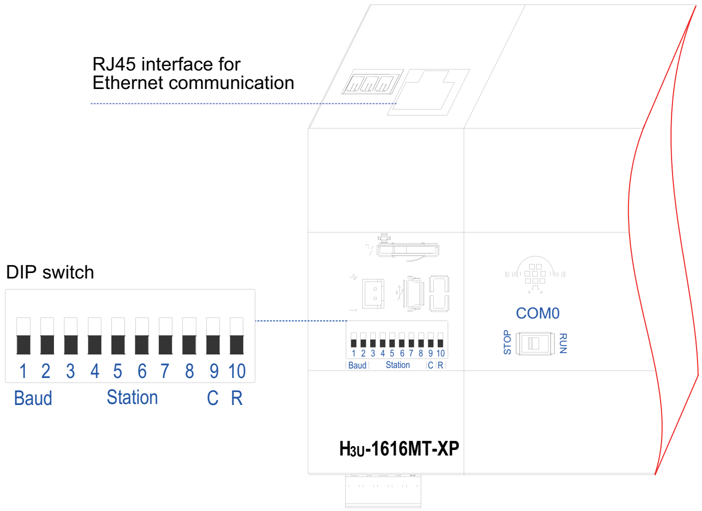
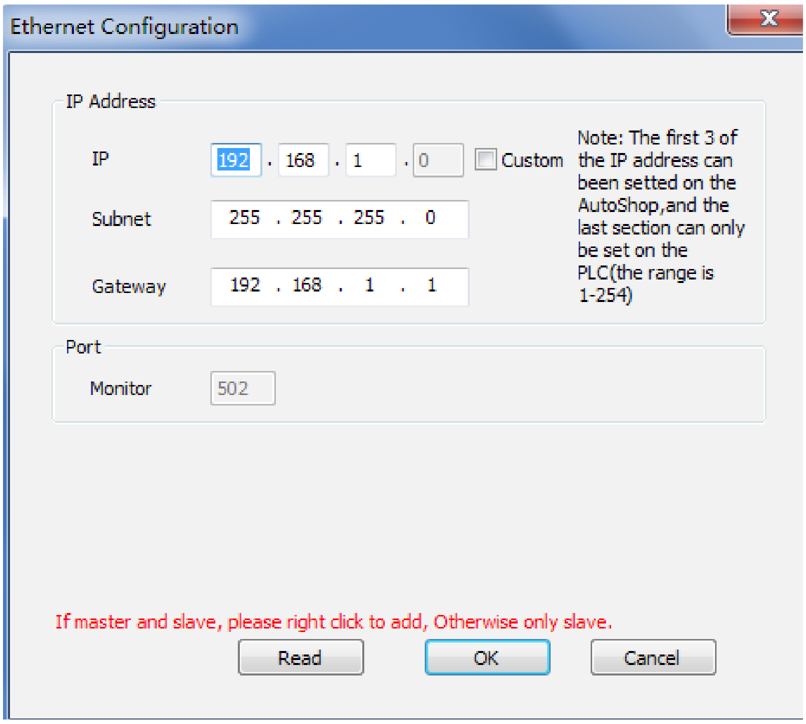


Diagram of H3U-1616MT/R-XP Ethernet communication interface

- The fourth segment of the Ethernet IP address of the H3U-1616MT/R-XP model overlaps the CAN address. For details, see "2) DIP switch" on page 619.

9.8.2 Ethernet Configuration

In the **Project Manager** window, double-click **Ethernet**.



If the device serves as a slave station, you only need to configure the IP address.

If the device serves as the master station, you need to configure the master station.

- IP address: It is a device identifier on a network. The IP address of each device must be unique; otherwise, the device cannot be connected to the network. The first three segments of an IP address are configured using AutoShop. The last segment can be customized or configured through the DIP switch, and the value ranges from 1 to 254.

• H3U-3232 model: When the rotary switch is set to 255, the IP address is forced to be 192.168.1.1.

• H3U-1616 model: When bits 1 to 8 of the DIP switch are set to ON, the IP address is forced to be 192.168.1.1.

• If the IP address is invalid, it will be set to 192.168.1.1.

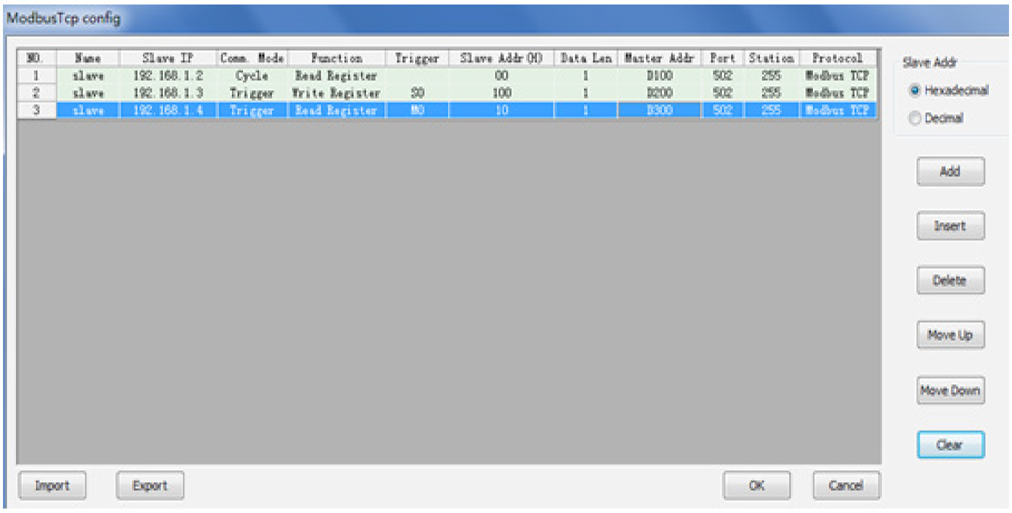
- Subnet mask: Multiple physical networks are identified with the same address. A mask is used to divide an IP address into subnet address and host address. To obtain a subnet address, you can reserve the digits of the IP address corresponding to "1" digits of the mask, and then replace other digits with "0". If not specified, the subnet mask is 255.255.255.0.

- Gateway address: It can be used to route messages to devices outside the network. If no gateway is available, the value is 0.0.0.0.

- Port: Port TCP 502 listening is reserved for Modbus TCP. This port must be listened on, which cannot be configured.

1) H3U master station configuration

When an H3U PLC serves as the master station, in addition to the IP address, you need to configure other parameters such as data length. Click **Ethernet Master Station Config**. The following window is displayed.



- Device name: It is auxiliary information, which can be customized.
- Slave station IP address: It should be configured based on the device address. You can double-click the space so that the existing IP address will be displayed, or you can create an IP address. There can be multiple configurations for one IP address.
- Communication mode: In cyclic mode, the master station accesses a slave station cyclically. In triggered mode, trigger elements are used: when the element is ON, the master station accesses the slave station; after the operation is finished, the element is automatically set to OFF.
- Function: Functions include reading coils, writing coils, reading registers, and writing registers.
- Trigger condition: It can be a non-special M or S element.
- Slave register address: It is the address of the coil or register to be accessed (hexadecimal).

- Data length: It is the length of the data to be accessed. If the master station will access M0 to M10 (11 elements) of a slave station, the value is 11.
- Master buffer address: It is the head address of the master buffer. As shown in the preceding figure (No. 1), the master station reads data from the slave station and stores the read data in D100. In this case, D100 can be accessed in the user program. As shown in the preceding figure (No. 2), the master station writes values of 100 elements starting with D200 in 100 registers starting with 0.
- Port number: It is 502 by default. Port 502 is specified by Modbus TCP, and you do not need to modify it.
- Station number: It is a serial port number assigned to an Ethernet serial device. It is 255 by default, and you do not need to modify it.

Note: For each configuration item, there is an upper limit.

Maximum number of coils to be read	1968
Maximum number of coil groups to be written	1936
Maximum number of registers to be read	123
Maximum number of registers to be written	121

Modbus TCP command codes supported by an H3U model

Command Code	Function
0x01	Reads coils.
0x02	Reads coils.
0x03	Reads registers.
0x04	Reads registers.
0x05	Writes one coil.
0x06	Writes one register.
0x0F	Writes multiple coils.
0x10	Writes multiple registers.

After the Ethernet master station configuration is downloaded, the system automatically creates and manages connections.

9.8.3 Soft Element Access Addresses When the H3U PLC Serves as a Slave Station

H3U soft elements can be accessed through Modbus TCP devices. The following table lists addresses of soft elements.

Coil Addressing		
Variable	Head address	Number of coils
M0-M7679	0 (0)	7680
M8000-M8511	0x1F40 (8000)	512
SM0-SM1023	0x2400 (9216)	1024
S0-S4095	0xE000 (57344)	4096
T0-T511	0xF000 (61440)	512
C0-C255	0xF400 (62464)	512
X0-X377	0xF800 (63488)	256
Y0-Y377	0xFC00 (64512)	256

Register addressing		
Variable	Head address	Number of registers
D0-D8511	0 (0)	8512
SD0-SD1023	0x2400 (9216)	1024
R0-R32767	0x3000 (12288)	32768
T0-T511	0xF000 (61440)	512
C0-C199	0xF400 (62464)	200
C200-C255	0xF700 (63232)	56 (32-bit)

9.8.4 Special Ethernet Soft Elements

Element	Function	Element	Function
SM364	Flag of busy Ethernet, which cannot be automatically reset	SD364	Listening port of an Ethernet slave station
SM365	Flag of offline state, which can be automatically reset based on the online state of the slave station 0: No slave station is offline. 1: One or more slave stations are offline.	SD365	Offline station number (the fourth segment of the IP address), which can be displayed when SM365 is set.
SM366	Flag of Ethernet enabling/disabling 0: The Ethernet is enabled. 1: The Ethernet is disabled.	SD366	Error number (configuration table number)
SM367	Reserved	SD367	Modbus TCP error code (upper eight bits indicate the command, and lower eight bits indicate the error code)
SM368	Reserved	SD368	Reserved
SM369	Reserved	SD369	Timeout (in the unit of 10 ms), 20 by default
SM370	Reserved	SD370	IP address 1
SM371	Reserved	SD371	IP address 2
SM372	Reserved	SD372	IP address 3
SM373	Reserved	SD373	IP address 4
SM374	Reserved	SD374	MAC address 1
SM375	Reserved	SD375	MAC address 2
SM376	Reserved	SD376	MAC address 3
SM377	Reserved	SD377	MAC address 4
SM378	Reserved	SD378	MAC address 5
SM379	Reserved	SD379	MAC address 6

IP and MAC addresses are stored in read-only SD elements.

9.8.5 Detecting H3U PLC Connection Faults

- Whether the network connection is normal

Unstable network connection may be caused by interference or poor contact. Use a shielded network cable to make a new RJ45 connector. (You can use a **ping** command of the computer to preliminarily check the network state.)

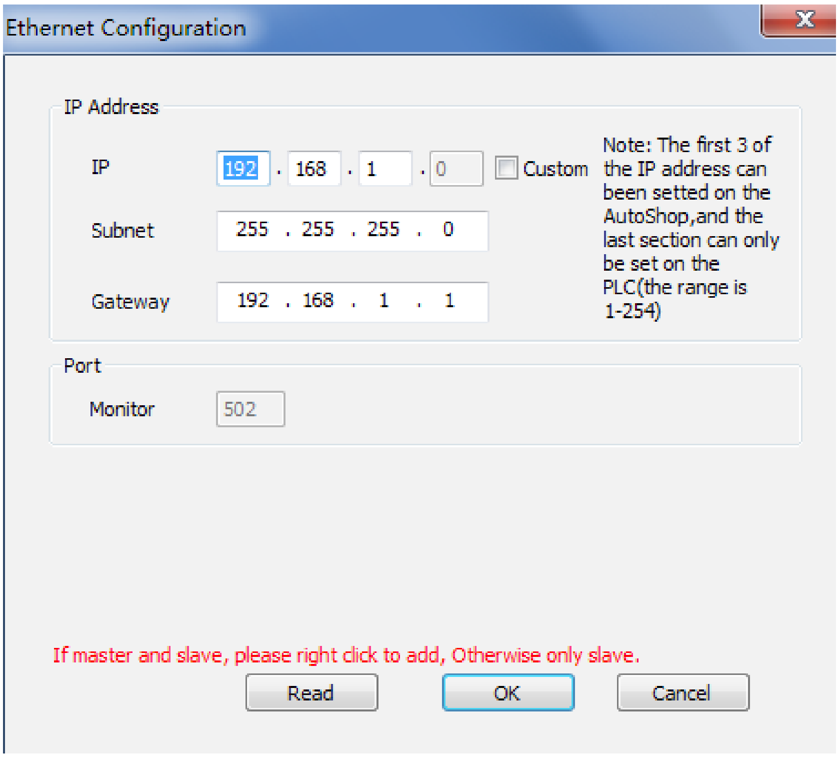
- To check whether the IP address is correctly configured, display SD370 to SD373.

9

- If a gateway is used, check if the gateway address is correctly configured.
- Check whether slave register addresses in the configuration table are correct (hexadecimal).
- If two devices with IP addresses of different network segments (the first three segments) need to communicate with each other, a router is needed to connect the devices.
- If communication timeout frequently occurs on slave stations (for example, the PC as a slave station returns frames slowly) after network problems have been resolved, increase the value of SD369 timeout.

9.8.6 Downloading over the Ethernet and Monitoring

1) Configure the IP address of a slave station and download the configuration. In the Project Manager window, click Ethernet to configure the Ethernet. Valid IP addresses can be displayed through SD370 to SD373.

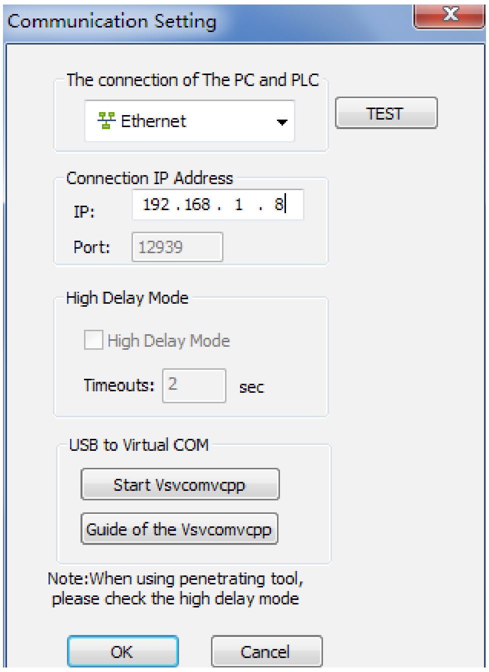


9

NOTE

- The default IP address is 192.168.1.*. The last digit depends on the rotary or DIP switch.
- The IP address of the PC on the LAN must be of the same network segment as the PLC (the first three segments); otherwise, a router is needed.

2) Choose **Tool** > **Communication Configuration**. The following window is displayed. Select an Ethernet. Select an updated and valid IP address of the PLC. Click OK.



NOTE

- Firmware upgrade is not supported.

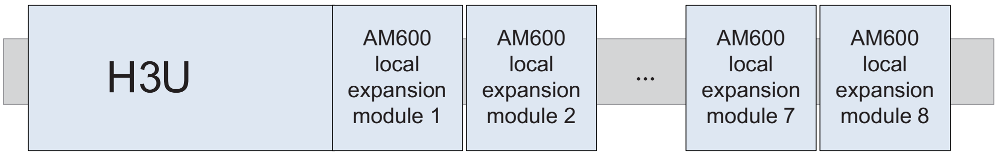
10 Extension Modules

Chapter 10 Extension Modules

10.1 Overview

The H3U enables access to local or remote extension modules through module configuration.

(1) Example of H3U configuration for local extension modules

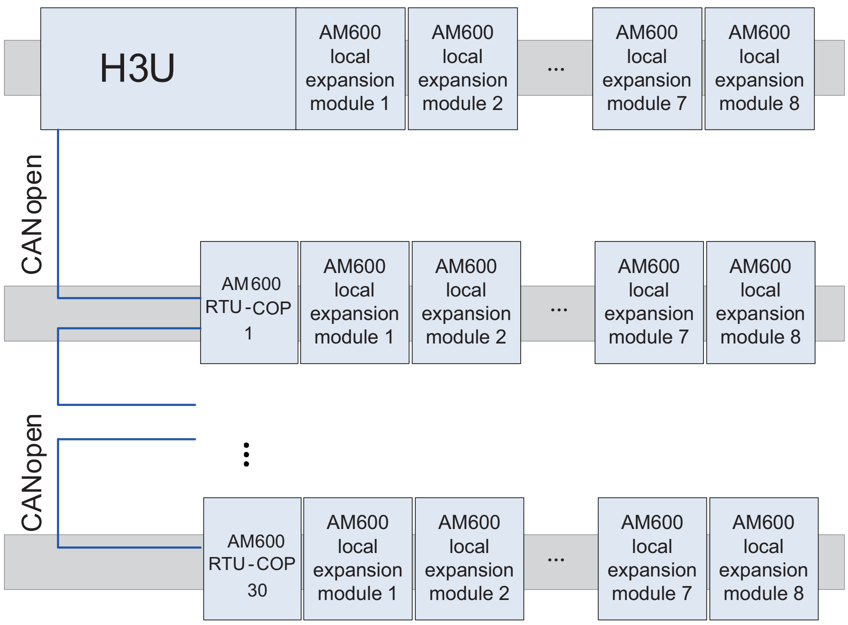


The H3U can connect up to eight local extension modules, but does not support the H2U series local extension modules and extension cards.

(2) Models of H3U-supported local extension modules

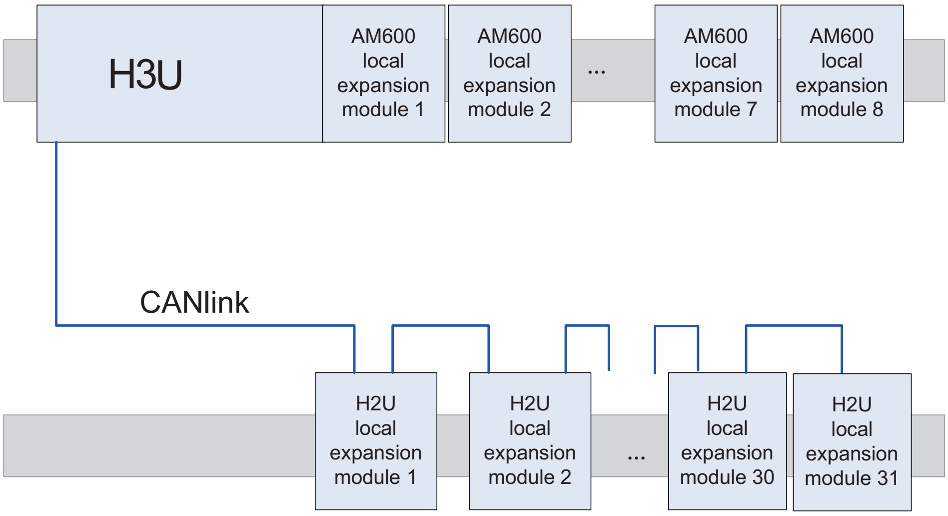
Product Name	Description
AM600-0016ETP	16-channel digital transistor output module – PNP
AM600-0016ETN	16-channel digital transistor output module – NPN
AM600-0016ER	16-channel digital relay output module – Relay
AM600-1600END	16-channel digital input module
AM600-4AD	4-channel analog input module
AM600-4DA	4-channel analog output module
AM600-4PT	4-channel input heating resistor temperature monitoring module
AM600-4TC	4-channel input thermocouple temperature monitoring module

(3) Example of H3U configuration for CANopen bus-based access to extension modules



The H3U can connect up to 30 CANopen AM600RTU-COP devices, and each AM600RTU-COP can connect up to eight local extension modules.

(4) Example of H3U configuration for CANlink bus-based access to extension modules



The H3U can connect up to 31 H2U series CANlink remote extension modules.

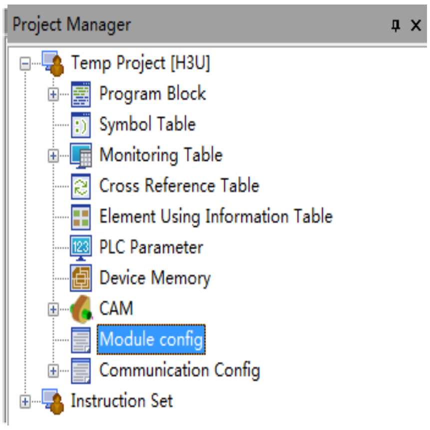
The following table lists the models of the H2U series CANlink remote extension modules supported by the H3U.

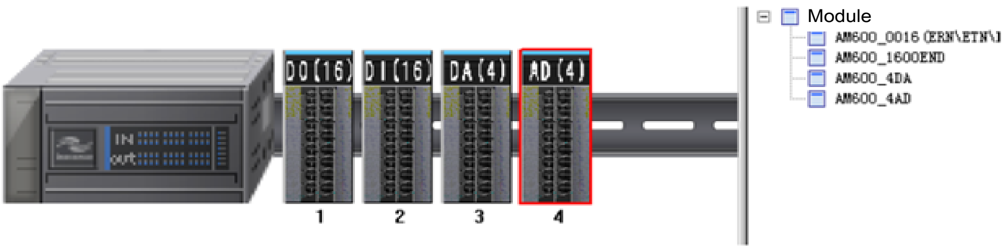
Product Name	Description
H2U-0016ERDR	16-channel relay output remote module
H2U-0016ETDR	16-channel transistor output remote module
H2U-1600ENDR	16-channel input remote module
H2U-0808ERDR	8-channel input and 8-channel relay output remote module
H2U-0808ETDR	8-channel input and 8-channel transistor output remote module
H2U-2ADR	2-channel analog input remote module
H2U-2DAR	2-channel analog output remote module
H2U-4ADR	4-channel analog input remote module
H2U-4DAR	4-channel analog output remote module
H2U-4PTR	4-channel heating resistor input remote module
H2U-4TCR	4-channel thermocouple input remote module
H2U-4AMR	2-channel analog input and 2-channel analog output remote module
H2U-6AMR	4-channel analog current input and 2-channel analog output remote module
H2U-6CMR	4-channel analog voltage input and 2-channel analog output remote module

10.2 Local Extension Modules

10.2.1 Configuration

Local extension modules operate based on hardware configuration. Double-click **Module config** in AutoShop.





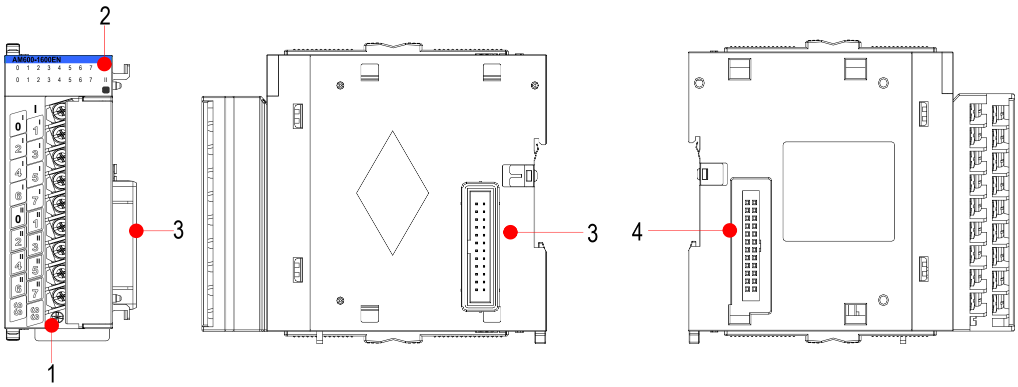
After a module is created, the configuration interface is displayed. Double-click the module to automatically add it to the extension rack, or use the left mouse button to drag it onto the extension rack.

10.2.2 Digital Input Module (AM600-1600END)

(1) Specifications

Item	Specifications
Input channel	16
Input connecting mode	18-point wiring terminal
Input type	Digital input
Input mode	SINK/SOURCE
Input voltage class	24 VDC (max.: 30 V)
Input current (typical)	5.3 mA
ON voltage	> 15 VDC
OFF voltage	< 5 VDC
Port filter time	10 ms
Input resistance	4.3 kΩ
Input signal form	DC voltage input, supporting SINK/SOURCE input
Isolation method	Opto-couplers isolation
Input action display	Input indicator ON when the input is in the driving state

(2) Terminal definition and external wiring



SN	Interface Name	Function
1	User input terminal	8-channel inputs x 2
2	Signal indicators	Corresponding to various input signals ON: input active OFF: input inactive
3	Local extension module back-end interface	Connect back-end module
4	Local extension module front-end interface	Connect front-end module

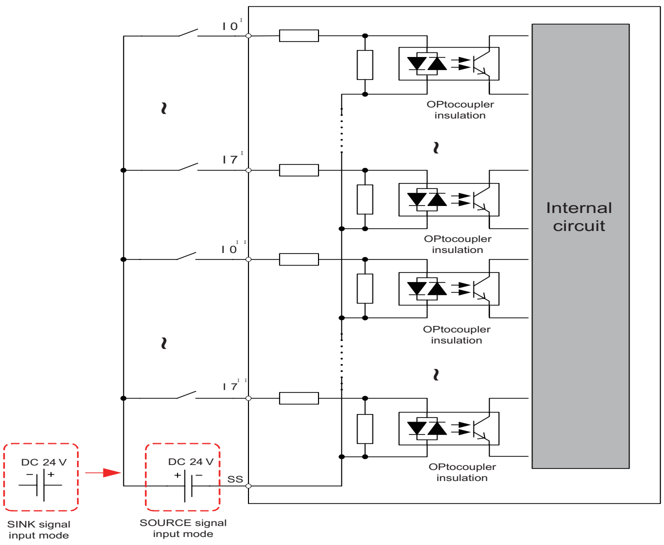
- Terminal definition

SN	Network Name	Type	Function	Remarks
The first Group of Input Interfaces				
1	I0	Input	The 1st group of user input 0	SINK/SOURCE input
2	I1	Input	The 1st group of user input 1	SINK/SOURCE input
3	I2	Input	The 1st group of user input 2	SINK/SOURCE input
4	I3	Input	The 1st group of user input 3	SINK/SOURCE input
5	I4	Input	The 1st group of user input 4	SINK/SOURCE input
6	I5	Input	The 1st group of user input 5	SINK/SOURCE input
7	I6	Input	The 1st group of user input 6	SINK/SOURCE input
8	I7	Input	The 1st group of user input 7	SINK/SOURCE input

SN	Network Name	Type	Function	Remarks
The 2nd Group of Input Interfaces				
9	I0	Input	The 2nd group of user input 0	SINK/SOURCE input
10	I1	Input	The 2nd group of user input 1	SINK/SOURCE input
11	I2	Input	The 2nd group of user input 2	SINK/SOURCE input
12	I3	Input	The 2nd group of user input 3	SINK/SOURCE input
13	I4	Input	The 2nd group of user input 4	SINK/SOURCE input
14	I5	Input	The 2nd group of user input 5	SINK/SOURCE input
15	I6	Input	The 2nd group of user input 6	SINK/SOURCE input
16	I7	Input	The 2nd group of user input 7	SINK/SOURCE input
Common				
17	SS	Power	Common	-
18	SS	Power	Common	-

- External wiring

Digital input module wiring diagram



(3) Module usage

When a local digital input extension module is connected to the main module, the SN of the X port on the extension module follows that of the X port on the main module. For example, the main module H3U connects the AM600-1600END. The SN of the last X port of the main module is X37. The SNs of the 16 X ports of the extension module are X40 to X47 and X50 to X57. Use the same method to number X ports of the following digital input extension modules.

10.2.3 Digital Output Module (AM600-0016ETN, AM600-0016ETP, and AM600- 0016ER)

(1) Specifications

- Digital transistor output modules

Item		AM600-0016ETP	AM600-0016ETN
Output channel		16	16
Output connecting mode		18-point wiring terminal	18-point wiring terminal
Output type		Transistor, high-side output	Transistor, low-side output
Output mode		SOURCE	SINK
Supply voltage		24 VDC (–15% to +20%)	
Output voltage class		12 to 24 V (–5% to +20%)	
OFF max. leakage current		0.5 mA below	
ON response time		0.5 ms below (hardware response time)	
OFF response time		0.5 ms below (hardware response time)	
Max. load	Resistive load	0.5 A/point, 2 A/common	
	Inductive load	12 W/24 VDC (total)	
	Lamp load	2W/24 VDC (total)	
Isolation method		Opto-couplers isolation	
Output action display		Output indicator ON when opto-coupler driving is applied	
Short circuit-proof output		Yes (The current is limited to 1 to 1.7 A when short circuit protection is applied.)	None

- Digital relay output module

Item		AM600-0016ER
Output channel		16
Output connecting mode		16+2 (COM) point wiring terminals
Power supply of the module[1]		24 VDC (–15% to +20%)
Output type		Relay output
Output mode		-
Voltage of the output control circuit		110 to 220 VAC
Rated current of relay		240 VAC/24 VDC, 2 A
OFF max. leakage current		-
ON response time		20 ms below (hardware response time)
OFF response time		20 ms below (hardware response time)
Max. load	Resistive load	Single-point 1 A/point
	Lamp load	Single-point 30 W
	Inductive load	220 VAC, 2 A/1 point
	Capacitive load	Not recommended

Item	AM600-0016ER
Isolation method	Mechanical isolation
Output action display	Output indicator ON when the relay is excited

(2) Terminal definition and external wiring

- Terminal definition of the transistor output module (AM600-0016ETP)

SN	Network Name	Type	Function	Remarks
Interface for the 1st group of outputs				
1	Q0	Output	User output 0 in the 1st group	SOURCE output, active high
2	Q1	Output	User output 1 in the 1st group	SOURCE output, active high
3	Q2	Output	User output 2 in the 1st group	SOURCE output, active high
4	Q3	Output	User output 3 in the 1st group	SOURCE output, active high
5	Q4	Output	User output 4 in the 1st group	SOURCE output, active high
6	Q5	Output	User output 5 in the 1st group	SOURCE output, active high
7	Q6	Output	User output 6 in the 1st group	SOURCE output, active high
8	Q7	Output	User output 7 in the 1st group	SOURCE output, active high
Interface for the 2nd group of outputs				
9	Q0	Output	User output 0 in the 2nd group	SOURCE output, active high
10	Q1	Output	User output 1 in the 2nd group	SOURCE output, active high
11	Q2	Output	User output 2 in the 2nd group	SOURCE output, active high
12	Q3	Output	User output 3 in the 2nd group	SOURCE output, active high
13	Q4	Output	User output 4 in the 2nd group	SOURCE output, active high
14	Q5	Output	User output 5 in the 2nd group	SOURCE output, active high
15	Q6	Output	User output 6 in the 2nd group	SOURCE output, active high
16	Q7	Output	User output 7 in the 2nd group	SOURCE output, active high
Power connector				
17	24 V	Power	24 V power supply	24 VDC power input
18	COM	Power	Power ground	24 VDC power common

- Terminal definition of the transistor output module (AM600-0016ETN)

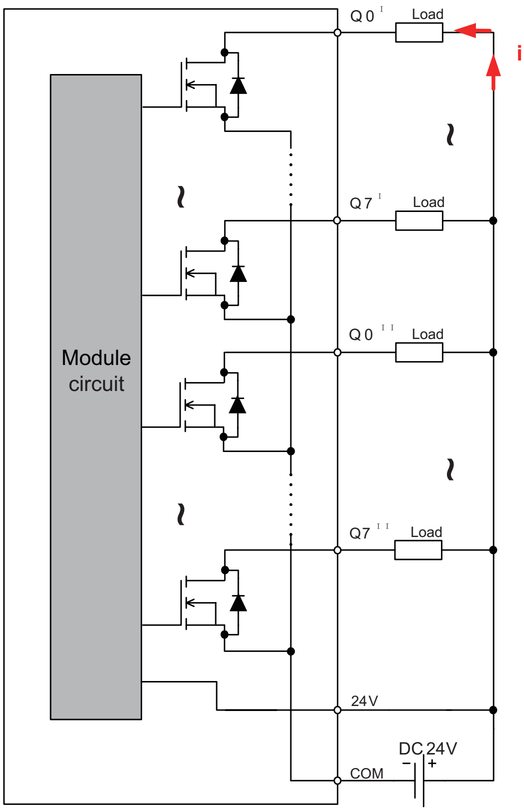
SN	Network Name	Type	Function	Remarks
Interface for the 1st group of outputs				
1	Q0	Output	User output 0 in the 1st group	SINK, active low
2	Q1	Output	User output 1 in the 1st group	SINK, active low
3	Q2	Output	User output 2 in the 1st group	SINK, active low
4	Q3	Output	User output 3 in the 1st group	SINK, active low
5	Q4	Output	User output 4 in the 1st group	SINK, active low
6	Q5	Output	User output 5 in the 1st group	SINK, active low
7	Q6	Output	User output 6 in the 1st group	SINK, active low
8	Q7	Output	User output 7 in the 1st group	SINK, active low
Interface for the 2nd group of outputs				
9	Q0	Output	User output 0 in the 2nd group	SINK, active low
10	Q1	Output	User output 1 in the 2nd group	SINK, active low
11	Q2	Output	User output 2 in the 2nd group	SINK, active low
12	Q3	Output	User output 3 in the 2nd group	SINK, active low
13	Q4	Output	User output 4 in the 2nd group	SINK, active low
14	Q5	Output	User output 5 in the 2nd group	SINK, active low
15	Q6	Output	User output 6 in the 2nd group	SINK, active low
16	Q7	Output	User output 7 in the 2nd group	SINK, active low
Power connector				
17	24 V	Power	24 V power supply	24 VDC power input
18	COM	Power	Power ground	24 VDC power common

- Terminal definition of the relay output module (AM600-0016ER)

SN	Network Name	Type	Function	Remarks
Interface for the 1st group of outputs				
1	Q0	Output	User output 0 in the 1st group	SINK output, active high
2	Q1	Output	User output 1 in the 1st group	SINK output, active high
3	Q2	Output	User output 2 in the 1st group	SINK output, active high
4	Q3	Output	User output 3 in the 1st group	SINK output, active high
5	Q4	Output	User output 4 in the 1st group	SINK output, active high
6	Q5	Output	User output 5 in the 1st group	SINK output, active high
7	Q6	Output	User output 6 in the 1st group	SINK output, active high
8	Q7	Output	User output 7 in the 1st group	SINK output, active high
9	COM0	Output common	Common	Common of the first group
Interface for the 2nd group of outputs				
10	Q0	Output	User output 0 in the 2nd group	SINK output, active high
11	Q1	Output	User output 1 in the 2nd group	SINK output, active high
12	Q2	Output	User output 2 in the 2nd group	SINK output, active high
13	Q3	Output	User output 3 in the 2nd group	SINK output, active high
14	Q4	Output	User output 4 in the 2nd group	SINK output, active high
15	Q5	Output	User output 5 in the 2nd group	SINK output, active high
16	Q6	Output	User output 6 in the 2nd group	SINK output, active high
17	Q7	Output	User output 7 in the 2nd group	SINK output, active high
18	COM1	Output common	Common	Common of the second group
Power connector				
1	24 V	Power input	24 VDC power supply	24 VDC power input
2	COM	Power supply common	Common	24 VDC power common

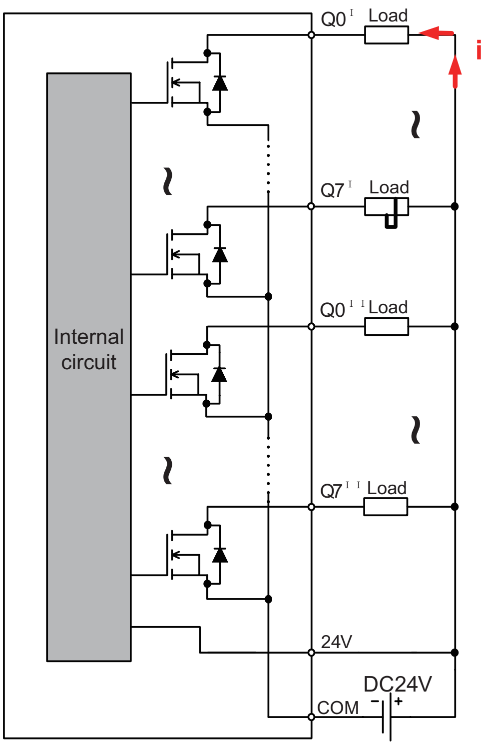
- AM600-0016ETN output wiring

AM600-0016ETN output wiring diagram



- AM600-0016ETP output wiring

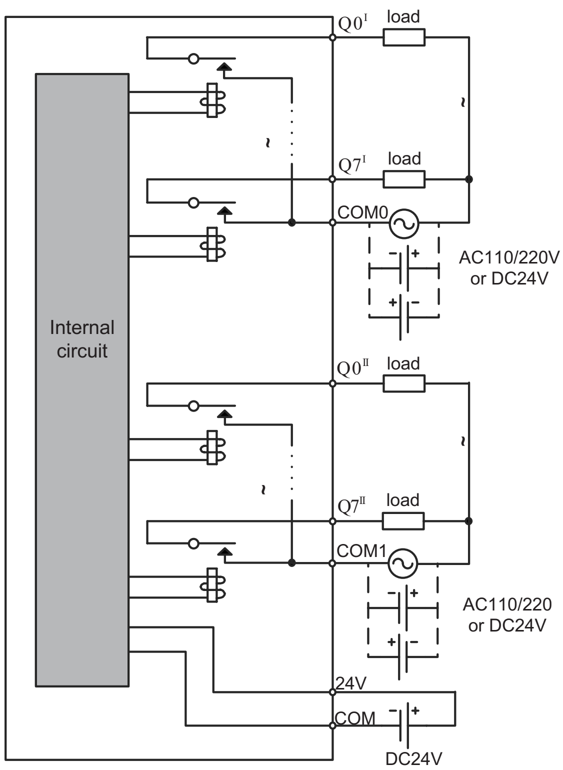
AM600-0016ETP output wiring diagram



10

- AM600-0016ER output wiring

AM600-0016ER output wiring diagram



(3) Module usage

When a local digital output extension module is connected to the main module, the SN of the Y port on the extension module follows that of the Y port on the main module. For example, the main module H3U connects the AM600-0016ETN. The SN of the last Y port of the main module is Y37. The SNs of the 16 Y ports of the extension module are Y40 to Y47 and Y50 to Y57. Use the same method to number Y ports of the following digital output extension modules.

Note: SN of Y ports of the extension module starts from units position 0 in octonary system. The relay output extension module and transistor output extension module can be connected to the relay or transistor main module.

10.2.4 Analog Input Module (AM600-4AD)

(1) Specifications

Item	Specifications
Input channel	4
Supply voltage	24 VDC (20.4 VDC to 28.8 VDC) (–15% to +20%)
Voltage input impedance	> 1 MΩ
Current sampling impedance	250 Ω
Voltage input range	Bipolar: ±5 V, ±10 V; Unipolar: +5 V, +10 V
Current input range	0 mA to 20 mA, 4 mA to 20 mA, ±20 mA
Resolution	16 bits
Sampling time	1 ms
Accuracy (normal temperature: 25°C)	Voltage: ±0.1%, current: ±0.1% (full ranges)

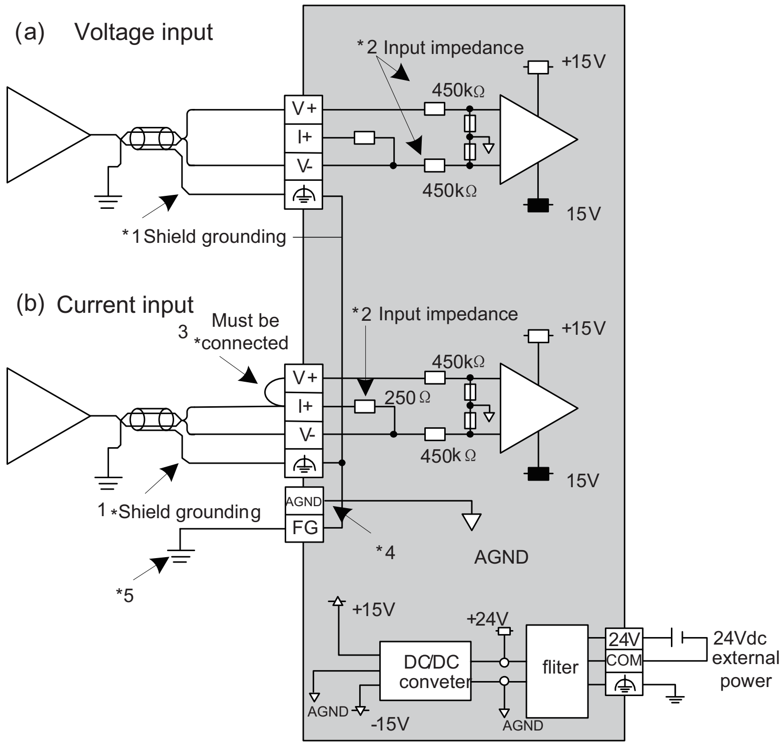
Item	Specifications
Accuracy (ambient temperature: 0–55°C)	Voltage: ±0.3%, current: ±0.8% (full ranges)
Voltage limits	±15 V
Current limits	±30 mA (transient), ±24 mA (average)
Maximum common mode voltage between channels	30 VDC
Isolation method	I/O terminals isolated from power supply; Non-isolation between channels.
System program updated via	USB interface

(2) Terminal definition and external wiring

SN	Network Name	Type	Function	Remarks
1	V+	Input	V+ of channel 0	Voltage input
2	VI-	Input	V-/I- of channel 0	Voltage/Current input
3	I+	Input	I+ of channel 0	Current Input
4	⏚	-	Shielding ground	Internally connected to housing ground
5	V+	Input	V+ of channel 1	Voltage input
6	VI-	Input	V-/I- of channel 1	Voltage/Current input
7	I+	Input	I+ of channel 1	Current Input
8	⏚	-	Shielding ground	Internally connected to housing ground
9	V+	Input	V+ of channel 2	Voltage input
10	VI-	Input	V-/I- of channel 2	Voltage/Current input
11	I+	Input	I+ of channel 2	Current Input
12	⏚	-	Shielding ground	Internally connected to housing ground
13	V+	Input	V+ of channel 3	Voltage input
14	VI-	Input	V-/I- of channel 3	Voltage/Current input
15	I+	Input	I+ of channel 3	Current Input
16	AGND	Analog ground	Analog ground	-
17	24 V	Power	24 V power supply	-
18	COM	Power ground	Power ground	-

- External wiring

Analog input wiring diagram



- Use 2-core shielded twisted pair cable for analog signal.
- Indicates input impedance of 4AD.
- For current input, terminal (V+) must be connected to terminal (I+).
- When the input signal is a differential signal, "AGND" can be connected to analog ground of compatible devices to eliminate the difference of common mode voltage between devices and ensure the accuracy of module sampling.
- The module should be mounted on a well-grounded metal bracket, and ensure that the metal shrapnel at the bottom of the module is in good contact with the bracket.

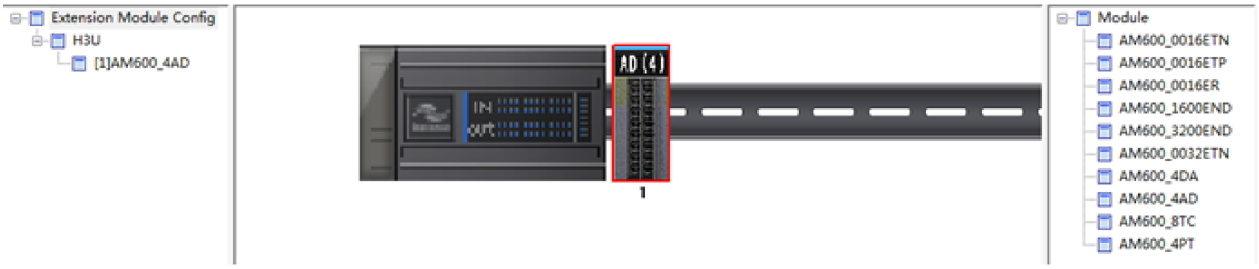
(3) Wiring precautions

- Do not bundle the cable together with AC cable, main lines, high voltage cable and so forth; otherwise, it may result in an increased noise, surge and induction.
- Apply single-point grounding for the shielding of shielded cable and solder sealed cable.
- Tubed and solderless crimp terminal cannot be used with terminal block. Using marking sleeve or insulation sleeve to cover the cable connector part of the crimp terminals is recommended.

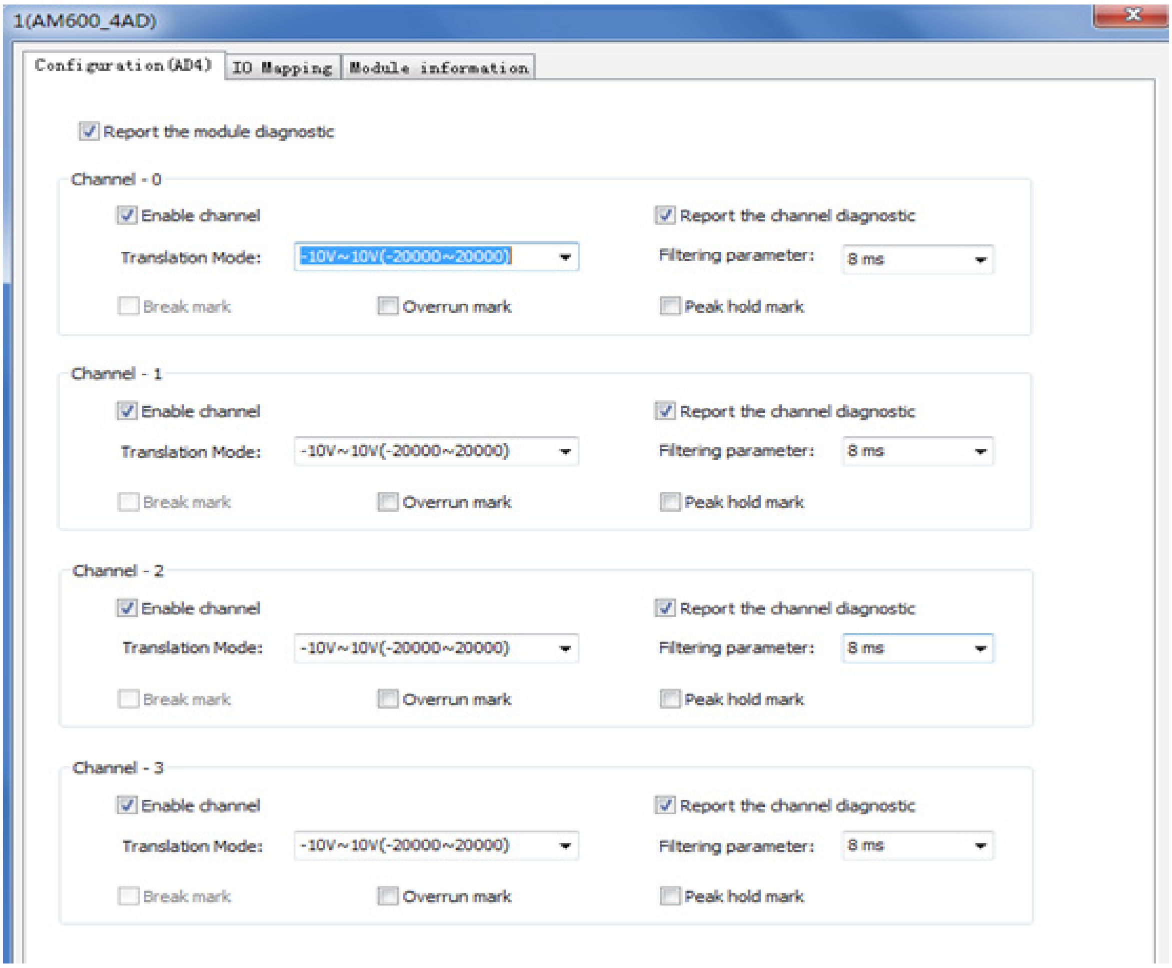
10

(4) Example for H3U+AM600-4AD programming

① Select the module AM600-4AD to be added from the module list. Double-click the module to automatically add it to the extension rack, or use the left mouse button to drag it onto the extension rack.



② Double-click the AM600-4AD module on the rack, and the configuration interface appears (as below).



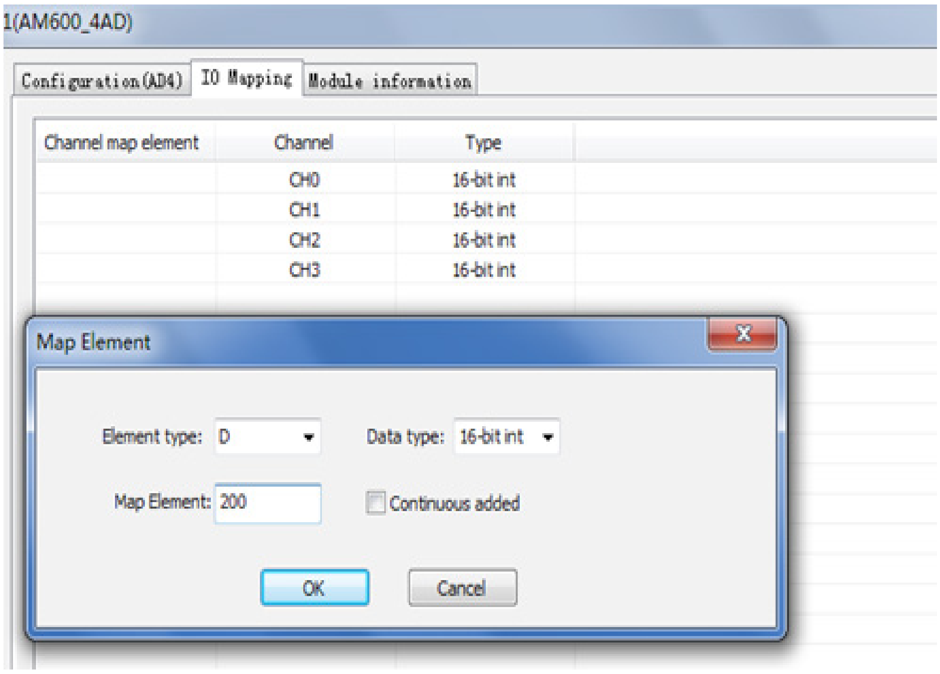
On the configuration interface, enable the corresponding channel. Disable unused channels to reduce the scan duration.

Select voltage or current of different ranges from the **Translate** drop-down list. In this example, **Translate** is configured as **-10V~10V** voltage input.

Select filter time from the **Filtering** drop-down list.

③ On the **IO Map** tab page, map CH0 of 4AD module to D200 of element D.

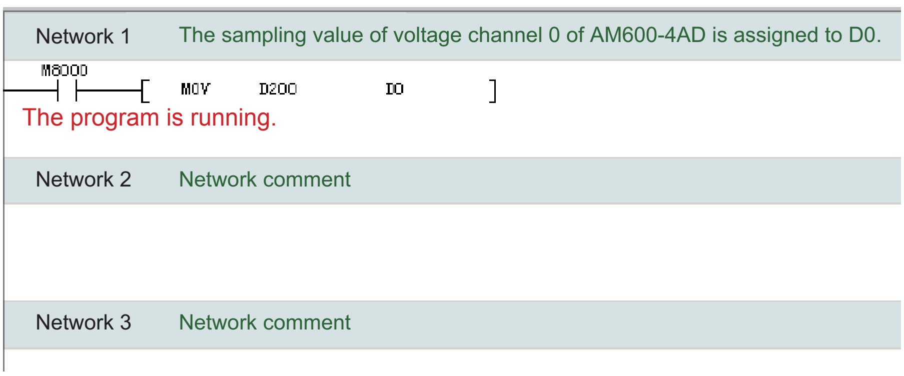
10



The relationship between mapping and actual input analog value is as follows:

	Rated Input Range	Rated Digital Value	Input Limit Range	Limiting Digital Value
Analog voltage input	–10 V to +10 V	–20,000 to +20,000	–11 V to +11 V	–22,000 to +22,000
	0 V to 10 V	0 to 20,000	–0.5 V to +10.5 V	–1000 to +21,000
	–5 V to +5 V	–20,000 to +20,000	–5.5 V to +5.5 V	–22,000 to +22,000
	0 V to 5 V	0 to 20,000	–0.25 V to +5.25 V	–1000 to +21,000
	1 V to 5 V	0 to 20,000	0.8 V to 5.2 V	–1000 to +21,000
Analog current input	–20 mA to +20 mA	–20,000 to +20,000	–22 mA to +22 mA	–22,000 to +22,000
	0 mA to 20 mA	0 to 20,000	–1 mA to +21 mA	–1000 to +21,000
	4 mA to 20 mA	0 to 20,000	3.2 mA to 20.8 mA	–1000 to +21,000

④ Use ladder graphic programming language to program AD sampling. Change mapping tag of CH0 from D200 to D0.



⑤ After successful compiling, download the project and run it.

10

10.2.5 Analog Output Module (AM600-4DA)

(1) Specifications

Item	Specifications
Output channel	4
Supply voltage	24 VDC (20.4 VDC to 28.8 VDC) (–15% to +20%)
Voltage output load	1 kΩ to 1 MΩ
Current load impedance	0 Ω to 600 Ω
Output voltage range	Bipolar: ±5 V, ±10 V; Unipolar: +5 V, +10 V
Output current range	4 mA to 20 mA, 0 mA to 20 mA
Accuracy (normal temperature: 25°C)	Voltage: ±0.1%, current: ±0.1% (full ranges)
Accuracy (ambient temperature: 0–55°C)	Voltage: ±0.15%, current: ±0.8% (full ranges)
Resolution	16 bits
Conversion time	1 ms/ch.
Isolation method	I/O terminals isolated from power supply; Non-isolation between channels.
Output short-circuit protection	Yes
System program updated via	USB interface

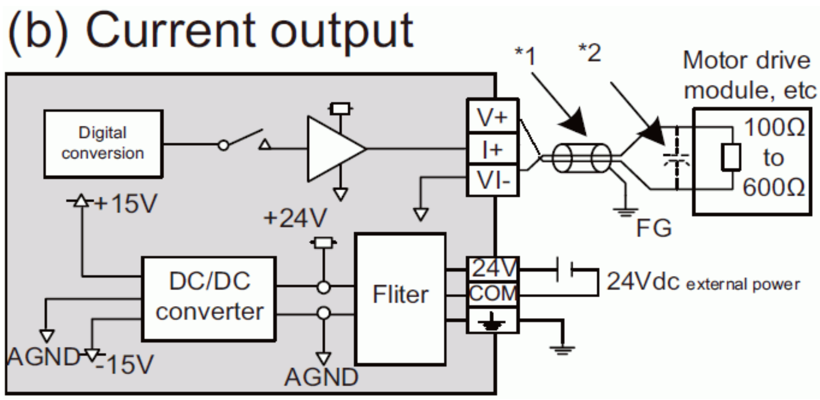
(2) Terminal definition and external wiring

- Terminal definition

SN	Network Name	Type	Function	Remarks
1	V+	Output	V+ of channel 0	Voltage output
2	VI-	Output	V-/I- of channel 0	Voltage/Current output
3	I+	Output	I+ of channel 0	Current output
4	⏚	-	Shielding ground	Internally connected to housing ground
5	V+	Output	V+ of channel 1	Voltage output
6	VI-	Output	V-/I- of channel 1	Voltage/Current output
7	I+	Output	I+ of channel 1	Current output
8	⏚	-	Shielding ground	Internally connected to housing ground

SN	Network Name	Type	Function	Remarks
9	V+	Output	V+ of channel 2	Voltage output
10	VI-	Output	V-/I- of channel 2	Voltage/Current output
11	I+	Output	I+ of channel 2	Current output
12	⏚	-	Shielding ground	Internally connected to housing ground
13	V+	Output	V+ of channel 3	Voltage output
14	VI-	Output	V-/I- of channel 3	Voltage/Current output
15	I+	Output	I+ of channel 3	Current output
16	AGND	Analog ground	Analog ground	-
17	24 V	Power	24 V power supply	-
18	COM	Power ground	Power ground	-

- External wiring

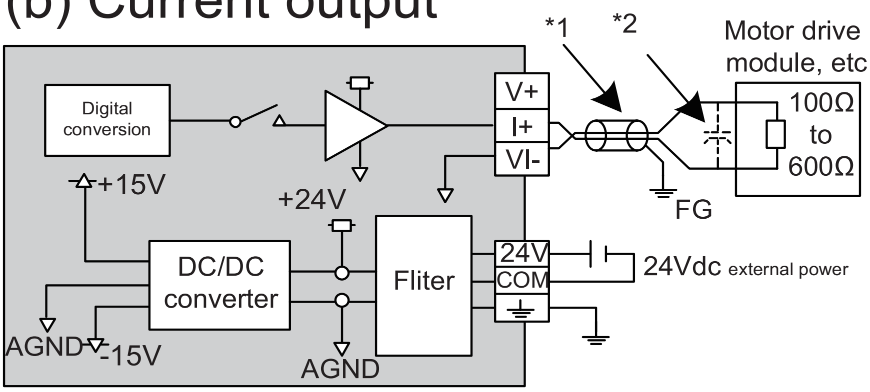


① Connection for voltage-controlled signal

NOTE

- Use 2-core shielded twisted pair cable as power cable.
- If noises or ripples are generated in external wiring, connect a capacitor of 0.1 to 0.47mF, 25V between terminals V+/I+ and VI-.

(b) Current output



② Connection for current-controlled signal

NOTE

- Use 2-core shielded twisted pair cable as power cable.
- If noises or ripples are generated in external wiring, connect a capacitor of 0.1 to 0.47mF25V between terminals V+/I+ and VI-.

③ Wiring precautions

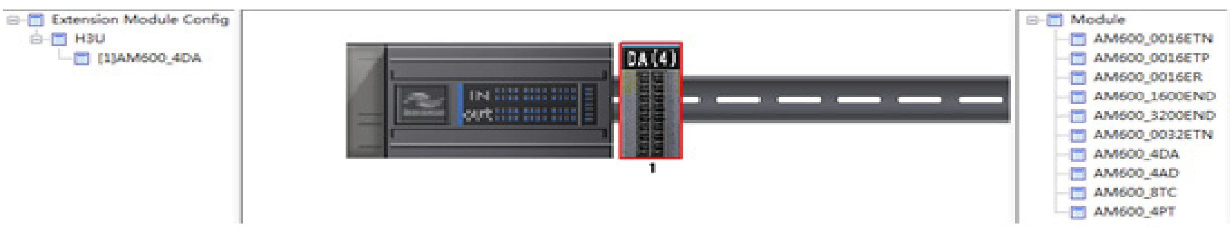
Do not bundle the cable together with AC cable, main lines, high voltage cable and so forth; otherwise, it may result in an increased noise, surge and induction.

Apply single-point grounding for the shielding of shielded cable and solder sealed cable.

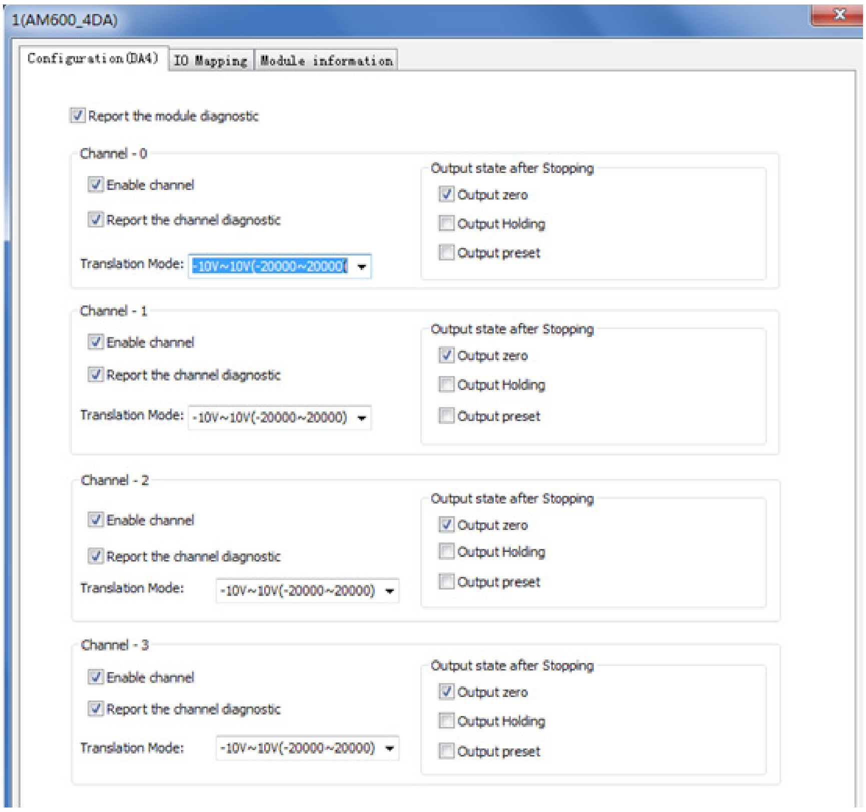
Tubed and solderless crimp terminal cannot be used with terminal block. Using marking sleeve or insulation sleeve to cover the cable connector part of the crimp terminals is recommended.

3) Example for H3U+AM600-4DA Programming

① Select the module AM600-4DA to be added from the module list. Double-click the module to automatically add it to the extension rack, or use the left mouse button to drag it onto the extension rack.



② Double-click the AM600-4DA module on the rack, and the configuration interface appears (as below).

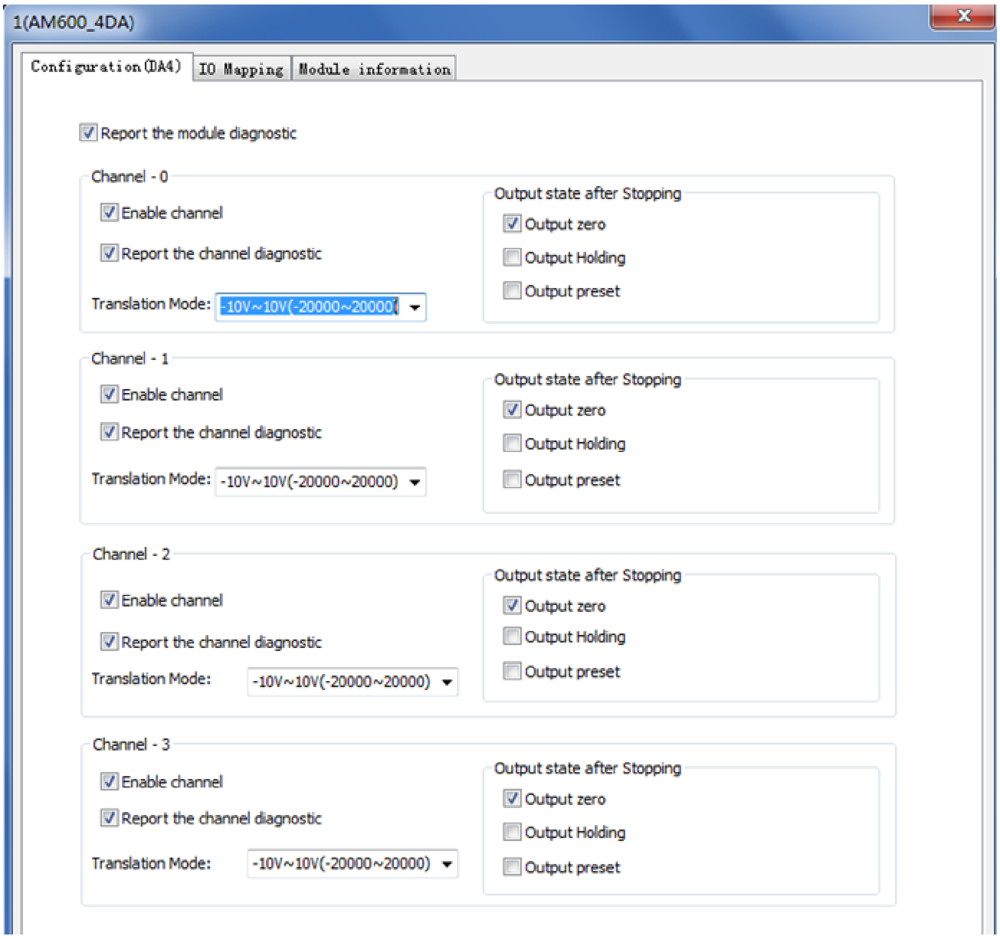


On the **General Configuration** interface of the AM600-4DA module, enable Channel-0, and configure **Conversion Mode** as **-10V~10V** voltage output. **State Output After Stop** can also be configured.

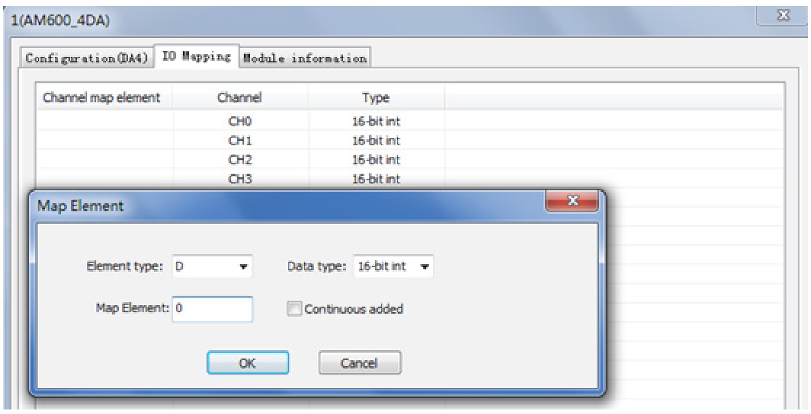
On the configuration interface, enable the corresponding channel. Disable unused channels to reduce the scan duration.

Select voltage or current of different modes from the **Translate Mode** drop-down list. In this example, **Translate Mode** is configured as **-10V~10V** voltage output.

Output state after Stopping sets the module output condition when the PLC is in the stopped state. The options include **Output zero**, **Output hold**, and **Output preset**.



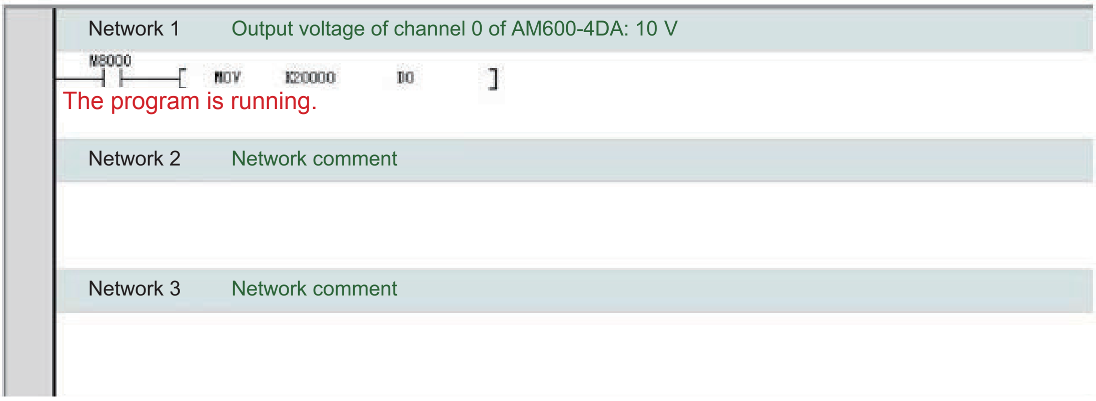
③ On the **IO Map** tab page, map CH0 of 4DA module to D0 of element D.



The relationship between elements and actual analog values is as follows:

	Rated Output Range	Rated Digital Value	Output Limit Range	Limiting Digital Value
Analog Voltage Output	–10 V to +10 V	–20,000 to +20,000	–11 V to +11 V	–22,000 to +22,000
	0 V to 10 V	0 to 20,000	–0.5 V to +10.5 V	–1000 to +21,000
	–5 V to +5 V	–20,000 to +20,000	–5.5 V to +5.5 V	–22,000 to +22,000
	0 V to 5 V	0 to 20,000	–0.25 V to +5.25 V	–1000 to +21,000
	1 V to 5 V	0 to 20,000	0.8 V to 5.2 V	–1000 to +21,000
Analog current output	0 mA to 20 mA	0 to 20,000	0 mA to 21 mA	0 to 21,000
	4 mA to 20 mA	0 to 20,000	3.2 mA to 20.8 mA	–1000 to +21,000

④ Use ladder graphic programming language to program DA output. As –10 V to +10 V corresponds to the digital of –20,000 to +20,000, assign 20,000 to D0, and the output voltage of the module's channel 0 is +10 V.



⑤ After successful compiling, download the project and run it.

10.3 Remote Extension Modules

10.3.1 AM600RTU-COP Remote Extension Module

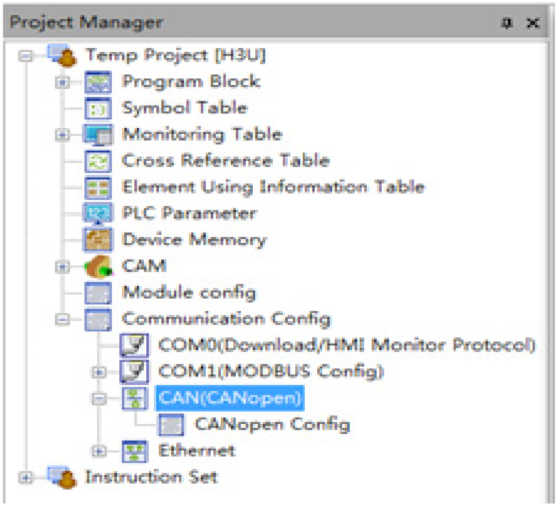
AM600-RTU-COP is a CANopen communication module, which can be directly connected to a local digital or analog module. The H3U connects to AM600RTU-COP over the CANopen bus for access to remote extension modules.

(1) Specifications

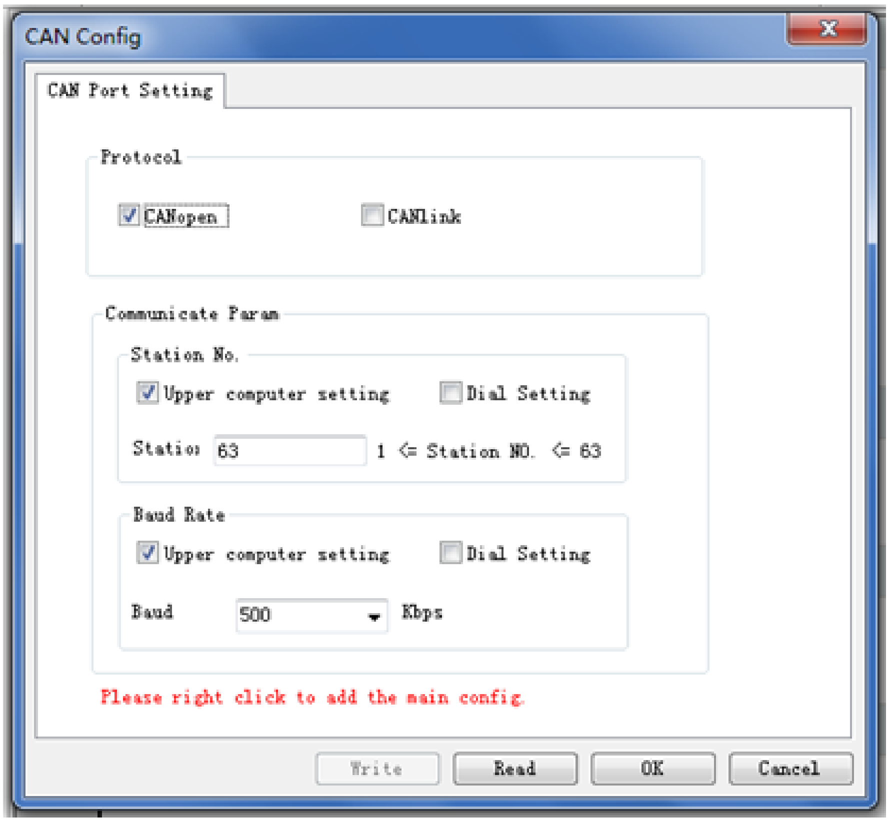
Item	Specifications
Protocol for communication with CPU module	CANopen
CANopen communication rate	Max.: 1 Mbps
Station number range	1 to 127 (The station number can be set with two round DIP switches.)
Expandability of subsequent I/O modules	Able to expand up to eight I/O modules
Protocol for communication with I/O extension module	Module extension bus protocol; data exchange rate: 8 Mbps, 4 Mbps, and 2 Mbps, adjustable based on the number of connected I/O modules (The rate is inversely proportional to the module quantity.)
CANopen network interface	One DB9 male connector interface

(2) Networking configuration

① Start AutoShop and click **Communication Config** on the project management interface.

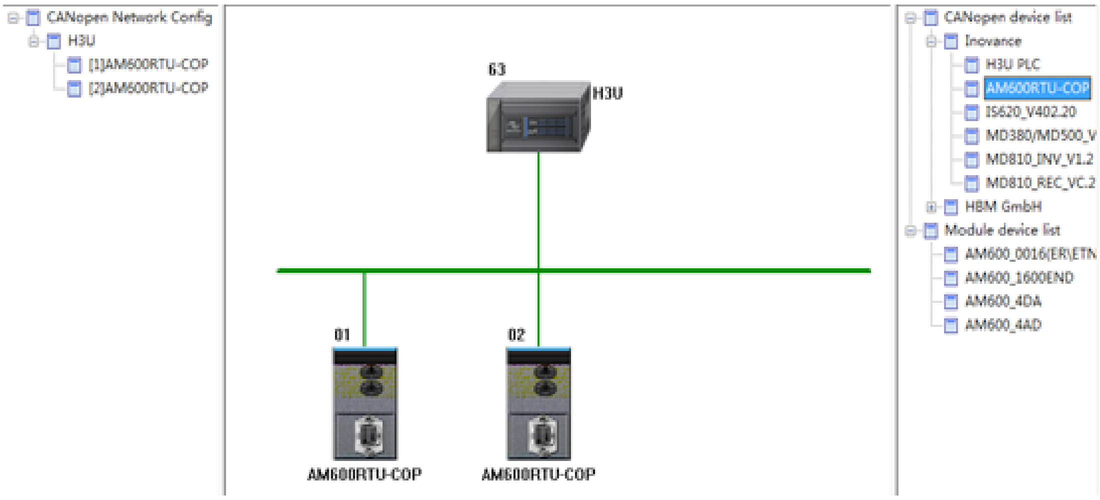


② Right-click **CAN Config** and choose **CANOpen**.



Click **OK**. The following interface is displayed.

10



(3) Addition of the CANopen device

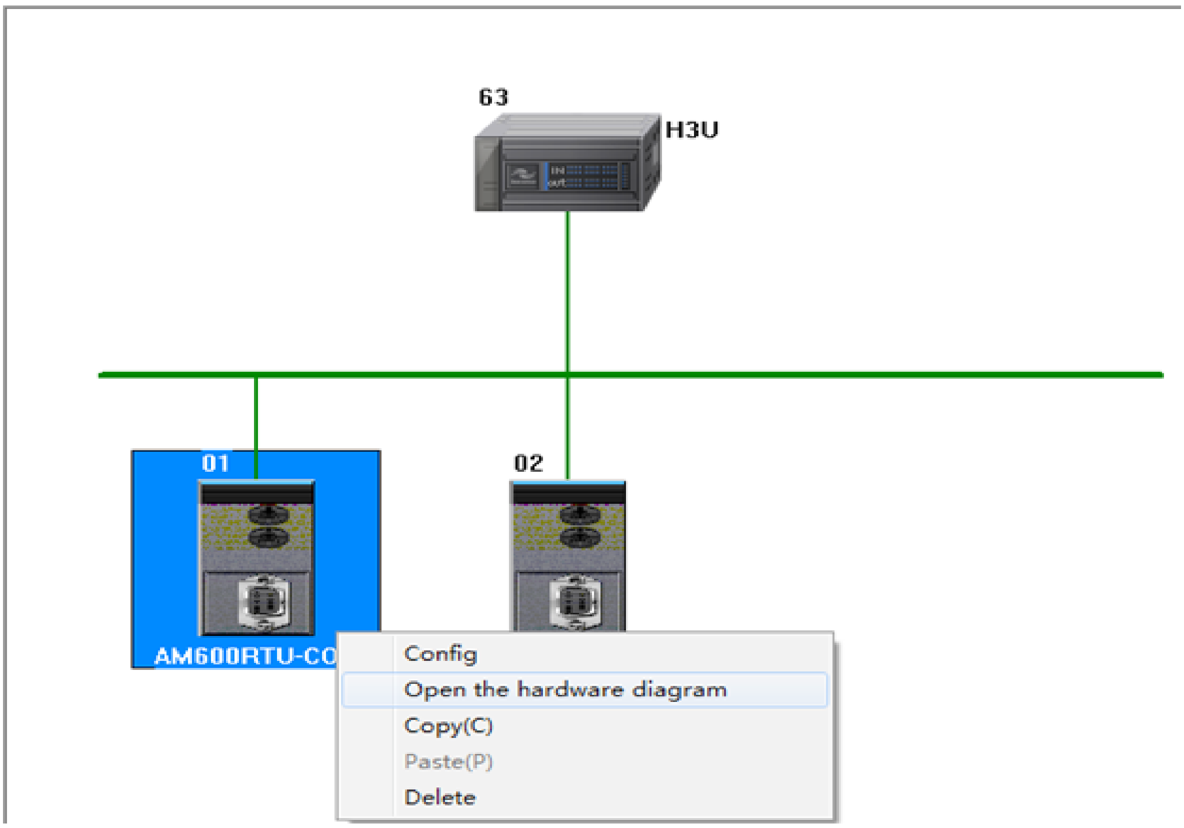
① Add the AM600RTU-COP CANopen device.

Choose **CANopen device list** > **Inovance** and double-click **AM600RTU-COP** or use the left mouse button to drag it to the pane on the right. The AM600RTU-COP communication module is automatically connected to the H3U.

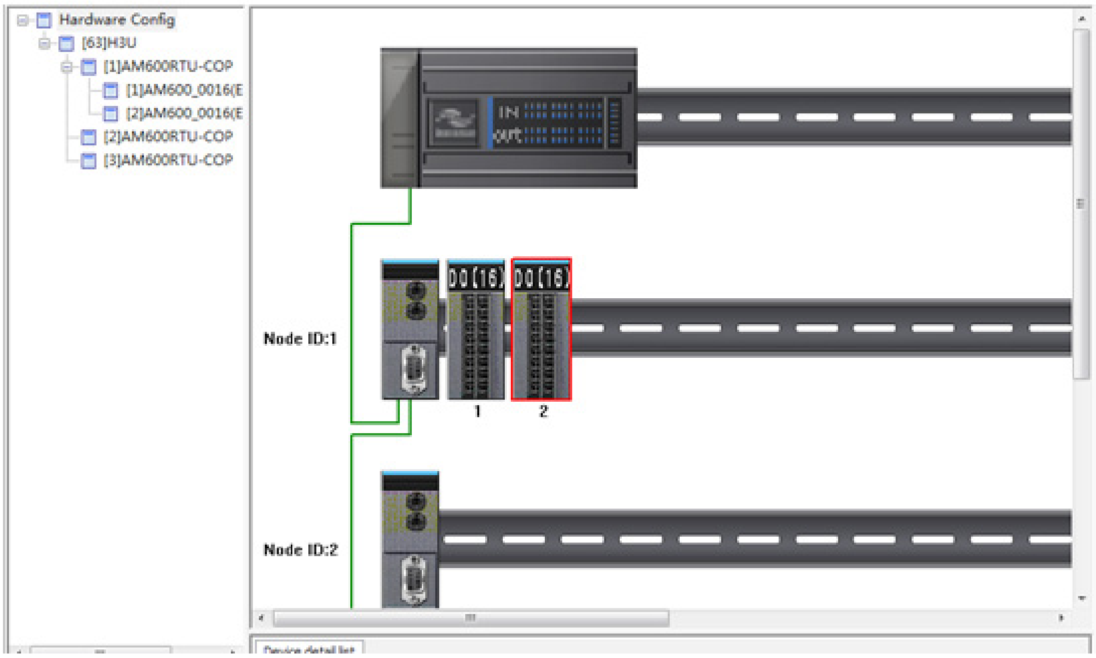
② Configure the extension module.

After AM600RTU-COP is added, configure the hardware of the digital or analog extension module connected to AM600RTU-COP.

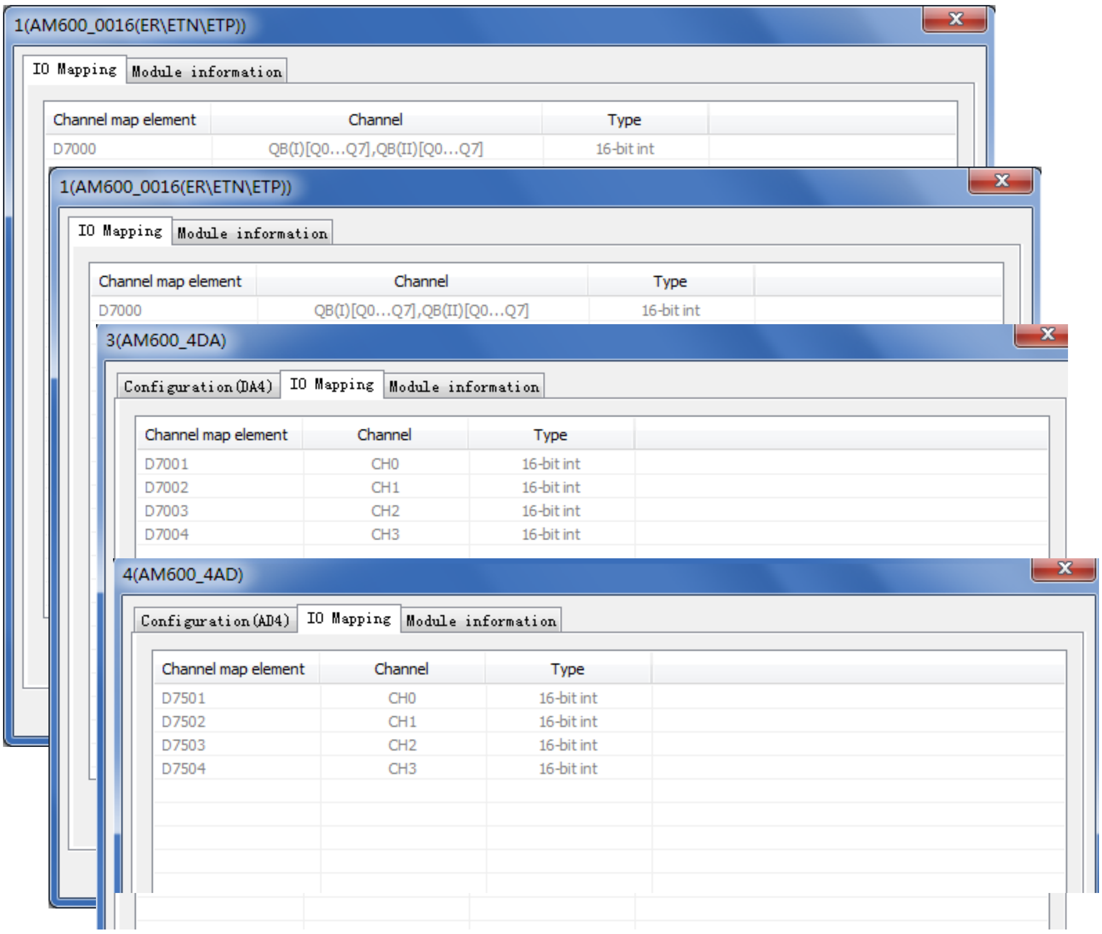
Right-click the communication module to be configured and choose **Open the hardware program** from the context menu. The module configuration interface is displayed.



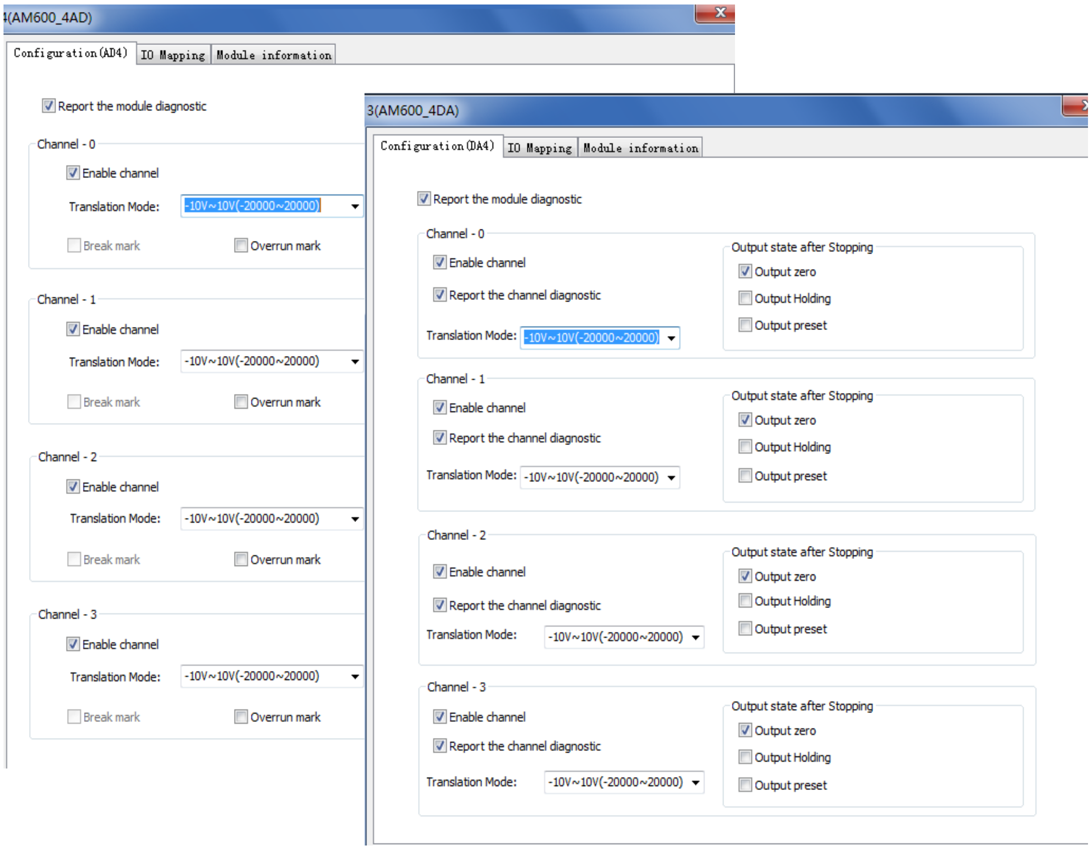
Add the corresponding digital or analog extension module to the network node rack based on the actual module usage condition.



Double-click the extension module. The **IO Map** tab page shows the register mapping. The main PLC communicates with the communication module over CANopen and reads/writes the map register for remote I/O access.

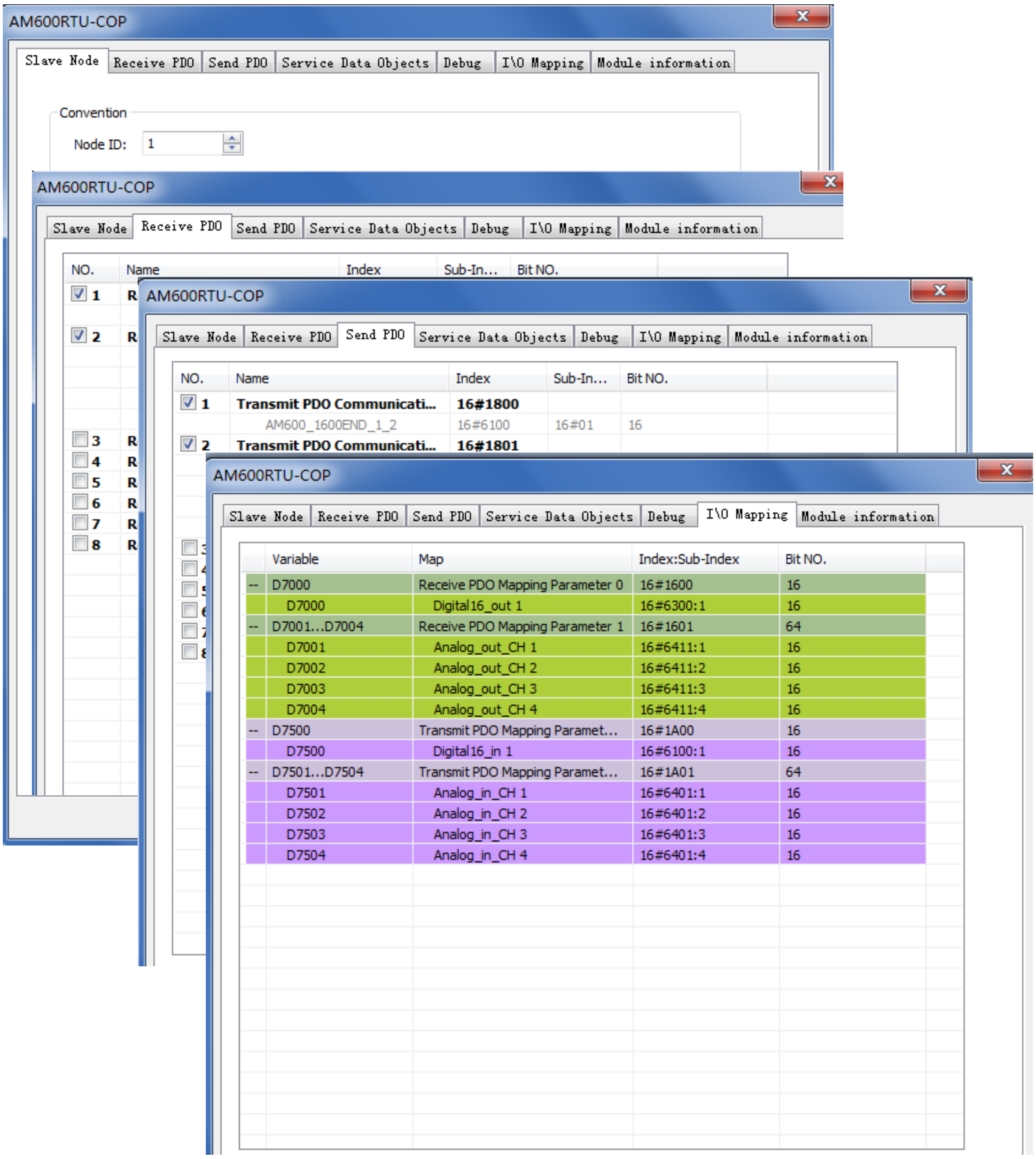


If an analog module is used, you can configure the analog input or output mode on the configuration interface. For details, see the description of local module analog setting.



4) CANopen configuration

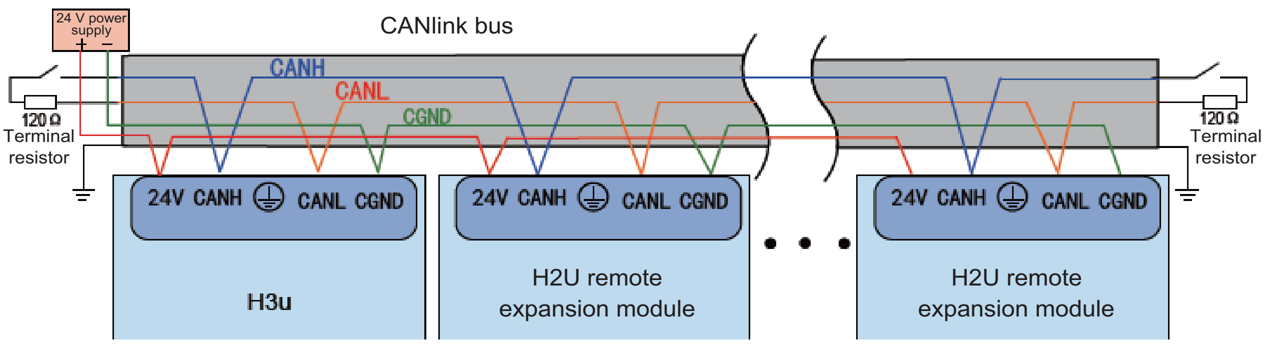
On the CANopen configuration interface, double-click AM600RTU-COP or right-click it and choose **Configuration**. The **AM600RTU-COP** configuration interface is displayed, where you can view or set communication parameters. For details, see section Page 635, "9.7 CANopen Communication" .



10.3.2 CANlink Remote Extension Module

The CANlink bus can be used to connect to H2U series remote modules for remote extension.

The following figure shows the network diagram of connection between the H3U and H2U series remote modules over the CANlink bus.



For details about the configuration of H3U CANlink communication, see section Page 616, "9.6 CANlink Communication".

10

11 Interrupt

Chapter 11 Interrupt

11.1 Overview

11.1.1 Overview

An interrupt is a trigger signal that causes the device to immeately execute interrupt programs (interrupt subprograms) regardless of the calculation cycle of the main program.

During sequential programming, the calculation cycle causes a delay and time deviation affects mechanical actions. The interrupt function can help solve the problem.

11.1.2 Type

The following table lists interrupt types for an H3U standard model.

		Overview	
I	Interrupt	External interrupt	X000-X007 input interrupt numbered I00 □, I10 □, I20 □, I30 □, I40 □, I50 □, I56 □, and I57 □, eight points (□ 0 incates a falling pulse interrupt, and 1 indicates a rising pulse interrupt.) When the pulse interrupt disabling flag register is set to ON, the corresponding input interrupt is disabled.
		Timing interrupt	I6□□, I7□□, I8□□, 3 points (□□ = 1 to 99, time base = 1 ms)
		Counting complete interrupt	I010, I020, I030, I040, I050, I060, I070, and I080. Eight points (for the DHSCS instruction)
		Pulse complete interrupt	I502 to I506, five points

The following table lists interrupt types for an H3U-PM motion control model.

		Overview	
I	Interrupt	External interrupt	PG0-PG2 input interrupt numbered I00 □, I10 □, and I20 □, three points (□ 0 indicates a falling pulse interrupt, and 1 indicates a rising pulse interrupt.) When the pulse interrupt disabling flag register is set to ON, the corresponding input interrupt is disabled.
		Timing interrupt	I6□□, I7□□, and I8□□, three points (□□ = 1 to 99, time base = 1 ms)
		Counting complete interrupt	I010, I020, I030, I040, I050, I060, I070, and I080, eight points (for the DHSCS instruction)
		Pulse complete interrupt	I502 to I504, three points

11

11.2 External Interrupt

11.2.1 Overview

The device uses X000-X007 inputs to execute the interrupt subprogram.

As external inputs can be processed regardless of the calculation cycle of the PLC, external interrupts can be used for high-speed control and short pulse acquisition.

11.2.2 Type

1) External interrupt event numbers and actions for an H3U standard model

Input Number	Interrupt Number		Disable Interrupt
	Rising pulse interrupt	Falling pulse interrupt	
X00	I001	I000	M8050
X01	I101	I100	M8051
X02	I201	I200	M8052
X03	I301	I300	M8053
X04	I401	I400	M8054
X05	I501	I500	M8055
X06	I561	I560	M8080
X07	I571	I570	M8081

Note: When M8050 to M8055, M8080, and M8081 are ON, the corresponding interrupt events are disabled.

2) External interrupt event numbers and actions for an H3U-PM model

Input Number	Interrupt Number		Disable Interrupt
	Rising pulse interrupt	Falling pulse interrupt	
PG0	I001	I000	M8050
PG1	I101	I100	M8051
PG2	I201	I200	M8052

Note: When M8050 to M8052 are ON, the corresponding interrupt events are disabled.

NOTE

Pay attention to the following tips.

- Prohibited reuse of input numbers
  - The number of the input relay as an interrupt pointer cannot be the same as the ID of any application instruction within the same input range, for example, the high-speed counter instruction, pulse capture instruction, and pulse density instruction.
- Automatic input filter adjustment
  - When the input interrupt pointer I□0□ is specified, the input filter of the input relay will be automatically used for high-speed reading. Therefore, you do not need to use the REFF instruction and special register D8020 (input filter adjustment) to change the filter adjustment.
  - In addition, the input filter of the input relay not used as an external signal input interrupt pointer remains effective for 10 ms (initial value).

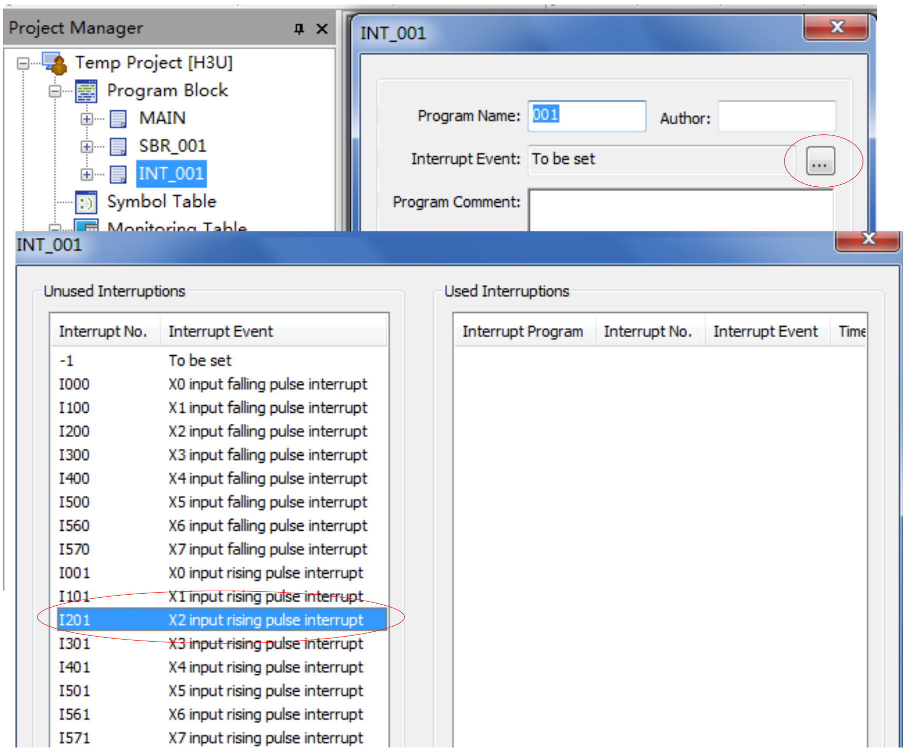
- Reuse of pointer numbers

• Rising pulse interrupt and falling pulse interrupt (for example, I001 and I000) sharing one input number cannot be written simultaneously.

11.2.3 Application

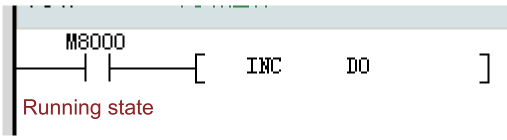
- Program used for counting rising pulses corresponding to the external input interrupt X02

① Establishing an X02 rising pulse interrupt subprogram I201

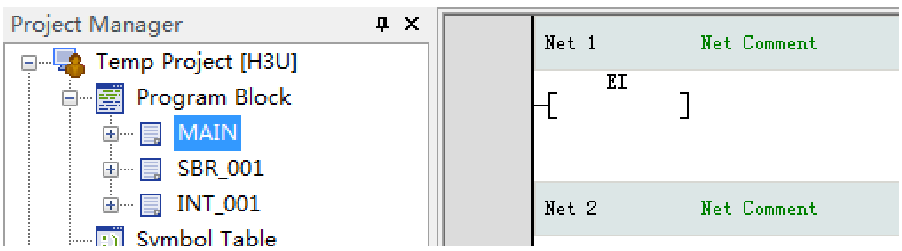


② Content of the interrupt subprogram

D0 increases by 1 each time an X02 rising pulse interrupt is generated.



③ Enabling an interrupt (EI) in the main program (MAIN)



11.3 Timing Interrupt

11.3.1 Overview

Regardless of the calculation cycle of the PLC, the device executes an interrupt program every 1 to 99 ms.

The function can be used to process programs at a high speed when the calculation cycle of the main program is long, or to execute programs at particular intervals during sequential calculation.

11.3.2 Type

Pointer Number	Interrupt Cycle	Disable Interrupt
I6□□	Enter integers (1 to 99) in "□□" For example, I710 means a timing interrupt per 10 ms.	M8056
I7□□		M8057
I8□□		M8058

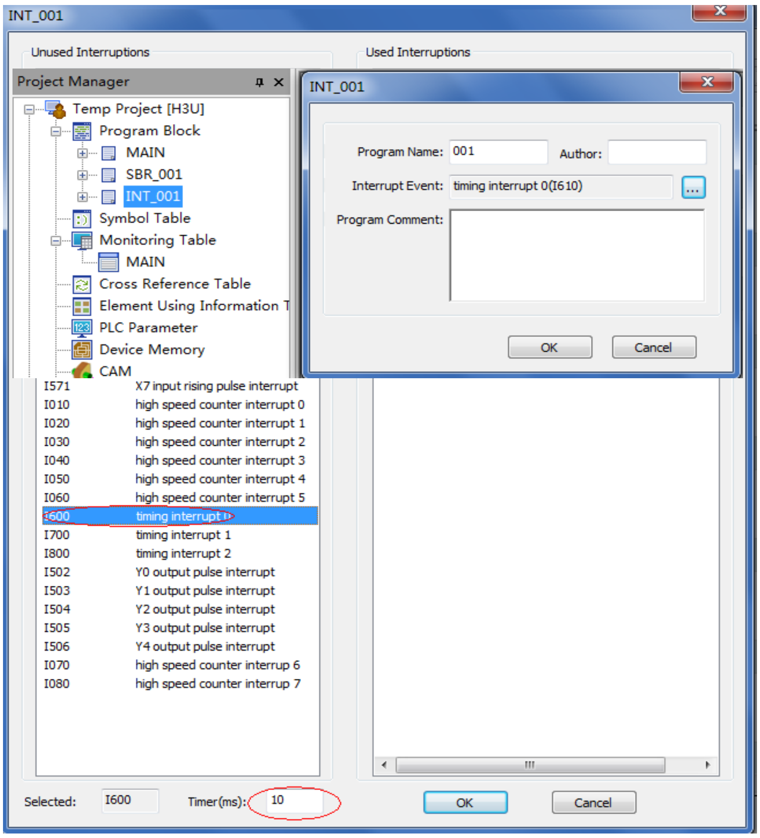
NOTE

- Note: When M8056 and M8057 are ON, the corresponding interrupt events are disabled.
- Pointer numbers (I6, I7, and I8) cannot be reused.

11.3.3 Application

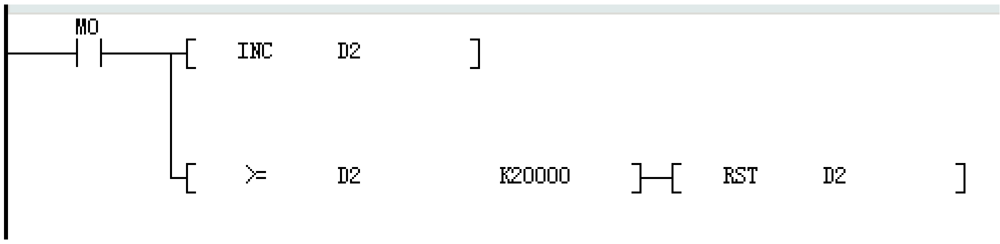
- Performing data addition operation every 10 ms and comparing the result with the preset value

① Establishing a 10 ms timing interrupt subprogram I610

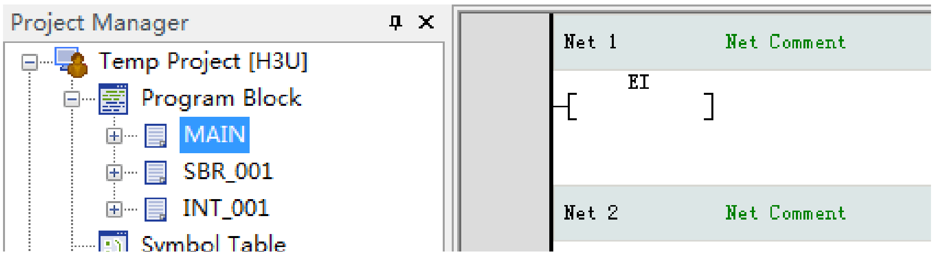


② Content of the interrupt subprogram

After M0 is connected, D2 increases by 1 every 10 ms. When D2 is greater than K20000, D2 is reset.



③ Enabling an interrupt (EI) in the main program (MAIN)



11.4 Counting Complete Interrupt

11.4.1 Overview

The function is used with the DHSCS comparison setting instruction. When the current value of the high-speed counter reaches the specified one, the device executes an interrupt program.

11.4.2 Type of High-speed Counting Interrupt

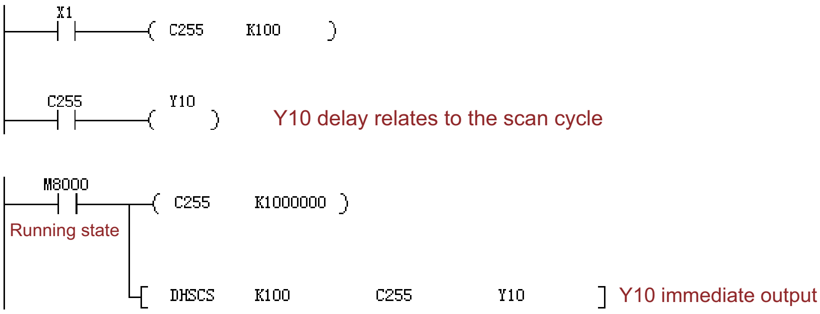
Pointer Number	Disable Interrupt
I010, I020, I030, I040, I050, I060, I070, and I080	M8059

NOTE

- When M8059 is ON, all counting interrupt events are disabled.
- Pointer numbers cannot be reused.

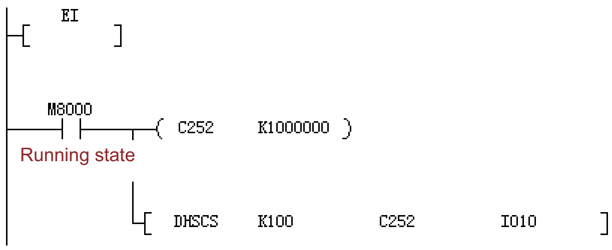
11.4.3 Application

- Example 1:

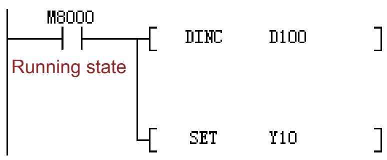


- Example 2:

a) Main program:



b) I010 interrupt subprogram



I0x0 (x = 1 to 8) can be specified based on the D operand range of the DHSCS instruction. When the counter value reaches the preset one, the device executes an interrupt program.

If M8059 is set to ON, all high-speed counter interrupts are disabled.

Note differences of the ON signal on D device with I010 or Y, M, or S outputs.

1. Y output: When the present value of C252 changes from 99 to 100 or from 101 to 100, Y is set to ON immediately and remains ON. Even if values of C252 and K100 are not equal by comparison, Y remains ON, unless it is reset.

2. I010: When the present value of C251 changes from 99 to 100 or from 101 to 100, I010 will trigger only one interrupt and will not always remain ON.

11.5 Pulse Complete Interrupt

11.5.1 Overview

On an H3U standard PLC, when special bits M8352, M8372, M8392, M8412, and M8432 (corresponding to Y0 to Y4, respectively) are ON, the pulse output complete interrupt function can be enabled by executing the PLSY, PLSR, DRVA, or DRVI positioning instruction.

The following table lists the correlation.

Port Number	Special Bit in Use	User Interrupt
Y00	M8352	I502
Y01	M8372	I503
Y02	M8392	I504
Y03	M8412	I505
Y04	M8432	I506

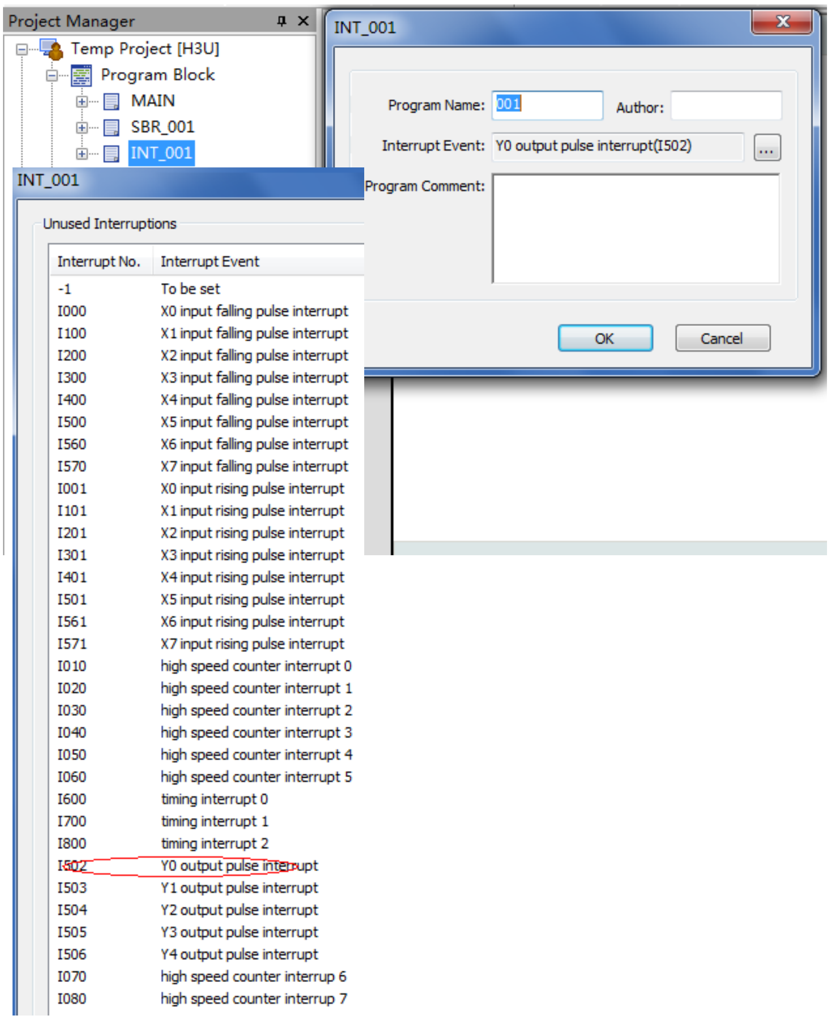
On an H3U-PM PLC, when special bits M8352, M8372, and M8392 (corresponding to the x-axis, y-axis, and z-axis, respectively) are ON, the pulse output complete interrupt function can be enabled by executing the PLSY, PLSR, DRVA, or DRVI positioning instruction.

The following table lists the correlation.

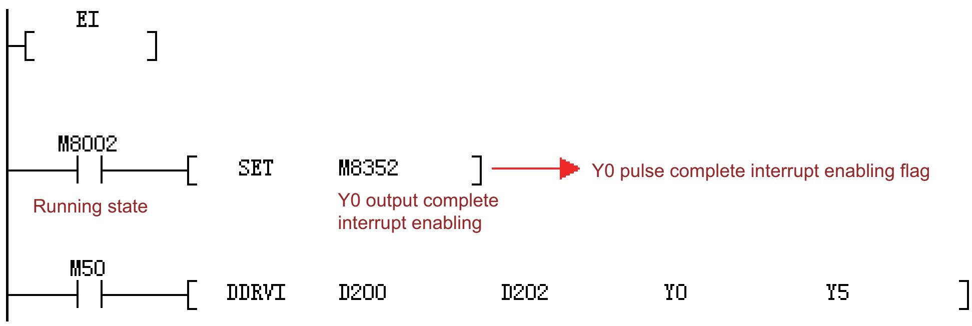
	Interrupt Object	Correlation with H3U
X axis	X axis output complete interrupt	Y0 pulse output interrupt I502
Y axis	Y axis output complete interrupt	Y1 pulse output interrupt I503
Z axis	Z axis output complete interrupt	Y2 pulse output interrupt I504

11.5.2 Application

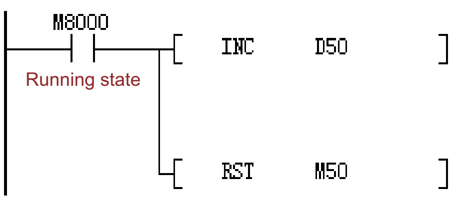
① Establishing a Y0 high-speed output port pulse complete interrupt I502



② Enabling an interrupt in the main program



③ Pulse complete interrupt INT program



11

12 Subprogram

Chapter 12 Subprogram

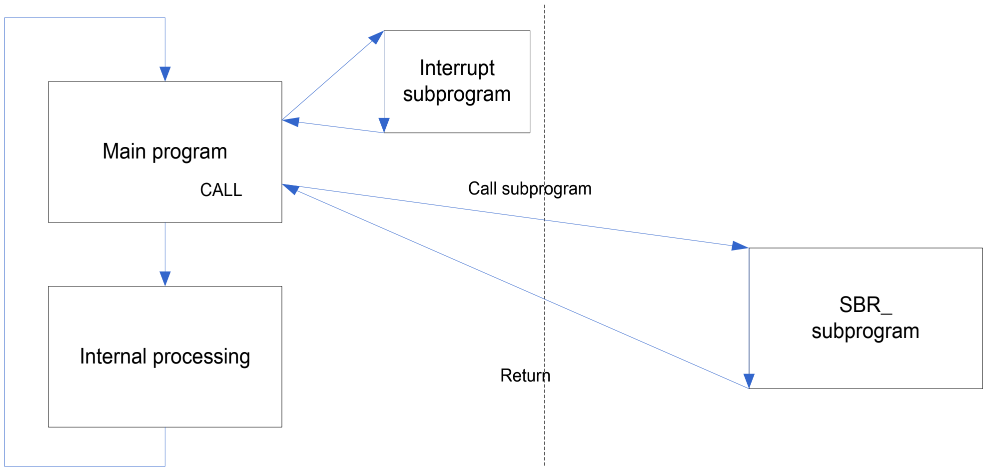
12.1 Overview

12.1.1 H3U Subprogram

		Overview	
SBR	For the CALL instruction	Up to 512 subprograms are supported. The subprograms can be set to general subprograms, encrypted subprograms, subprograms with parameters, and encrypted subprograms with parameters. The capacities of encrypted subprograms, subprograms with parameters, and general subprograms are not restricted. Such three types of subprograms share the 64K-step capacity of the system.	
P	For the CJ instruction	512 points, used with the LBL instruction	
I	Interrupt subprogram	External interrupt	X000-X007 input interrupt numbered I00 □, I10 □, I20 □, I30 □, I40 □, I50 □, I56 □, and I57 □, eight points (□ 0 indicates a falling pulse interrupt, and 1 indicates a rising pulse interrupt.) When the pulse interrupt disabling flag register is set to ON, the corresponding input interrupt is disabled.
		Timing interrupt	I6□□, I7□□, and I8□□, three points (□□ = 1 to 99, time base = 1 ms)
		Counting complete interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points (used by the DHSCS instruction)
		Pulse complete interrupt	I502 to I506, five points
MC	Motion control subprogram (Only supported by the H3U-PM series)	A maximum of 64 motion control subprograms are supported, numbered from MC0 to MC63. In addition, one G-code subprogram numbered CNC00 (corresponding to MC10000) is supported. The G-code subprogram file supports multiple Oxxxx codes numbered from O0000 to O9999. The capacities of motion subprograms and other subprograms are not limited. They share the 64K-step capacity of the system.	

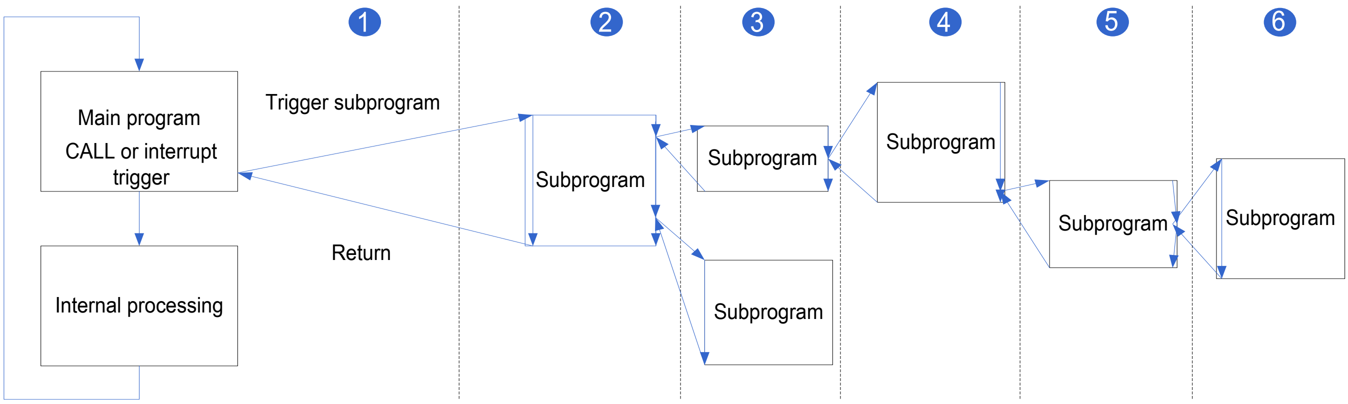
12.1.2 H3U Subprogram Execution Mechanism

The following figure shows the execution logic and cyclic scan mode of the main program and subprograms.



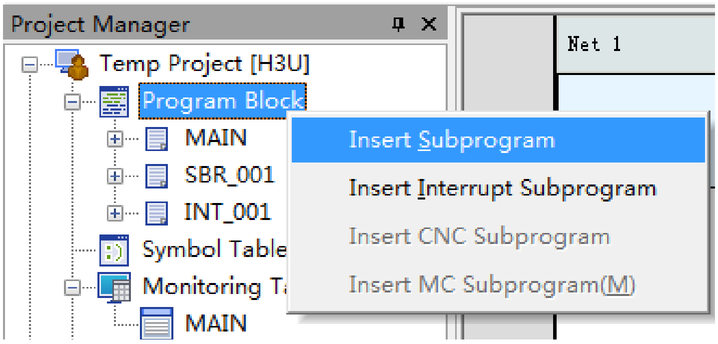
- Layers of nested subprograms

A maximum of six subprograms can be nested. The main program calls a subprogram as the first layer. One layer is added each time a subprogram is called. If a subprogram has returned, no layer is added, as shown in the following figure.

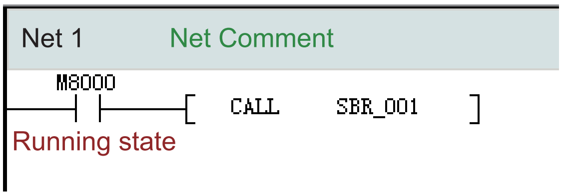


12.2 Application of a General Subprogram

1) Creating a general subprogram



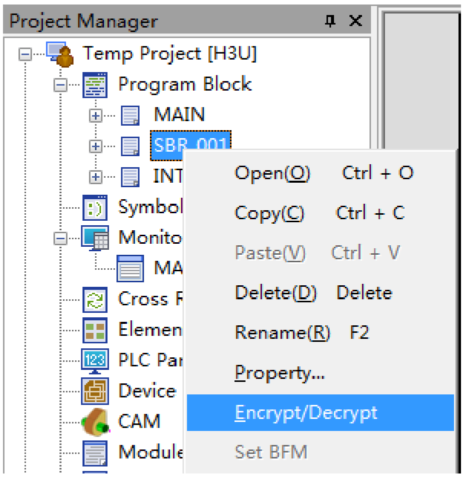
2) Calling a general subprogram



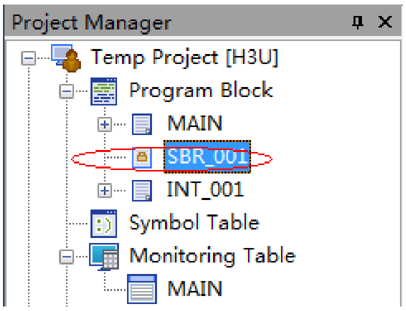
12.3 Application of an Encrypted Subprogram

12.3.1 Encrypting a General Subprogram

For example, right-click **SBR_001** and select **Encrypt/Decrypt** to encrypt SBR_001.



After encryption:



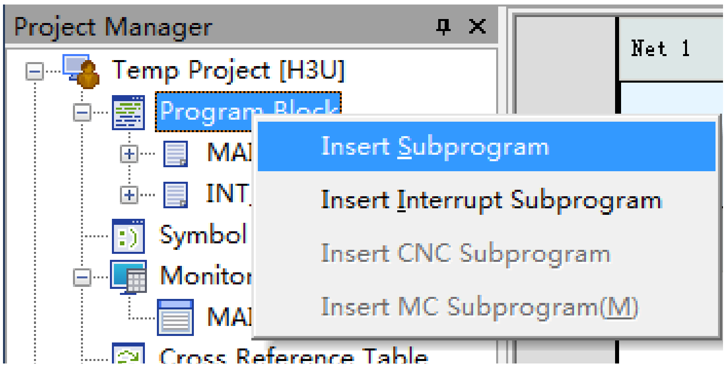
12.3.2 Calling an Encrypted Subprogram

An encrypted subprogram is called the same way as a general subprogram.

12.4 Application of a Subprogram with Parameters

Subprograms with parameters can be called on an H3U PLC. A subprogram with parameters provides local variables (VM and VD variables), allows parameter transmission, and defines I/O parameters.

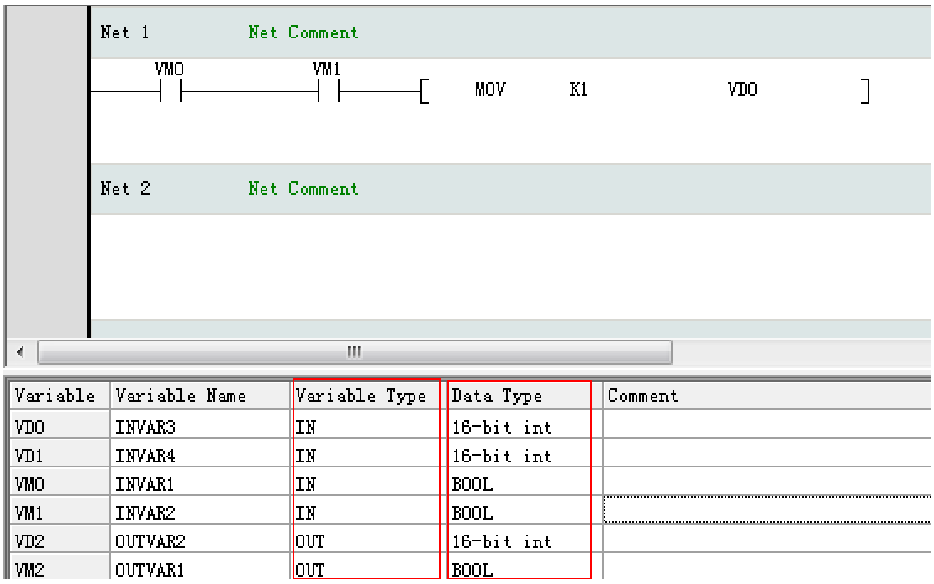
12.4.1 Creating a Subprogram with Parameters



12.4.2 Defining I/O Parameters

3) Variable address

A maximum of 32 VM elements and 96 VD elements are supported. Apart from I/O parameters, VM and VD variables can be used as temporary variables.

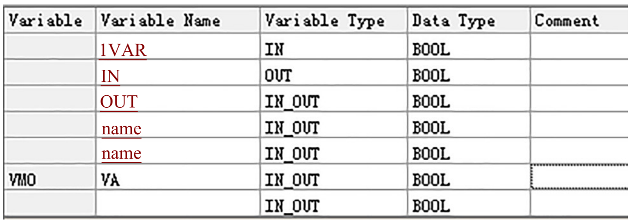


4) Syntax rule for naming variables

- A variable name consists of letters, digits, and underlines. The first character cannot be a digit.
- A keyword, for example, IN, OUT, LD, and ADD, cannot be used as a variable name.
- A symbol name consists of a maximum of eight characters (or four Chinese characters).

5) Indicating incorrect variable names

- AutoShop marks in red and underlines incorrect variable names.



Variable	Variable Name	Variable Type	Data Type	Comment
	1VAR	IN	BOOL	
	IN	OUT	BOOL	
	OUT	IN_OUT	BOOL	
	name	IN_OUT	BOOL	
	name	IN_OUT	BOOL	
VMO	VA	IN_OUT	BOOL	
		IN_OUT	BOOL	

- Red text indicates invalid syntax.
- A variable name must not start with a digit.
- A variable name must not be a keyword.
- A variable name must not be reused.

6) Variable type

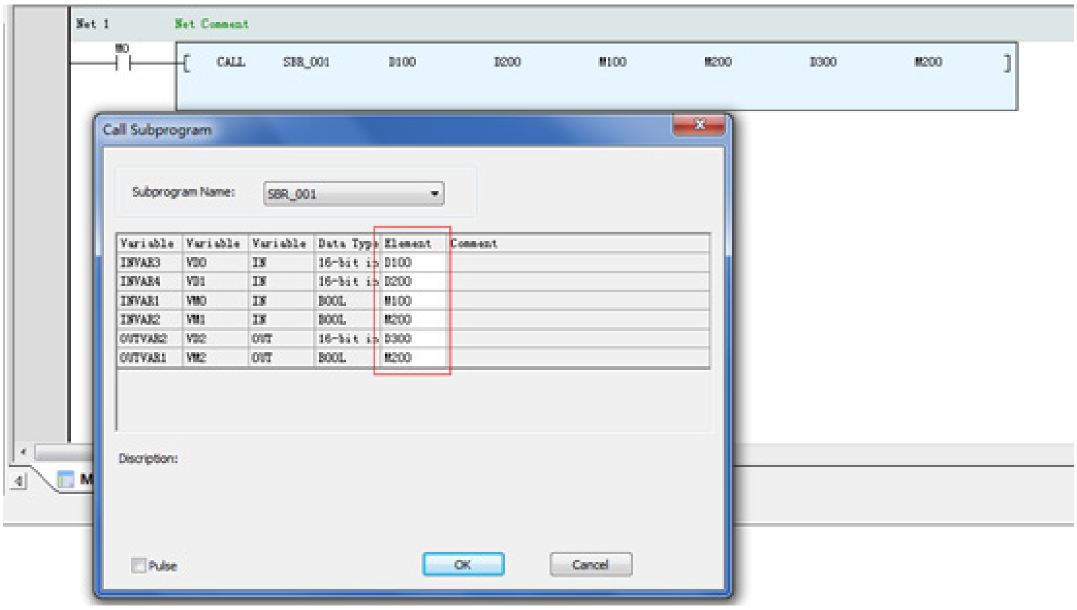
Variable Type	Description
IN	To transmit a value to a subprogram, create a variable in the variable table of the subprogram and specify the variable type as IN.
OUT	To return a value established in a subprogram to the calling program, create a variable in the variable table of the subprogram and specify the variable type as OUT.
IN_OUT	The value specifying the parameter position is transmitted to a subprogram, and the result is returned from the subprogram to the same address.

- A maximum of 16 IN, OUT, and IN_OUT parameters are defined.

7) Data type

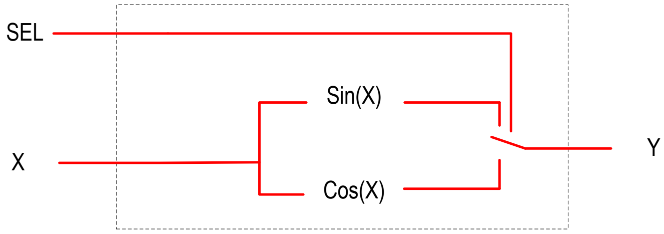
BOOL data, 16-bit integers, 32-bit integers, and floating-point numbers are supported.

12.4.3 Calling a Subprogram with Parameter

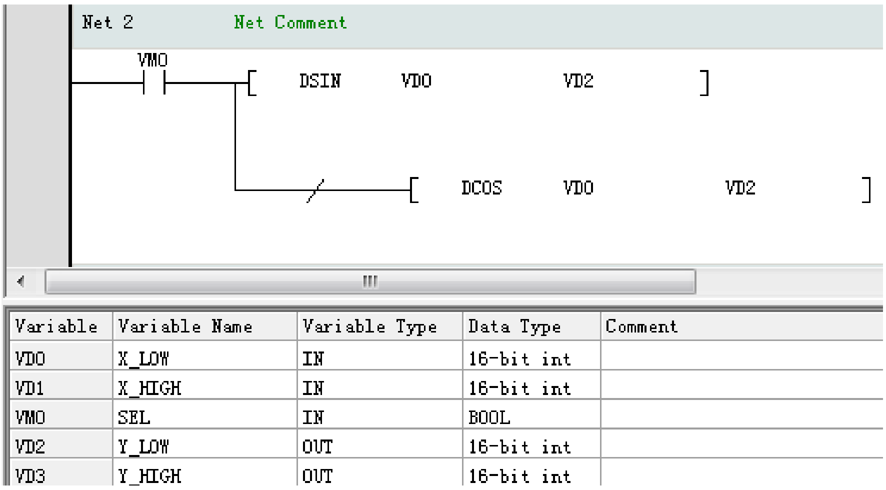


Enter **CALL** and press **Space**. The **Call Subprogram** window is displayed. Select the desired subprogram with parameters. As parameter names have been entered, the subprogram alias is displayed. Enter parameters to be input and output.

- Example: During an arithmetical operation, if SEL = ON, Sin(X) is output; otherwise, Cos(X) is output.

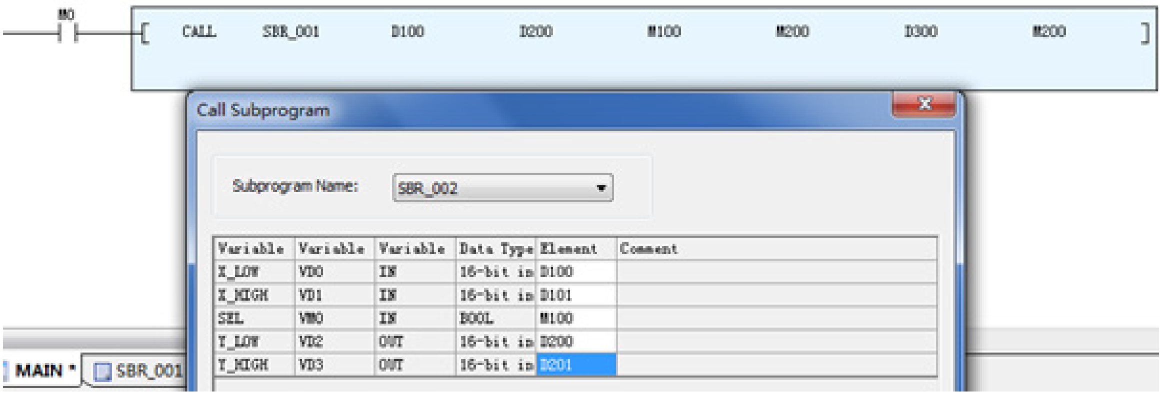


X and Y are 32-bit floating-point numbers. SEL is a bit variable. As a subprogram with parameters supports transmission of only 16-bit data, two word elements need to be input. The case of data output is similar.



Variable	Variable Name	Variable Type	Data Type	Comment
VD0	X_LOW	IN	16-bit int	
VD1	X_HIGH	IN	16-bit int	
VM0	SEL	IN	BOOL	
VD2	Y_LOW	OUT	16-bit int	
VD3	Y_HIGH	OUT	16-bit int	

The following shows the **Call Subprogram** window.



Variable	Variable	Variable	Data Type	Element	Comment
X_LOW	VD0	IN	16-bit in	D100	
X_HIGH	VD1	IN	16-bit in	D101	
SEL	VM0	IN	BOOL	M100	
Y_LOW	VD2	OUT	16-bit in	D200	
Y_HIGH	VD3	OUT	16-bit in	D201	

12.4.4 Precautions

- As re-access is not allowed, a subprogram cannot be called recursively or called within an interrupt.
- High-speed input and output instructions such as OUT C (235-255), PLSY, DPLSY, PLSR, DPLSR, DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV, and SPD instructions cannot be used.

12.5 Application of an Encrypted Subprogram with Parameters

When a subprogram with parameters is encrypted, it is called an encrypted subprogram with parameters.

12.6 Application of an Interrupt Subprogram

For details about application of an interrupt subprogram to an H3U standard model, see "5.1.2 Input Interrupts" on page 339.

For details about application of an interrupt subprogram to an H3U-PM motion control model, see "5.3.2 Input Interrupts" on page 355.

12.7 Application of a Motion Control Subprogram

For details about application, see "7.5 Similarities and Differences Between MC Subprograms and G-code Subprograms" on page 474

Memo No. ____________

Date / /

Appendix

Appendix

Appendix A Allocation of Soft Elements SM, SD, D8000, and M8000

A.1 SM Flag and SD Register

The following table lists special element registers.

H3U	SM Range	SD Range
0 to 299	For motion control (H3U-PM model)	For motion control (H3U-PM model)
300 to 363	Reserved	CANopen data sending and receiving area
364 to 399	Reserved	For the Ethernet
400 to 599	For manipulator instructions	For manipulator instructions
600 to 699	Reserved (occupied by customized 2-axis Delta manipulator)	Reserved (occupied by customized 2-axis Delta manipulator)
700 to 1023	Reserved	Reserved

A.2 Special Soft Element Register Range

Special element registers range from SM0 to SM299, as shown in the following table.

X-axis	Y-axis	Z-axis	Attribute
SM0 to 11	SM100 to 111	SM200 to 211	Reserved
SM12	SM112	SM212	Flag of DRVZ homing direction
SM13	SM113	SM213	Specified flag of the ZRN signal, which is DOG signal by default, or PG signal after setting
SM14 to 16	SM114 to 116	SM214 to 216	Reserved
SM17	SM117	SM217	S-curve acceleration/deceleration enabling flag
SM18	SM118	SM218	Axis origin return disabling
SM19	SM119	SM219	Reserved
SM20	SM120 (reserved)	SM220 (reserved)	Flag of enabling continuous interpolation
SM21 to 69	SM121 to 169	SM221 to 269	Reserved
SM70	SM170	SM270	Axis trigger mode selection for an electronic cam OFF: Software trigger; ON: Hardware trigger
SM71	SM171	SM271	Axis input source selection for an electronic cam OFF: Internal virtualization; ON: External input
SM72	SM172	SM272	Synchronization of x-, y-, and z-axes for an electronic cam OFF: Disabled; ON: Enabled
SM73	SM173	SM273	Cyclic execution of an electronic cam OFF: Disabled; ON: Enabled
SM74	SM174	SM274	External hardware stop OFF: Disabled; ON: Enabled
SM75	SM175	SM275	Startup latency enabling for an electronic cam OFF: Disabled; ON: Enabled
SM76	SM176	SM276	Left limit enabling OFF: Disabled; ON: Enabled

X-axis	Y-axis	Z-axis	Attribute
SM77	SM177	SM277	Right limit enabling OFF: Disabled; ON: Enabled
SM78	SM178	SM278	Electronic cam enabling OFF: Disabled; ON: Enabled
SM79	SM179	SM279	Cam cycle end flag OFF: Unfinished; ON: Finished
SM80	SM180	SM280	Electronic cam/gear stop flag OFF: Unfinished; ON: Finished
SM81	SM181	SM281	Stop mode selection OFF: Stop after the current cycle ON: Immediate stop
SM82	SM182 (reserved)	SM282 (reserved)	Electronic cam modification complete flag
SM83	SM183	SM283	Key point modification mode selection for an electronic cam OFF: Effective upon restart ON: Effective during the next cam cycle
SM84 to 88	SM184 to 188	SM284 to 288	Reserved
SM89	SM189	SM289	Initialization complete flag OFF: Initialization started ON: Initialization completed
SM90	SM190 (reserved)	SM290 (reserved)	Motion control subprogram MCX enabling flag
SM91	SM191 (reserved)	SM291 (reserved)	Motion control subprogram MCX execution complete flag
SM92 to 99	SM192 to 199	SM292 to 299	Reserved

Special element registers range from SD0 to SD299, as shown in the following table.

X-axis	Y-axis	Z-axis	Attribute
SD0	SD100	SD200	Reserved
SD1	SD101	SD201	Reserved
SD2	SD102	SD202	Reserved
SD3	SD103	SD203	Reserved
SD4	SD104	SD204	Reserved
SD5	SD105	SD205	Reserved
SD6 and SD7	SD106 and SD107	SD206 and SD207	Number of pulses required when the motor rotates a circle (A)
SD8 and SD9	SD108 and SD109	SD208 and SD209	Movement distance when the motor finishes a circle (B), in the unit of um or 0.001° (mechanical unit)
SD10 and SD11	SD110 and SD111	SD210 and SD211	Maximum speed (Vmax)
SD12 and SD13	SD112 and SD113	SD212 and SD213	Base speed (starting speed) (Vbias)
SD16 and SD17	SD116 and SD117	SD216 and SD217	Homing speed (VRT)
SD18 and SD19	SD118 and SD119	SD218 and SD219	Homing creep speed (VCR)
SD20	SD120	SD220	Acceleration time (Vacc)
SD21	SD121	SD221	Deceleration time (Vdec)
SD22	SD122	SD222	Number of PG signals (N)
SD23	SD123	SD223	Number of pulses for homing (P), pulse offset in case of a DOG signal
SD24 and SD25	SD124 and SD125	SD224 and SD225	Home position (HP)
SD26 and SD27	SD126 and SD127	SD226 and SD227	Electrical origin position
SD28 and SD29	SD128 and SD129	SD228 and SD229	Target position I (P [I])

X-axis	Y-axis	Z-axis	Attribute
SD30 and SD31	SD130 and SD131	SD230 and SD231	Running speed I (V [I])
SD32 and SD33	SD132 and SD133	SD232 and SD233	Target position II (P [II])
SD34 and SD35	SD134 and SD135	SD234 and SD235	Running speed II (V [II])
SD36, SD37	SD136, SD137	SD236, SD237	Current position (CP [PLS])
SD38 and SD39	SD138 and SD139	SD238 and SD239	Current speed (CS [PPS])
SD40 and SD41	SD140 and SD141	SD240 and SD241	Current position (CP [mechanical and floating-point])
SD42 and SD43	SD142 and SD143	SD242 and SD243	Current speed (CP [mechanical and floating-point])
SD44	SD144	SD244	Electronic gear ratio numerator
SD45	SD145	SD245	Electronic gear ratio denominator
SD46 and SD47	SD146 and SD147	SD246 and SD247	Current input frequency
SD48 and SD49	SD148 and SD149	SD248 and SD249	Number of master axis pulses
SD50 and SD51	SD150 and SD151	SD250 and SD251	Axis offset compensation value (DRV, LIN, and INTR)
SD52 and SD53	SD152 and SD153	SD252 and SD253	Axis center coordinate offset compensation value (CW and CCW)
SD54 and SD55	SD154 and SD155 (reserved)	SD254 and SD255 (reserved)	Axis arc radius offset compensation value (CW and CCW)
SD56 to 59	SD156 to 159	SD256 to 259	Reserved
SD60	SD160	SD260	Setting of high-speed pulse input and count
SD61	SD161	SD261	High-speed pulse output setting
SD62	SD162	SD262	Display of the status of special PM input point
SD63	SD163	SD263	Display of the status of special PM output point
SD64 to 69	SD164 to 169	SD264 to 269	Reserved
SD70	SD170	SD270	Electronic cam axis selection table: 0: Cam and hand gear disabled by default; 10: Hand gear; 11: Cam 1; 12: Cam 2; 13: Cam 3
SD71	SD171	SD271	Setting of electronic cam input axis numbers
SD72	SD172	SD272	Times of non-cyclic cam execution
SD73	SD173	SD273	Reserved
SD74 and SD75	SD174 and SD175	SD274 and SD275	Lower limit of cam synchronization position
SD76 and SD77	SD176 and SD177	SD276 and SD277	Upper limit of cam synchronization position
SD78 and SD79	SD178 and SD179	SD278 and SD279	Number of pulses (startup latency)
SD80	SD180	SD280	Input pole selection
SD81 and SD82	SD181 and SD182	SD281 and SD282	Finished cam cycles
SD83 to 89	SD183 to 189	SD283 to 289	Reserved
SD90	SD190 (reserved)	SD290 (reserved)	Motion control subprogram MCX marker setting register
SD91 to 99	SD191 to 199	SD291 to 299	Reserved

A.3 M8000 Flag and D8000 Register

The following table lists special element registers.

M Element	Description	D Element	Description
System State			
M8000	ON during operation of the user program	D8000	Monitor timer for user program operation
M8001	Inverted M8000 state	D8001	Board program version, 24xxx
M8002	ON in the first operation cycle of the user program	D8002	Maximum capacity of the user program: 4 KB, 8 KB, or 16 KB
M8003	Inverted M8002 state	D8003	Maximum capacity of the user data: 4 KB, 8 KB, or 16 KB
M8004	Valid if any of M8060 to M8067 (except M8062) is ON	D8004	Incorrect BCD values of M8060 to M8067 (normal: 0)
M8005	Action upon battery undervoltage (not for an XP model)	D8005	Current battery voltage BCD value (not for an XP model)
M8006	Action upon low battery voltage (latch)	D8006	Detected battery undervoltage value, 2.6 V by default
M8007	Action after 5 ms of AC de-energization, which keeps ON for a scan cycle. If the de-energization time value is not greater than the D8008 value, the program continues.	D8007	M8007 action storage count, which should be cleared upon de-energization
M8008	Action after 5 ms of AC de-energization, which is reset when the de-energization time value reaches the D8008 value	D8008	AC de-energization detection time, 20 ms by default
M8009	Reserved	D8009	Reserved
System clock			
M8010	Reserved	D8010	Current scan time, starting from the 0-step (0.1 ms)
M8011	Free-run clock with a cycle of 10 ms	D8011	Minimum scan time (0.1 ms)
M8012	Free-run clock with a cycle of 100 ms	D8012	Maximum scan time (0.1 ms)
M8013	Free-run clock with a cycle of 1s	D8013	Clock second (0 to 59)
M8014	Free-run clock with a cycle of 1 minute	D8014	Real-time clock (RTC) minute (0 to 59)
M8015	Clock stop and presetting	D8015	RTC hour (0 to 23)
M8016	Clock reading stop	D8016	RTC day (1 to 31)
M8017	±30s calibration	D8017	RTC month (1 to 12)
M8018	Reserved	D8018	RTC calendar year (2000 to 2099)
M8019	RTC error	D8019	RTC week
Application instruction expansion function 1			
M8020	Zero flag	D8020	X000-X007 general input filter constants 1 to 60 (in the unit of ms, corresponding to 1 to 60 ms, 10 ms by default)
M8021	Borrow flag	D8021	X000-X007 high-speed input filter constants 1 to 100 (in the unit of 0.25 us, corresponding to 0.25 to 25 us)
M8022	Carry flag	D8022	FPGA version (lower 16 bits)
M8023	Reserved	D8023	FPGA version (upper 16 bits)
M8024	Direction of the BMOV instruction	D8024	Reserved
M8025	Summation flag of the MEAN instruction	D8025	Reserved
M8026	RAMP instruction mode	D8026	Reserved
M8027	PR mode	D8027	Reserved
M8028	Customized function enabling flag of the ROTC instruction	D8028	Z0 register

M Element	Description			D Element	Description
M8029	Multi-cycle instruction execution complete flag			D8029	V0 register
System mode					
M8030	The system disables the low battery alarm when M8030 is ON.			D8030	Reserved
M8031	The system clears all volatile memories when M8031 is ON.			D8031	Reserved
M8032	The system clears all non-volatile memories when M8032 is ON.			D8032	Reserved
M8033	When M8033 is ON, all soft elements remain unchanged when the device is stopped.			D8033	Reserved
M8034	When M8034 is ON, all outputs are OFF.			D8034	Reserved
M8035	Command for forced running 1			D8035	Reserved
M8036	Command for forced running 2			D8036	Reserved
M8037	Command for forced stop			D8037	Reserved
M8038	Reserved			D8038	Reserved
M8039	Constant scan control			D8039	Constant scan time, 0 by default, in the unit of ms
Step ladder					
M8040	Transfer disabling			D8040	Store the smallest action address numbers of S0 to S899 and S1000 to S4095 in D8040. Store the greatest address numbers in D8047.
M8041	Transfer start			D8041	
M8042	Pulse output corresponding to startup input			D8042	
M8043	Homing end flag			D8043	
M8044	Mechanical origin action detected			D8044	
M8045	Output reset disabling			D8045	
M8046	After M8047 acts, if any of S0 to S899 and S1000 to S4095 is ON, M8046 is ON.			D8046	
M8047	Valid STL monitoring [D8040 to D8047 valid]			D8047	
M8048	When M8049 is ON, if any of S900 to S999 is valid, M8048 is valid.			D8048	Reserved
M8049	Valid signal alarm [D8049 valid]			D8049	Store the smallest S900-S999 alarm address numbers.
Disable interrupt					
M8050	Driver I00□ (X0 pulse) interrupt disabling			D8050	Reserved
M8051	Driver I10□ (X1 pulse) interrupt disabling			D8051	Reserved
M8052	Driver I20□ (X2 pulse) interrupt disabling			D8052	Reserved
M8053	Driver I30□ (X3 pulse) interrupt disabling			D8053	Reserved
M8054	Driver I40□ (X4 pulse) interrupt disabling			D8054	Reserved
M8055	Driver I50□ (X5 pulse) interrupt disabling			D8055	Reserved
M8056	Driver I6□□ (timing) interrupt disabling			D8056	Reserved
M8057	Driver I7□□ (timing) interrupt disabling			D8057	Reserved
M8058	Driver I8□□ (timing) interrupt disabling			D8058	Reserved
M8059	Driver counter interrupt disabling			D8059	Reserved
System error detection					
Element	Name	Error indicator	Running		
M8060	I/O composition error []	OFF	RUN	D8060	I/O address number of I/O composition error
M8061	PLC hardware error	Blinking	STOP	D8061	PLC hardware error code number
M8062	PLC configuration error	OFF	RUN	D8062	PLC configuration error

M Element	Description			D Element	Description
M8063	PLC communication error	OFF	RUN	D8063	PLC communication error code
M8064	Parameter error	Blinking	STOP	D8064	Parameter error code
M8065	Syntax error	Blinking	STOP	D8065	Syntax error code
M8066	Circuit error	Blinking	STOP	D8066	Circuit error code
M8067	Operation error	OFF	RUN	D8067	Operation error code
M8068	Operation error latch	OFF	RUN	D8068	Step number of the latched program with an operation error
M8069	System error flag, set to ON			D8069	M8065-M8067 error step numbers
Parallel connection (1:1)					
M8070	Reserved			D8070	Duration of incorrect parallel connection, 500 ms by default
M8071	Reserved			D8071	Reserved
M8072	ON during parallel connection			D8072	Reserved
M8073	Incorrect M8070/M8071 setting for parallel connection			D8073	Reserved
High-speed input and counting					
M8074	Reserved			D8074	Reserved
M8075	Reserved			D8075	Reserved
M8076	Driver I56□ (X6 pulse) interrupt disabling			D8076	Reserved
M8077	Driver I57□ (X7 pulse) interrupt disabling			D8077	Reserved
M8078	Reserved			D8078	Reserved
M8079	Reserved			D8079	Reserved
M8080	X0 pulse capture enabling			D8080	Reserved
M8081	X1 pulse capture enabling			D8081	Reserved
M8082	X2 pulse capture enabling			D8082	Reserved
M8083	X3 pulse capture enabling			D8083	Reserved
M8084	X4 pulse capture enabling			D8084	Reserved
M8085	X5 pulse capture enabling			D8085	Reserved
M8086	X6 pulse capture enabling			D8086	Reserved
M8087	X7 pulse capture enabling			D8087	Reserved
M8088	Reserved			D8088	Reserved
M8089	Reserved			D8089	Reserved
M8090	Flag of X0 pulse captured			D8090	Reserved
M8091	Flag of X1 pulse captured			D8091	Reserved
M8092	Flag of X2 pulse captured			D8092	Reserved
M8093	Flag of X3 pulse captured			D8093	Reserved
M8094	Flag of X4 pulse captured			D8094	Reserved
M8095	Flag of X5 pulse captured			D8095	Reserved
M8096	Flag of X6 pulse captured			D8096	Reserved
M8097	Flag of X7 pulse captured			D8097	Reserved
M8098	Reserved			D8098	Reserved
M8099	High-speed ring counter started			D8099	[0 to 32,767] rising action ring counter (0.1 ms)
M8100	Reserved			D8100	Board program version, 24xxx
M8101	Reserved			D8101	Auxiliary board program version, 0xBxxx, in hexadecimal format

M Element	Description	D Element	Description
M8102	Reserved	D8102	Customized version of customized software Fxxx, in hexadecimal format
M8103	Reserved	D8103	Customized iterative version of customized software Fxxx, in hexadecimal format
M8104	Reserved	D8104	Major version of FPGA software
M8105	Reserved	D8105	Auxiliary version of FPGA software
M8106	Reserved	D8106	Reserved
M8107	Reserved	D8107	Reserved
M8108	Reserved	D8108	Reserved
M8109	Reserved	D8109	Reserved
COM0 communication link			
M8110	Reserved	D8110	Communication format, set on the UI, 0 by default
M8111	Reserved	D8111	Communication station number, set on the UI, 1 by default
M8112	Modbus - communication execution state	D8112	Download and HMI monitoring protocols - communication format setting
M8113	Modbus - communication error flag	D8113	Reserved
M8114	Reserved	D8114	Reserved
M8115	Reserved	D8115	Reserved
M8116	Reserved	D8116	Communication protocol, set on the UI, 0 by default
M8117	Reserved	D8117	Reserved
M8118	Reserved	D8118	Modbus - number of station with communication errors
M8119	Timeout criterion	D8119	Communication timeout criterion, set on the UI, 10 (100 ms) by default
COM1 communication link			
M8120	Reserved	D8120	Communication format, set on the UI, 0 by default
M8121	RS instruction - sending	D8121	Communication station number, set on the UI, 1 by default
M8122	Modbus - communication execution state RS instruction - flag of sending	D8122	Download and HMI monitoring protocols - communication format setting RS instruction - volume of residual data transmitted
M8123	Modbus - communication error flag RS instruction - flag of receipt	D8123	RS instruction - volume of received data
M8124	RS instruction - receiving	D8124	RS instruction - STX
M8125	Reserved	D8125	RS instruction - ETX
M8126	Reserved	D8126	Communication protocol, set on the UI, 0 by default
M8127	Reserved	D8127	PC link protocol - head data address required
M8128	Reserved	D8128	Modbus - number of station with communication errors PC link protocol - volume of sent data required
M8129	Timeout criterion	D8129	Communication timeout criterion, set on the UI, 10 (100 ms) by default
High-speed input instruction			
M8130	HSZ instruction platform control mode	D8130	Use of the HSZ high-speed comparison platform (number recorded)
M8131	Used with M8130	D8131	Use of HSZ and PLSY speed models (number recorded)
M8132	HSZ & PLSY speed modes	D8132	Use of HSZ and PLSY speed model frequency
M8133	Used with M8132	D8133	

M Element	Description	D Element	Description
M8134	Reserved	D8134	Use of HSZ and PLSY speed model comparison pulses
M8135	Reserved	D8135	
M8136 to M8159	Reserved	D8136 to D8159	Reserved
Application instruction expansion function 2			
M8160	SWAP function of (XCH)	D8160	Set proface screen flag (1)
M8161	ASC/ASCII/HEX/RS/CCD/LRC/CRC processing mode	D8161	Reserved
M8162	High-speed parallel connection (1:1) mode	D8162	Reserved
M8163	BINDA instruction output switchover flag	D8163	Reserved
M8164	Flag of enabling customized functions of a combinatorial balance	D8164	Reserved
M8165	Flag of descending sort enabling for the SORT2 instruction	D8165	Reserved
M8166	Reserved	D8166	Reserved
M8167	(HEY) HEX data processing function	D8167	Reserved
M8168	(SMOV) HEX data processing function	D8168	Reserved
M8169	Reserved	D8169	Reserved
Serial port communication link			
M8170	N:N extensible protocol data transmission slave station 8 error	D8170	Reserved
M8171	N:N extensible protocol data transmission slave station 9 error	D8171	Reserved
M8172	N:N extensible protocol data transmission slave station 10 error	D8172	Reserved
M8173	N:N extensible protocol data transmission slave station 11 error	D8173	N:N and N:N extensible communication station number state
M8174	N:N extensible protocol data transmission slave station 12 error	D8174	N:N and N:N extensible communication substation state
M8175	N:N extensible protocol data transmission slave station 13 error	D8175	N:N and N:N extensible communication refresh range state
M8176	N:N extensible protocol data transmission slave station 14 error	D8176	N:N and N:N extensible communication station number setting
M8177	N:N extensible protocol data transmission slave station 15 error	D8177	Setting of number of N:N and N:N extensible communication substations
M8178	Reserved	D8178	N:N and N:N extensible communication refresh range setting
M8179	Reserved	D8179	Setting of number of N:N and N:N extensible communication retries
M8180	Reserved	D8180	N:N and N:N extensible communication timeout setting
M8181	Reserved	D8181	Reserved
M8182	Reserved	D8182	Z1 register
M8183	N:N data transmission master station error	D8183	V1 register
M8184	N:N data transmission slave station 1 error	D8184	Z2 register
M8185	N:N data transmission slave station 2 error	D8185	V2 register
M8186	N:N data transmission slave station 3 error	D8186	Z3 register
M8187	N:N data transmission slave station 4 error	D8187	V3 register
M8188	N:N data transmission slave station 5 error	D8188	Z4 register
M8189	N:N data transmission slave station 6 error	D8189	V4 register

M Element	Description	D Element	Description
M8190	N:N data transmission slave station 7 error	D8190	Z5 register
M8191	N:N data - transmitting	D8191	V5 register
M8192	N:N protocol compatible with Mitsubishi communication format	D8192	Z6 register
M8193	Reserved	D8193	V6 register
M8194	Reserved	D8194	Z7 register
M8195	C251 multiplication control	D8195	V7 register
M8196	C252 multiplication control	D8196	Reserved
M8197	C253 multiplication control	D8197	Reserved
M8198	C254 multiplication control	D8198	CANlink remote device identifier (10224 for H2U and 10226 for H1U)
M8199	C255 multiplication control	D8199	Reserved
Counter increment/decrement or control		Serial port communication link	
M8200	C200 control	D8200	Auxiliary board program version, 0xBxxx, in hexadecimal format
M8201	C201 control	D8201	Current N:N and N:N extensible communication connection scan time
M8202	C202 control	D8202	Maximum N:N and N:N extensible communication connection time
M8203	C203 control	D8203	Number of N:N master station communication errors
M8204	C204 control	D8204	Number of N:N slave station 1 communication errors
M8205	C205 control	D8205	Number of N:N slave station 2 communication errors
M8206	C206 control	D8206	Number of N:N slave station 3 communication errors
M8207	C207 control	D8207	Number of N:N slave station 4 communication errors
M8208	C208 control	D8208	Number of N:N slave station 5 communication errors
M8209	C209 control	D8209	Number of N:N slave station 6 communication errors
M8210	C210 control	D8210	Number of N:N slave station 7 communication errors
M8211	C211 control	D8211	N:N master station communication error code Number of N:N extensible protocol slave station 8 communication errors
M8212	C212 control	D8212	N:N slave station 1 communication error code Number of N:N extensible protocol slave station 9 communication errors
M8213	C213 control	D8213	N:N slave station 2 communication error code Number of N:N extensible protocol slave station 10 communication errors
M8214	C214 control	D8214	N:N slave station 3 communication error code Number of N:N extensible protocol slave station 11 communication errors
M8215	C215 control	D8215	N:N slave station 4 communication error code Number of N:N extensible protocol slave station 12 communication errors
M8216	C216 control	D8216	N:N slave station 5 communication error code Number of N:N extensible protocol slave station 13 communication errors
M8217	C217 control	D8217	N:N slave station 6 communication error code Number of N:N extensible protocol slave station 14 communication errors
M8218	C218 control	D8218	N:N slave station 7 communication error code Number of N:N extensible protocol slave station 15 communication errors

M Element	Description	D Element	Description
M8219	C219 control	D8219	Reserved
M8220	C220 control	D8220	Reserved
M8221	C221 control	D8221	Reserved
M8222	C222 control	D8222	Reserved
M8223	C223 control	D8223	Reserved
M8224	C224 control	D8224	Reserved
M8225	C225 control	D8225	Reserved
M8226	C226 control	D8226	Reserved
M8227	C227 control	D8227	Reserved
M8228	C228 control	D8228	Reserved
M8229	C229 control	D8229	Reserved
M8230	C230 control	D8230	Reserved
M8231	C231 control	D8231	Reserved
M8232	C232 control	D8232	Reserved
M8233	C233 control	D8233	Reserved
M8234	C234 control	D8234	Reserved
M8235	C235 control	D8235	Reserved
M8236	C236 control	D8236	Reserved
M8237	C237 control	D8237	Reserved
M8238	C238 control	D8238	Reserved
M8239	C239 control	D8239	Reserved
M8240	C240 control	D8240	CAN occupied and unavailable
M8241	C241 control	D8241	CAN occupied and unavailable
M8242	C242 control	D8242	CAN occupied and unavailable
M8243	C243 control	D8243	CAN occupied and unavailable
M8244	C244 control	D8244	CAN occupied and unavailable
M8245	C245 control	D8245	CAN occupied and unavailable
M8246	C246 state	D8246	CAN occupied and unavailable
M8247	C247 state	D8247	CAN occupied and unavailable
M8248	C248 state	D8248	CAN occupied and unavailable
M8249	C249 state	D8249	CAN occupied and unavailable
M8250	C250 state	D8250	CAN occupied and unavailable
M8251	C251 state	D8251	CAN occupied and unavailable
M8252	C252 state	D8252	CAN occupied and unavailable
M8253	C253 state	D8253	CAN occupied and unavailable
M8254	C254 state	D8254	CAN occupied and unavailable
M8255	C255 state	D8255	CAN occupied and unavailable
M8256	Reserved	D8256	Reserved
M8257	Reserved	D8257	Reserved
M8258	Reserved	D8258	Reserved
M8259	Reserved	D8259	Reserved
COM2 communication link			
M8260 to M8269	Reserved	D8260 to D8269	Reserved

M Element	Description	D Element	Description
COM3 communication link			
M8270 to M8279	Reserved	D8270 to D8279	Reserved
CAN communication			
M8280	Protocol switchover flag 0: CANlink3.0 protocol 1: CANopen protocol	D8280	Valid protocol display 300: CANlink3.0 (300 by default) 100: CANopen
M8281	Reserved	D8281	Reserved
M8282	Reserved	D8282	CANlink heartbeat
M8283	Valid address for online CAN monitoring	D8283	Head address for CAN online monitoring
M8284	0: CAN address set by the DIP switch and displayed by D8284 1: CAN address set by D8284	D8284	CAN address setting/display
M8285	0: Baud rate set by the DIP switch and displayed by D8285 1: Baud rate set by D8286	D8285	Baud rate display
M8286	Reserved	D8286	10: 10 Kbps 20: 20 Kbps 50: 50 Kbps 100: 100 Kbps 125: 125 Kbps 250: 250 Kbps 500: 500 Kbps 800: 800 Kbps 1000: 1 Mbps
M8287	Reserved	D8287	Number of the station with CANopen configuration errors
M8288	Reserved	D8288	CANopen configuration error number
M8289	Reserved	D8289	CAN bus error
M8290	Reserved	D8290	CAN receipt error
M8291	Reserved	D8291	Reserved
M8292	Reserved	D8292	Reserved
M8293	Reserved	D8293	Reserved
M8294	Reserved	D8294	Reserved
M8295	Reserved	D8295	Reserved
M8296	Incorrect address	D8296	Reserved
M8297 to M8306	Reserved	D8297 to D8306	Reserved
M8307	Reserved	D8307	CANlink3.0 synchronous write error
M8308	Reserved	D8308	CANlink3.0 configuration error
M8309	Reserved	D8309	Reserved
Application instruction expansion function 3			
M8310	Manipulator customized - function enabling flag	D8310	(RND) Random lower bit
M8311	Manipulator customized - single-cycle operation enabling flag	D8311	(RND) Random upper bit
M8312	Manipulator customized - suspension flag	D8312	Manipulator customized - function switchover
M8313	Manipulator customized - flag of completion of all actions	D8313	Manipulator customized - display of IDs of active programs

M Element	Description	D Element	Description
M8314	Manipulator customized -	D8314	Manipulator customized - display of action numbers
M8315	Manipulator customized -	D8315	Manipulator customized - error display
M8316	Manipulator customized -	D8316	Manipulator customized - total number of actions
M8317	Manipulator customized -	D8317	Manipulator customized -
M8318	Manipulator customized -	D8318	Manipulator customized -
M8319	Manipulator customized -	D8319	Manipulator customized -
M8320	Flag of enabling values sharing the same matrix comparison	D8320	Manipulator customized -
M8321	Matrix search end flag	D8321	Manipulator customized -
M8322	Matrix search start flag	D8322	Reserved
M8323	Flag of matrix found	D8323	Reserved
M8324	Matrix pointer error flag	D8324	Reserved
M8325	Matrix pointer + 1 enabling flag	D8325	Reserved
M8326	Matrix pointer clearance flag	D8326	Reserved
M8327	Matrix shift output carry flag	D8327	Reserved
M8328	Matrix shift input placeholder flag	D8328	Reserved
M8329	Matrix shift direction flag	D8329	Reserved
M8330	Reserved	D8330	Reserved
M8331	(MBC) Flag of counting "0" or "1" matrix bits	D8331	Reserved
M8332	(MBC) Set to ON when the counting result is 0	D8332	Reserved
M8333	Flag of all BKCMP instruction matrix comparison results equal to 1	D8333	Reserved
M8334	Reserved	D8334	Reserved
M8335	(DUTY) Timing pulse signal output	D8335	Scan count corresponding to M8335
M8336	(DUTY) Timing pulse signal output	D8336	Scan count corresponding to M8336
M8337	(DUTY) Timing pulse signal output	D8337	Scan count corresponding to M8337
M8338	(DUTY) Timing pulse signal output	D8338	Scan count corresponding to M8338
M8339	(DUTY) Timing pulse signal output	D8339	Scan count corresponding to M8339
High-speed output and positioning			
Y0 pulse output port			
M8340	Y0 monitoring during pulse output	D8340	Y0 current value register (PLS) (32-bit)
M8341	Y0 active output label for the DSZR/ZRN and other clearing signals	D8341	
M8342	Y0 designation of DSZR/ZRN direction	D8342	Y0 maximum speed (Hz) (32-bit) [default value: 200,000]
M8343	Y0 forward rotation stroke end	D8343	
M8344	Y0 reverse rotation stroke end	D8344	Y0 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8345	Y0 near point signal logical inversion	D8345	
M8346	Y0 homing signal logical inversion	D8346	Y0 creep speed (Hz) [default value: 2000]
M8347	Y0 S-curve acceleration/deceleration enabling	D8347	Y0 base speed (Hz) [default value: 500]
M8348	Y0 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8348	Y0 acceleration time (ms) [default value: 100]
M8349	Y0 pulse output stop flag	D8349	Y0 deceleration time (ms) [default value: 100]
M8350	Y0 [positioning instruction] separate setting of the acceleration/deceleration time and modification to the pulse supported	D8350	Y0 number of clearing soft element
M8351	Y0 output initialization flag	D8351	Reserved
M8352	Y0 output complete interrupt enabling	D8352	Reserved
M8353	Reserved	D8353	Reserved

M Element	Description	D Element	Description
M8354	Y0 flag of abnormal end of high-speed output	D8354	Reserved
M8355	Y0 PLSV2 instruction accelerating flag	D8355	Reserved
M8356	Y0 PLSV2 instruction decelerating flag	D8356	Reserved
M8357	Reserved	D8357	Reserved
M8358	Reserved	D8358	Reserved
M8359	Reserved	D8359	Reserved
Y1 pulse output port			
M8360	Y1 monitoring during pulse output	D8360	Y1 current value register (PLS) (32-bit)
M8361	Y1 active output label for the DSZR/ZRN and other clearing signals	D8361	
M8362	Y1 designation of DSZR/ZRN direction	D8362	Y1 maximum speed (Hz) (32-bit) [default value: 200,000]
M8363	Y1 forward rotation stroke end	D8363	
M8364	Y1 reverse rotation stroke end	D8364	Y1 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8365	Y1 near point signal logical inversion	D8365	
M8366	Y1 homing signal logical inversion	D8366	Y1 creep speed (Hz) [default value: 2000]
M8367	Y1 S-curve acceleration/deceleration enabling	D8367	Y1 base speed (Hz) [default value: 500]
M8368	Y1 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8368	Y1 acceleration time (ms) [default value: 100]
M8369	Y1 pulse output stop flag	D8369	Y1 deceleration time (ms) [default value: 100]
M8370	Y1 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8370	Y1 number of clearing soft element
M8371	Y1 output initialization flag	D8371	Reserved
M8372	Y1 output complete interrupt enabling	D8372	Reserved
M8373	Reserved	D8373	Reserved
M8374	Y1 flag of abnormal end of high-speed output	D8374	Reserved
M8375	Y1 PLSV2 instruction accelerating flag	D8375	Reserved
M8376	Y1 PLSV2 instruction decelerating flag	D8376	Reserved
M8377	Reserved	D8377	Reserved
M8378	Reserved	D8378	Reserved
M8379	Reserved	D8379	Reserved
Y2 pulse output port			
M8380	Y2 monitoring during pulse output	D8380	Y2 current value register (PLS) (32-bit)
M8381	Y2 active output label for the DSZR/ZRN and other clearing signals	D8381	
M8382	Y2 designation of DSZR/ZRN direction	D8382	Y2 maximum speed (Hz) (32-bit) [default value: 200,000]
M8383	Y2 forward rotation stroke end	D8383	
M8384	Y2 reverse rotation stroke end	D8384	Y2 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8385	Y2 near point signal logical inversion	D8385	
M8386	Y2 homing signal logical inversion	D8386	Y2 creep speed (Hz) [default value: 2000]
M8387	Y2 S-curve acceleration/deceleration enabling	D8387	Y2 base speed (Hz) [default value: 500]
M8388	Y2 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8388	Y2 acceleration time (ms) [default value: 100]
M8389	Y2 pulse output stop flag	D8389	Y2 deceleration time (ms) [default value: 100]
M8390	Y2 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8390	Y2 number of clearing soft element
M8391	Y2 output initialization flag	D8391	Reserved
M8392	Y2 output complete interrupt enabling	D8392	Reserved
M8393	Reserved	D8393	Reserved

M Element	Description	D Element	Description
M8394	Y2 flag of abnormal end of high-speed output	D8394	Reserved
M8395	Y2 PLSV2 instruction accelerating flag	D8395	Reserved
M8396	Y2 PLSV2 instruction decelerating flag	D8396	Reserved
M8397	Reserved	D8397	Reserved
M8398	Reserved	D8398	Reserved
M8399	Reserved	D8399	Reserved
Y3 pulse output port			
M8400	Y3 monitoring during pulse output	D8400	Y3 current value register (PLS) (32-bit)
M8401	Y3 active output label for the DSZR/ZRN and other clearing signals	D8401	
M8402	Y3 designation of DSZR/ZRN direction	D8402	Y3 maximum speed (Hz) (32-bit) [default value: 200,000]
M8403	Y3 forward rotation stroke end	D8403	
M8404	Y3 reverse rotation stroke end	D8404	Y3 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8405	Y3 near point signal logical inversion	D8405	
M8406	Y3 homing signal logical inversion	D8406	Y3 creep speed (Hz) [default value: 2000]
M8407	Y3 S-curve acceleration/deceleration enabling	D8407	Y3 base speed (Hz) [default value: 500]
M8408	Y3 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8408	Y3 acceleration time (ms) [default value: 100]
M8409	Y3 pulse output stop flag	D8409	Y3 deceleration time (ms) [default value: 100]
M8410	Y3 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8410	Y3 number of clearing soft element
M8411	Y3 output initialization flag	D8411	Reserved
M8412	Y3 output complete interrupt enabling	D8412	Reserved
M8413	Reserved	D8413	Reserved
M8414	Y3 flag of abnormal end of high-speed output	D8414	Reserved
M8415	Y3 PLSV2 instruction accelerating flag	D8415	Reserved
M8416	Y3 PLSV2 instruction decelerating flag	D8416	Reserved
M8417	Reserved	D8417	Reserved
M8418	Reserved	D8418	Reserved
M8419	Reserved	D8419	Reserved
Y4 pulse output port			
M8420	Y4 monitoring during pulse output	D8420	Y4 current value register (PLS) (32-bit)
M8421	Y4 active output label for the DSZR/ZRN and other clearing signals	D8421	
M8422	Y4 designation of DSZR/ZRN direction	D8422	Y4 maximum speed (Hz) (32-bit) [default value: 200,000]
M8423	Y4 forward rotation stroke end	D8423	
M8424	Y4 reverse rotation stroke end	D8424	Y4 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8425	Y4 near point signal logical inversion	D8425	
M8426	Y4 homing signal logical inversion	D8426	Y4 creep speed (Hz) [default value: 2000]
M8427	Y4 S-curve acceleration/deceleration enabling	D8427	Y4 base speed (Hz) [default value: 500]
M8428	Y4 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8428	Y4 acceleration time (ms) [default value: 100]
M8429	Y4 pulse output stop flag	D8429	Y4 deceleration time (ms) [default value: 100]
M8430	Y4 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8430	Y4 number of clearing soft element
M8431	Y4 output initialization flag	D8431	Reserved
M8432	Y4 output complete interrupt enabling	D8432	Reserved
M8433	Reserved	D8433	Reserved

M Element	Description	D Element	Description
M8434	Y4 flag of abnormal end of high-speed output	D8434	Reserved
M8435	Y4 PLSV2 instruction accelerating flag	D8435	Reserved
M8436	Y4 PLSV2 instruction decelerating flag	D8436	Reserved
M8437	Reserved	D8437	Reserved
M8438	Reserved	D8438	Reserved
M8439	Reserved	D8439	Reserved
Y5 pulse output port (reserved)			
M8440	Y5 monitoring during pulse output	D8440	Y5 current value register (PLS) (32-bit)
M8441	Y5 active output label for the DSZR/ZRN and other clearing signals	D8441	
M8442	Y5 designation of DSZR/ZRN direction	D8442	Y5 maximum speed (Hz) (32-bit) [default value: 200,000]
M8443	Y5 forward rotation stroke end	D8443	
M8444	Y5 reverse rotation stroke end	D8444	Y5 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8445	Y5 near point signal logical inversion	D8445	
M8446	Y5 homing signal logical inversion	D8446	Y5 creep speed (Hz) [default value: 2000]
M8447	Y5 S-curve acceleration/deceleration enabling	D8447	Y5 base speed (Hz) [default value: 500]
M8448	Y5 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8448	Y5 acceleration time (ms) [default value: 100]
M8449	Y5 pulse output stop flag	D8449	Y5 deceleration time (ms) [default value: 100]
M8450	Y5 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8450	Y5 number of clearing soft element
M8451	Y5 output initialization flag	D8451	Reserved
M8452	Y5 output complete interrupt enabling	D8452	Reserved
M8453	Reserved	D8453	Reserved
M8454	Y5 flag of abnormal end of high-speed output	D8454	Reserved
M8455	Y5 PLSV2 instruction accelerating flag	D8455	Reserved
M8456	Y5 PLSV2 instruction decelerating flag	D8456	Reserved
M8457	Reserved	D8457	Reserved
M8458	Reserved	D8458	Reserved
M8459	Reserved	D8459	Reserved
Y6 pulse output port (reserved)			
M8460	Y6 monitoring during pulse output	D8460	Y6 current value register (PLS) (32-bit)
M8461	Y6 active output label for the DSZR/ZRN and other clearing signals	D8461	
M8462	Y6 designation of DSZR/ZRN direction	D8462	Y6 maximum speed (Hz) (32-bit) [default value: 200,000]
M8463	Y6 forward rotation stroke end	D8463	
M8464	Y6 reverse rotation stroke end	D8464	Y6 DSZR homing speed (Hz) (32-bit) [default value: 50000]
M8465	Y6 near point signal logical inversion	D8465	
M8466	Y6 homing signal logical inversion	D8466	Y6 creep speed (Hz) [default value: 2000]
M8467	Y6 S-curve acceleration/deceleration enabling	D8467	Y6 base speed (Hz) [default value: 500]
M8468	Y6 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8468	Y6 acceleration time (ms) [default value: 100]
M8469	Y6 pulse output stop flag	D8469	Y6 deceleration time (ms) [default value: 100]
M8470	Y6 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8470	Y6 number of clearing soft element
M8471	Y6 output initialization flag	D8471	Reserved
M8472	Y6 output complete interrupt enabling	D8472	Reserved
M8473	Reserved	D8473	Reserved

M Element	Description	D Element	Description
M8474	Y6 flag of abnormal end of high-speed output	D8474	Reserved
M8475	Y6 PLSV2 instruction accelerating flag	D8475	Reserved
M8476	Y6 PLSV2 instruction decelerating flag	D8476	Reserved
M8477	Reserved	D8477	Reserved
M8478	Reserved	D8478	Reserved
M8479	Reserved	D8479	Reserved
Y7 pulse output port (reserved)			
M8480	Y7 monitoring during pulse output	D8480	Y7 current value register (PLS) (32-bit)
M8481	Y7 active output label for the DSZR/ZRN and other clearing signals	D8481	
M8482	Y7 designation of DSZR/ZRN direction	D8482	Y7 maximum speed (Hz) (32-bit) [default value: 200,000]
M8483	Y7 forward rotation stroke end	D8483	
M8484	Y7 reverse rotation stroke end	D8484	Y7 DSZR homing speed (Hz) (32-bit) [default value: 50,000]
M8485	Y7 near point signal logical inversion	D8485	
M8486	Y7 homing signal logical inversion	D8486	Y7 creep speed (Hz) [default value: 2000]
M8487	Y7 S-curve acceleration/deceleration enabling	D8487	Y7 base speed (Hz) [default value: 500]
M8488	Y7 flag of reserving the current position after the DSZR/ZRN instruction is executed	D8488	Y7 acceleration time (ms) [default value: 100]
M8489	Y7 pulse output stop flag	D8489	Y7 deceleration time (ms) [default value: 100]
M8490	Y7 [Positioning instruction] Separate setting of the acceleration/deceleration time and modification to the pulse supported	D8490	Y7 number of clearing soft element
M8491	Y7 output initialization flag	D8491	Reserved
M8492	Y7 output complete interrupt enabling	D8492	Reserved
M8493	Reserved	D8493	Reserved
M8494	Y7 flag of abnormal end of high-speed output	D8494	Reserved
M8495	Y7 PLSV2 instruction accelerating flag	D8495	Reserved
M8496	Y7 PLSV2 instruction decelerating flag	D8496	Reserved
M8497	Reserved	D8497	Reserved
M8498	Reserved	D8498	Reserved
M8499	Reserved	D8499	Reserved
Extra high-speed input functions			
M8500	Reserved	D8500	Y0-Yn maximum speed during execution of the positioning instruction [default value: 200,000]
M8501	Reserved	D8501	
M8502	Reserved	D8502	Y0-Yn base speed during execution of the positioning instruction [default value: 100]
M8503	Reserved	D8503	Y0-Yn acceleration/deceleration time during execution of the positioning instruction [default value: 100]
M8504 to M8511	Reserved	D8504 to D8511	Reserved

Appendix B System Error Code

The following table lists error codes.

B.1 System Error Code D8060

Error Code	Error
D8060	I/O range or setting error
1000-1377	X input signal error: incorrect or limit-exceeding serial number
0000-0377	Y input signal error: incorrect or limit-exceeding serial number

B.2 System Error Code D8061

Error Code	Content
D8061	PC hardware error definition
6101	RAM error
6102	Operation circuit error
6103	I/O hardware connection error
6104	External 24V power supply error
6105	System monitor error
6106	System flash read/write error
6107	System I/O setting error
6108	FPGA download error
6109	FPGA configuration data error in flash
6110	Ethernet hardware initialization failure
6111	Extension module configuration different from the actual
6112-6199	Reserved
16100-16199	Reserved
26100-26199	Reserved

B.3 System Error Code D8062

Error Code	Content
D8062	Communication error in control panel or program connection port
6200-6279 serial communication and configuration error codes	
6201	Odd/even check error, overflow error, and frame error
6202	Incorrect communication character
6203	Inconsistent communication data sum
6204	Incorrect data format
6205	Incorrect instruction
6206	Communication element exceeding range
6207	Communication port exceeding range or not existing
6208-6279	Reserved
6280-6299	CAN communication configuration error code
16200-16219 Ethernet configuration error code	
16200	Reserved
16201	Ethernet configuration: unsupported parameter number
16202	Ethernet configuration: incorrect head register address or incorrect head address plus number of registers

Error Code	Content
D8062	Communication error in control panel or program connection port
16203	Ethernet configuration: excessive registers
16204	Ethernet configuration: failure to read or write the register
16205	Ethernet configuration: ACK signal
16206	Ethernet configuration: busy slave station
16207	Ethernet configuration: incorrect station number
16208	Ethernet configuration: memory check error
16209	Reserved
16210	Ethernet configuration: gateway path error
16211	Ethernet configuration: destination gateway path error
16212-16215	Reserved
16216	Ethernet configuration: invalid IP address
16217-16219	Reserved
16220-16239	Extension module configuration error code
16240-16259	USB communication configuration error code
16260-16279 motion control configuration error code	
16260	Incorrect set value in a mechanical unit
16261	Incorrect set value of electronic gear ratio
16262	Using a cam table not configured in the background
16263	No external input master axis selected for electronic cam
16264	Electronic cam slave axis speed exceeding the maximum output speed allowed
16265	Synchronization lower limit larger than upper limit
16266	Master axis setting out of range
16267	Incorrect number of startup latency pulses
16268	Instruction written in cam key point, invalid key point value
16269	Cam encrypted, not allowing instruction to read key points
16270	Incorrect electronic cam slave axis zoom ratio
16271	Incorrect electronic cam configuration unit
16272	Failure to modify a running electronic cam
16273	Repeated use of the electronic cam modification instruction
16274-16279	Reserved
16280-16299	Reserved
26200-26299	Reserved

B.4 System Error Code D8063

Error Code	Content
D8063	Communication error
6300-6379 COM0-COMx serial communication error code	
6301	Odd/even check error, overflow error, and frame error
6302	Incorrect communication character
6303	Inconsistent communication data sum
6304	Incorrect data format
6305	Incorrect instruction
6306	Monitor timer timeout
6307	Reserved
6308	Reserved
6309	Reserved

Error Code	Content
D8063	Communication error
6310	Reserved
6311	Reserved
6312	Incorrect parallel control (1:1) protocol character
6313	Incorrect parallel control (1:1) protocol sum
6314	Incorrect parallel control (1:1) protocol format
6315	Parallel control (1:1) protocol communication timeout
6316-6329	Reserved
6330+10*X	Incorrect Modbus slave station address, larger than 247
6331+10*X	Data frame length incorrect, nonconforming, or smaller than 5
6332+10*X	Incorrect address, standard error frame, or inconsistent sending/receipt address
6333+10*X	CRC error
6334+10*X	Unsupported instruction code, standard error frame, inconsistent data sending/receiving command, or unsupported command
6335+10*X	Receiving timeout
6336+10*X	Data error, standard error frame
6337+10*X	Buffer overflow (none)
6338+10*X	Frame error, standard error frame
6339+10*X	Incorrect serial protocol: failure to configure the correct protocol during execution of the Modbus or RS instruction
6380-6399: CAN communication error code	
6380	Sending timeout
6381	Receiving timeout
6382	CAN transmitting busy
6383	CAN receiving busy
6384-6399	Reserved
16300-16319 Ethernet communication error code	
16300-16311	Reserved
16312	Protocol identifier error, Modbus protocol
16313	Frame length error
16314	Frame timeout
16315	Frame not identified by slave station (for master station only)
16316	Invalid IP address
16317-16319	Reserved
16320-16339	Extension module communication error code
16340-16359	USB communication error code
16360-16379	Control panel and interface communication error code
16380-16399	Reserved
26300-26399	Reserved

B.5 System Error Code D8064

Error Code	Content
D8064	Incorrect system parameter
6401	Program-parameter inconsistency
6402	Incorrect program capacity setting
6403	Incorrect setting of changeable power failure retentive area of soft elements

Error Code	Content
D8064	Incorrect system parameter
6404	Incorrect parameter area setting
6405	Incorrect program area setting
6406-6424	Reserved
6425	User program check error, incorrect downloaded data
6426	Incomplete user program, including motion control subprogram
6427	PLC identifier not matching user program identifier
6428	Factory commissioning error
6429-6452	Reserved
6453-6465	Incorrect setting of changeable power failure retentive area of soft elements
6466-6499	Reserved
16400-16499	Reserved
26400-26499	Reserved

B.6 System Error Code D8065

Error Code	Content
D8065	User program syntax error
6501	Reserved
6502	Reserved
6503	Instruction parameter error
6504	Repeated label definition
6505	Reserved
6506	Using undefined commands
6507	Incorrect Label P definition
6508	Incorrect Label I definition
6509	Reserved
6510	Reserved
6511	High-speed counter using the same input as interrupt
6512-6599	Reserved
16500-16599	Reserved
26500-26599	Reserved

B.7 System Error Code D8066

Error Code	Content
D8066	User program logic circuit error
6601-6604	Reserved
6605	Incorrect instruction used in STL
6606	Incorrect instruction in incorrect position
6607	FOR-NEXT operation error
6608	MC-MCR operation error
6609-6617	Reserved
6618	Instructions allowed only in main program existing in other areas
6619	Unusable instructions existing between FOR and NEXT instructions
6620	Excessive nested layers between FOR and NEXT instructions
6621	Incorrect FOR_NEXT quantitative relationship
6622	No NEXT instruction

Error Code	Content
D8066	User program logic circuit error
6623	No MC instruction
6624	No MCR instruction
6625	STL used for more than nine consecutive times
6626	Unusable instructions existing in STL-RET
6627	No RET instruction
6628	Invalid instructions in the main program
6629	No P or I
6630	No SRET or IRET instruction
6631	SRET unusable in the position
6632	FEND unusable in the position
6633-6699	Reserved
16600-16699	Reserved
26600-26699	Reserved

B.8 System Error Code D8067

Error Code	Content
D8067	Incorrect instruction parameter and running parameter
6701	CALL&CJ calling error
6702	CALL running for more than six times
6703	Reserved
6704	Incorrect communication parameter area setting
6705	Element unavailable or out of range
6706	Data incorrect or out of range
6707	FOR&NEXT, MC, MCR, STL, subprogram, interruption program relationship not clear
6708	FROM or TO instruction error
6709	IRET, SRET, FOR–NEXT relationship not in a match
6710	Local variable used in the main program
6711	Reuse or conflict of soft elements in an instruction
6712	Undefined interrupts used in the system
6713-6719	Reserved
6720	CALL instruction not matching SRET
6721	Subprogram with error parameters
6722	Manipulator instruction port function conflict
6723-6729	Reserved
6730	Sampling time (TS) < 0
6731	Reserved
6732	Abnormal input filter constant object
6733	Abnormal input proportionality coefficient
6734	Abnormal integral time
6735	Abnormal differential gain
6736	Abnormal differential time

Error Code	Content
D8067	Incorrect instruction parameter and running parameter
6737	Reserved
6738	Reserved
6739	Reserved
6740	Abnormal sampling time
6741	Reserved
6742	Measured variable overflow
6743	Abnormal offset
6744	Abnormal integral term
6745	Differential value overflow due to differential restrictor overflow
6746	Abnormal differential term
6747	Abnormal PID result
6748-6759	Reserved
6760-6799 High-speed input/output error code	
6760	Number of high-speed input instruction running entries exceeding limit
6761	High-speed input C counter multiplexing error
6762	High-speed input instruction port repeated or conflict
6763	High-speed input instruction element out of range
6764	High-speed input instruction data out of range
6765	High-speed output instruction element out of range
6766	High-speed output instruction data out of range
6767	Conflict in comparison objects setting of high-speed interrupt comparison instruction
6768	Reserved
6769	Reserved
6770	High-speed output instruction port repeated or conflict
6771	Incorrect high-speed output instruction signal
6772	Incorrect motion control subprogram instruction, compiling incorrect or not existing
6773	Motion control subprogram calling error
6774	Reserved
6775	Motion control subprogram instruction element out of range, function word not matching or existing
6776	Motion control subprogram data incorrect or out of range
6777	High-speed interpolation instruction arc length too small
6778	High-speed interpolation instruction arc parameter incorrect (center or circle setting incorrect, radius too long), resulting in arc generating failure
6779	Helical curve 3rd axis pulses of high-speed output interpolation instruction out of range
6780-6799	Reserved
16700-16799	Reserved
26700-26799	Reserved

Appendix C Customized Function - DHSTP Instruction

DHSTP: High-speed comparison interrupt output instruction

◆ Overview

The DHSTP instruction is used to output high-speed comparison interrupts.

DHSTP S1 S2 S3 S4 S5 S6		Comparison output instruction	Applicable model: H3U
S1	Comparison position		32-bit instruction (13 steps) Continuous execution
S2	Position spacing/number		
S3	Compared object		
S4	Comparison result output		
S5	Output width		
S6	Comparison mode		

◆ Operands

Operand	Bit Soft Element							Word Soft Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S5	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S6	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background are supported.

◆ Functions and actions

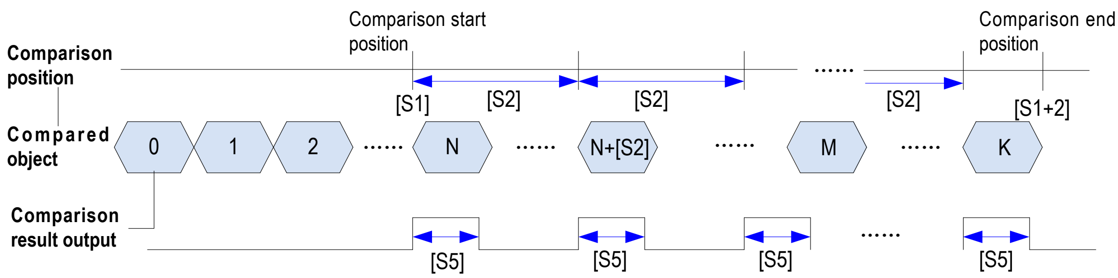
A high-speed comparison interrupt output instruction can be used for aerial photography. The comparison position value set based on instructions is compared with comparison objects successively. When the value equals an object, an output will be generated under control of hardware. The output delay is shorter than 1 us. A maximum of two DHSTP instructions can be output at a time.

	Uniformly-spaced Mode	Non-uniformly-spaced Mode
S1: Comparison position	[S1]: Comparison start position [S1+2]: Comparison end position	[S1]: Comparison position 1 [S1 + 2 x (n – 1)]: Comparison position n 2n elements are occupied. n indicates the number of comparisons set for [S2].
S2: Position spacing/ number	[S2]: Comparison spacing	[S2]: Number of comparisons (n)
	[S2+2]: Number of finished comparisons	
S3: Compared object	[S3]: High-speed counter C235 to C255, or current position value of Y0-Y4 high-speed output corresponding to K0 to K4 D8340/D8360/D8380/D8400/D8420	
S4: Comparison result output	[S4]: High-speed output ports Y0 to Y7	
S5: Output width	[S5]: Output width, in the unit of 0.1 ms, maximum value 50,000	
S6: Comparison mode	[S6] = 1: Uniformly-spaced mode	[S6] = 2: Non-uniformly-spaced mode

◆ Note

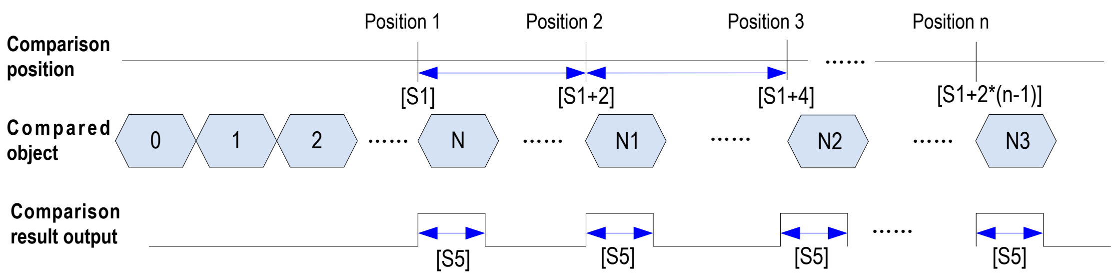
Objects can be compared in uniformly-spaced and non-uniformly-spaced modes.

a) Uniformly-spaced mode



You need to set the comparison start position, comparison end position, and comparison spacing. The comparison object count is compared with objects between the start position and end position in ascending order by the uniform spacing.

b) Non-uniformly-spaced mode



You need to set each comparison position and the number of comparisons. The comparison object count starts from position 1. If the value equals the object, it will be compared with position 2 until all comparisons are finished.

Appendix D Customized Function - Supplementary Description of Electronic Cam Functions

Note: This function is supported only by versions later than FPGA 386xxxxx (the version is displayed through D8022 [32-bit]).

1) Stop Mode Setting

You can use set the stop mode for electronic cams: a stop after the current cycle or an immediate stop by deceleration.

For example, when SM181 is OFF and the y- axis electronic cam enabling element (SM178) is OFF, the electronic cam will stop after the current cycle.

When SM181 is ON and the y- axis electronic cam enabling element (SM178) is OFF, the electronic cam will decelerate by 1 KHz/ms. When the pulse frequency is less than 1 KHz, the system stops outputting pulses. When the cam decelerates, motion superposition is enabled. The deceleration state can be displayed through the motion superposition state flag.

SM Element for Stop Mode Setting			SM Element for Stop by Deceleration		
X-axis	Y-axis	Z-axis	X-axis	Y-axis	Z-axis
SM81	SM181	SM281	SD86	SD186	SD286
OFF: Stop after the current cycle					
ON: Immediate stop/deceleration			OFF: Deceleration disabled ON: Deceleration enabled		

Motion Superposition State		
X-axis	Y-axis	Z-axis
SM22	SM122	SM222
OFF: End of motion superposition ON: Motion being superposed		

2) Using a virtual master axis

The slave axis of an electronic cam or gear is driven by the master axis. Signals of the master axis are needed. The signals can be generated through external input or internal virtual connection. If internal virtual connection is selected, you can select the x-output axis or the internally virtualized master axis. If external input is selected, you can select x-, y-, or z-axis input channel as the master axis.

When internal virtual connection is selected, if the SD value is 0, the internally virtualized master axis is selected, and you need to set the frequency and start/stop through a special register. If the value is 1, the x-output axis is selected. The x-output axis is virtually controlled, and the master axis is internally virtualized for an electronic cam or gear without external connection. When the x-axis serves as a slave axis of an electronic cam, internal virtual connection is unavailable.

If external input is selected, based on values of SD elements, you can select x-, y-, or z-axis input channel as the master axis. If signals of a master axis are externally input, modes of external input (SD60, SD160, and SD260) must match input signals.

SM Element for Master Axis Selection			SD Element for Master Axis Selection		
X-axis	Y-axis	Z-axis	X-axis	Y-axis	Z-axis
SM71	SM171	SM271	SD71	SD171	SD271
OFF: Internal virtual connection			0: Connection to the internally virtualized master axis 1: Internal connection to x-output axis		
ON: External input			1: X-axis input channel 2: Y-axis input channel 3: Z-axis input channel		

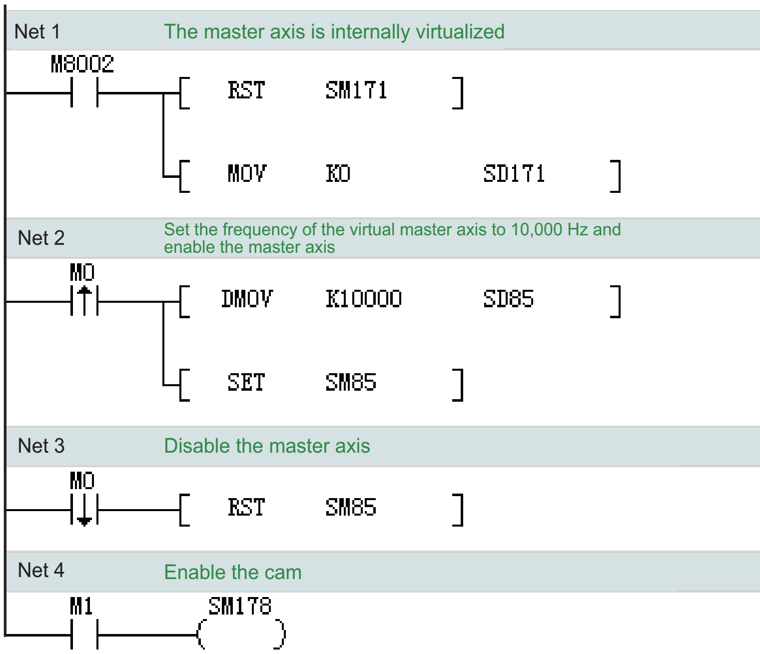
Control of virtual master axis frequency and start/stop

SD86 and SD85	32-bit virtual master axis frequency, ranging from 0 to 200 KHz; 1 KHz by default if the preset value is less than or equal to 0; 200 KHz if the preset value is greater than 200
SM85	ON: Virtual master axis enabled OFF: Virtual master axis disabled

Y-axis: Signals of the master axis are needed. The signals can be generated through external input or internal virtual connection. You can use SM171 to select the master axis.

If SM171 is OFF, internal virtual connection is selected. If the SD171 value is 0, you need to set the frequency (SD) and start/stop control. The master axis will be internally virtualized for an electronic cam or gear without external connection. If the SD171 value is 1, the x-axis will be controlled as the master axis that is internally virtualized for an electronic cam or gear without external connection.

Example: Connection to the internally virtualized master axis



3) Motion superposition

◆ Overview

It is motion pulse superposition.

MOVIMPOS S1 S2 S3 D		Motion superposition	Applicable model: H3U
S1	Number of pulses	Total number of superposed pulses	16-bit instruction (9 steps) MOVIMPOS: Continuous execution
S2	Frequency	Pulse superposition frequency	
S3	Mode	Superposition mode	
D	Output port	Superposed pulse output port	

◆ Operands

Operand	Bit Soft Element							Word Soft Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background correspond to H3U PLCs only.

◆ Functions and actions

Pulses are superposed at the preset frequency.

[S1] indicates the total number of superposed pulses, ranging from -32,767 to +32,767. "+" indicates an increase, and "-" indicates a reduction. Note: Pulses can be increased or reduced in only one direction. Pulse reduction cannot change the direction of a pulse output instruction. Pulses can be reduced until the pulse output frequency is 0.

[S2] indicates the pulse superposition frequency. The value ranges from 1 to 200, in the unit of KHz. The actual frequency may be different from the preset one. Maximum frequency: $Vmax = \sqrt{\frac{P}{2}}$. P indicates the total number of pulses set in [S1]. When the preset frequency is greater than Vmax, Vmax is regarded as the superposition frequency. Actual frequency of pulses output through the port = Positioning instruction frequency ± Superposition frequency

[S3] indicates the superposition mode; 0: acceleration/deceleration mode; others: reserved.

[D] indicates the pulse output port. Y0, Y1, and Y2 indicate the x-axis, y-axis, and z-axis respectively.

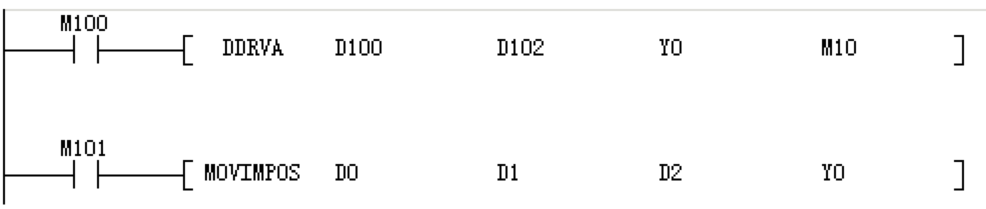
◆ Note

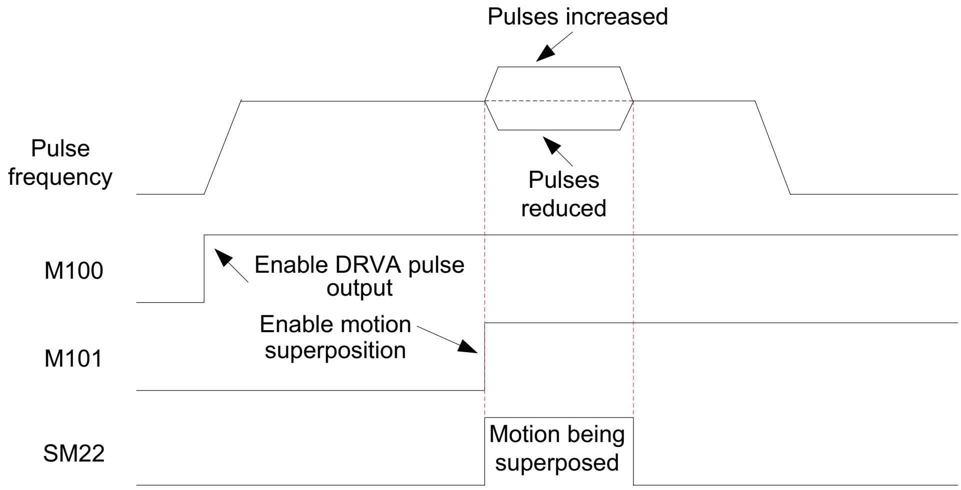
- Special flags can be used to monitor the motion superposition state.

X-axis	Y-axis	Z-axis	Function	
SM22	SM122	SM222	Motion superposition state	ON: Motion being superposed
				OFF: End of motion superposition

- After motion superposition is enabled, the output frequency is increased by 1 KHz/ms. Pulses can be increased or reduced by increasing or reducing the superposition frequency.

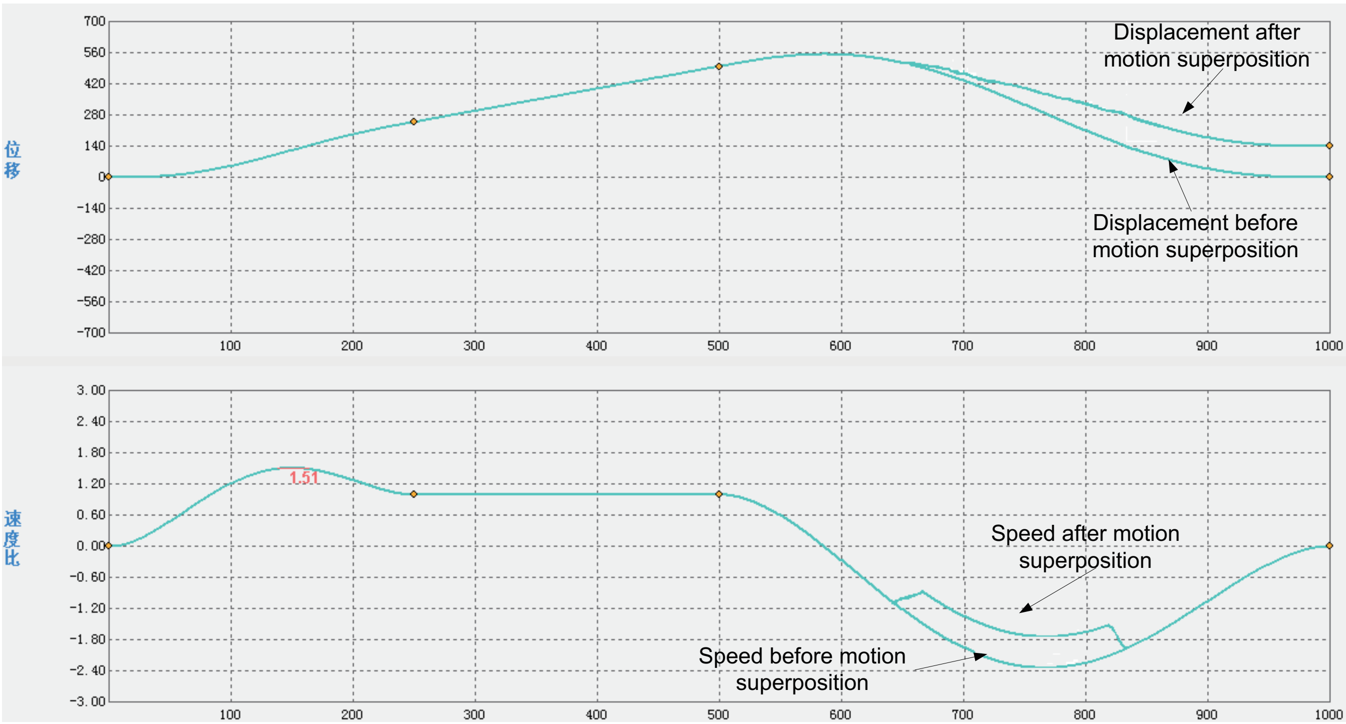
◆ Example





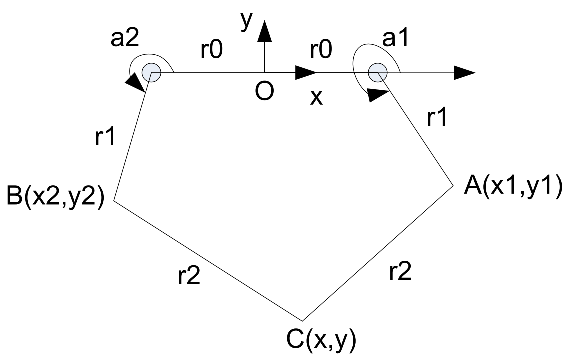
When M100 is ON and the DRVA positioning instruction is executed to output pulses, set M101 to enable motion superposition. In this way, the pulse frequency can be changed by superposing a motion so that pulses can be increased or reduced.

As for electronic cam marking, you can enable motion superposition by detecting the time or distance between two color labels to compensate for the feeding axis in real time and improve the marking accuracy.

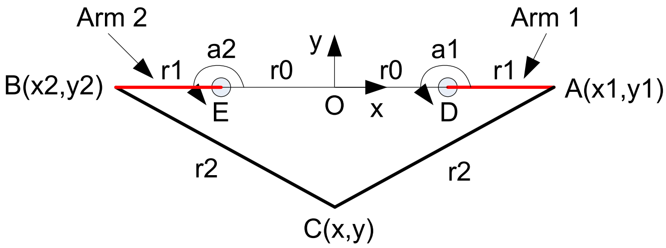


4) Special Delta instruction

The following figure shows a two-degree-freedom Delta manipulator.



As shown in the figure, O is the origin of coordinates, and C(x,y) is the end of the manipulator.



As shown in the preceding figure, the manipulator is in the homing position. Arm 1 and arm 2 rotate around D and E, respectively. C is the end of the manipulator, O is the origin of x- and y-coordinates (0,0). Arm 1 and arm 2 rotate counterclockwise around D and E respectively. The distance between O and D and between O and E is r0, the distance between D and A and between E and B is r1, and the distance between C and A and between C and B is r2.

The following table lists special elements used by a 2-axis Delta manipulator.

SD Element	Function
SD601 and SD600	r0, floating-point number, in the unit of mm
SD603 and SD602	r1, floating-point number, in the unit of mm
SD605 and SD604	r2, floating-point number, in the unit of mm
SD606	Vbas, the base speed of movement along the end path, ranging from 0 to 500 mm/s
SD607	Vmax, the maximum speed of movement along the end path, ranging from 0 to 2000 mm/s
SD608	Tacc, the acceleration time for movement along the end path, ranging from 0 to 2,000 ms
SD609	Tdec, the deceleration time for movement along the end path, ranging from 0 to 2,000 ms
SD611 and SD610	Current x-coordinate position of the manipulator end, floating-point number
SD613 and SD612	Current y-coordinate position of the manipulator end, floating-point number
SD614	Minimum swing angle of arm 1 (-180°to +180°)
SD615	Maximum swing angle of arm 1 (-180°to +180°)
SD616	Minimum swing angle of arm 2 (-180°to +180°)
SD617	Maximum swing angle of arm 2 (-180°to +180°)
SD625 and SD624	Teaching speed ratio, floating-point number, ranging from 1 to 100
SD630	Error message 1: Angle of arm 1 or arm 2 out of range 2: Target coordinate position out of range or manipulator dimensions incorrect 3: Manipulator dimensions inappropriate 4: Output port reused 5: Linear positioning distance insufficient 6: Gantry positioning lifting height insufficient 7: Gantry positioning x-direction movement insufficient

SM Element	Function
SM600	Acceleration/deceleration type, OFF: Trapezoid; ON: S-curve
SM601	ON: Teaching operation, X+
SM602	ON: Teaching operation, X-
SM603	ON: Teaching operation, Y+
SM604	ON: Teaching operation, Y-
SM620	Manipulator operation end flag, valid for one week
SM630	Error flag

◆ Instructions

DT2LIN: Linear positioning of a 2-axis Delta manipulator

DT2LIN S1 S2 S3		Linear positioning	**Applicable model:** H3U
S1	Target x-coordinate position	Target x-coordinate position of the manipulator end, floating-point number	32-bit instruction (13 steps) Continuous execution
S2	Target y-coordinate position	Target y-coordinate position of the manipulator end, floating-point number	
S3	Speed ratio	Ratio of the movement speed to the maximum speed	

◆ Operands

Operand	Bit Soft Element							Word Soft Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background correspond to H3U PLCs only.

◆ Functions and actions

A 2-axis Delta manipulator can be linearly positioned from the current position to the target position.

[S1] indicates the target x-coordinate position of the manipulator end, which is a floating-point number.

[S2] indicates the target y-coordinate position of the manipulator end, which is a floating-point number.

[S3] indicates the ratio of the movement speed to the maximum speed, which is a floating-point number ranging from 1 to 100.

DT2JUMP: Gantry path-based positioning of a 2-axis Delta

DT2JUMP	S1 S2 S3 S4 S5	Gantry positioning	Applicable model: H3U
S1	Target x-coordinate position	Target x-coordinate position of the manipulator end, floating-point number	32-bit instruction (21 steps) Continuous execution
S2	Target y-coordinate position	Target y-coordinate position of the manipulator end, floating-point number	
S3	Speed ratio	Ratio of the movement speed to the maximum speed, floating-point number	
S4	Lifting height	Lifting height from the start point (Lh), floating-point number	
S5	Arc radius	Radius of the arc path (Rh), floating-point number	

◆ Operands

Operand	Bit Soft Element							Word Soft Element															
	System·User							System·User					Bit Designation					Indexed Address			Constant		Real Number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E
S5	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modification	K	H	E

Note: The soft elements in gray background correspond to H3U PLCs only.

◆ Functions and actions

A 2-axis Delta manipulator gantry positioning instruction can be used to position the manipulator from the current position (start point) to the target position (end point) based on the gantry path.

[S1] indicates the target x-coordinate position of the manipulator end, which is a floating-point number.

[S2] indicates the target y-coordinate position of the manipulator end, which is a floating-point number.

[S3] indicates the ratio of the movement speed to the maximum speed, which is a floating-point number ranging from 1 to 100.

[S4] indicates the lifting height from the start point (Lh), which is a floating-point number.

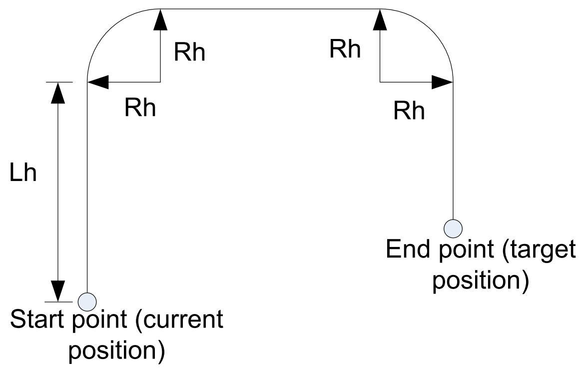
[S5] indicates the radius of the arc path (Rh), which is a floating-point number.

● Note

1. Lh must be greater than the height difference between the start point and end point (the difference between the start y-coordinate and end y-coordinate); otherwise, an error will be returned.

2. When Rh is greater than half of the x-direction movement distance, the system automatically regards Rh as half of the x-direction movement distance. When the x-direction movement distance is less than 10 mm, an error will be returned.

- Gantry positioning path



Memo No. ____________

Date / /

Memo No. ____________

Date / /

Shenzhen Inovance Technology Co., Ltd.
Add.: Building E, Hongwei Industry Park, Liuxian Road, Baocheng No. 70 Zone, Bao'an District, Shenzhen
Tel: +86-755-2979 9595
Fax: +86-755-2961 9897
Service Hotline: 400-777-1260
http: //www.inovance.com

Suzhou Inovance Technology Co., Ltd.
Add.: No. 16 Youxiang Road, Yuexi Town, Wuzhong District, Suzhou 215104, P.R. China
Tel: +86-512-6637 6666
Fax: +86-512-6285 6720
Service Hotline: 400-777-1260
http: //www.inovance.com