



## H1UH2U series programmable logic controller

## Instructions and programming manual



User Guide

A00

Data code 19011368

## Preface

H1U/H2U series PLC is a cost-effective control product developed by Inovance Technology. It has rich instructions, strong high-speed signal processing capabilities, and fast calculation speed. The allowable user program capacity can reach up to 8K steps for H1U and 16K steps for H2U and does not require external storage devices.

The controller is equipped with three independent communication ports as standard, which can be expanded to four, which is convenient for field wiring; the communication ports support multiple communication protocols, including MODBUS master station and slave station protocols, which are especially convenient for online control with inverters and other equipment; It has a strict user program confidentiality function and a separate encryption function for subprograms, which is convenient for users to protect the intellectual property rights of the unique control process. Support USB download and CANlink networking communication.

The controller provides a variety of programming languages, and users can choose programming methods such as ladder diagram, instruction list, step ladder diagram, and SFC sequence function diagram. The instruction system is familiar to most engineering and technical personnel, and the AutoShop software programming environment provided by our company integrates the advantages of many PLC programming environments. The rich online help information makes it easy to use without searching for instructions when programming.

The intellectual property rights of this "Inovance Small PLC Instructions and Programming Manual" belong to Inovance Technology. Our company will continue to update and upgrade according to the situation without prior notice. Users and readers are welcome to visit our company website at any time to download the latest version of the manual and information.

---

# Contents

Chapter 1 Ladder Diagram and Ladder Diagram Program.....	- 19 -
1.1 Ladder Diagram Programming Characteristics: .....	- 20 -
1.2 Component symbols used in ladder diagram programming: .....	- 21 -
1.3 PLC operating principle .....	- 22 -
1.4 Basic Knowledge of PLC Values .....	- 25 -
Chapter 2 How to use H1U/H2U series PLC .....	- 32 -
2.1 PLC System Requirements .....	- 33 -
2.2 Programming and User Program Download .....	- 34 -
2.3 Connecting to HMI .....	- 34 -
Chapter 3 Device elements description .....	- 35 -
3.1 Input Relay X .....	- 36 -
3.2 Output Relay Y.....	- 37 -
3.3 Auxiliary Relay M.....	- 38 -
3.4 Status Relay S .....	- 40 -
3.5 Timer T.....	- 40 -
3.6 Counter C.....	- 42 -
3.7 Register D .....	- 49 -
3.8 Sub-program and Interruption Indicator P, I.....	- 53 -
3.9 Constant K, H.....	- 57 -
3.10 Table of elements .....	- 58 -
Chapter 4 Instructions .....	- 61 -
4.1 Basic instructions.....	- 63 -
4.2 STL/SFC instruction.....	- 70 -
4.2.1 STL Programming instruction .....	- 70 -
4.2.2 SFC Sequential Command Diagram Programming .....	- 72 -
4.3 Application instruction table.....	- 82 -
4.3.1 Program flow (00~09) .....	- 86 -
4.3.2 Transmission and comparison (10~19).....	- 97 -
4.3.3 Four logical arithmetic operations (20~29).....	- 107 -

---

4.3.4 Cyclic shift (30~39) .....	- 117 -
4.3.5 Data processing (40~49) .....	- 124 -
4.3.6 High-speed processing (50~59) .....	- 134 -
4.3.7 Convenient instructions (60~69).....	- 152 -
4.3.8 Peripheral I/O (70~79) .....	- 167 -
4.3.9 Peripheral SER equipment (80~93) .....	- 187 -
4.3.10 Floating number (110~147) .....	- 212 -
4.3.11 Location control (155~159).....	- 237 -
4.3.12 Clock operation (160~169) .....	- 258 -
4.3.14 Contact comparison (224~246) .....	- 269 -
Chapter 5 Communication Function.....	- 273 -
5.1 Summary.....	- 275 -
5.1.1 Hardware and communication connection .....	- 275 -
5.1.2 H2U communication protocol switching logic description .....	- 276 -
5.2 HMI monitoring protocol.....	- 283 -
5.3 Parallel Protocol (1:1) .....	- 284 -
5.4 N:N protocol .....	- 286 -
5.5 N:N Enhanced Protocol .....	- 290 -
5.6 Computer Link protocol.....	- 293 -
5.7 MODBUS protocol description.....	- 301 -
5.7.1 MODBUS master station communication application .....	- 302 -
5.7.2 MODBUS slave communication application.....	- 308 -
5.7.3 The programming method of using MODBUS protocol communication between multiple PLCs... -	316 -
5.7.4 MODBUS configuration and programming method.....	- 319 -
5.7.5 Inovance QLink protocol and programming method .....	- 323 -
5.7.6 Communication programming method between PLC and MD500 series inverter.....	- 326 -
5.7.7 Function code setting of MD500 with PLC application.....	- 327 -
5.7.8 PLC program to modify MD500 function code .....	- 334 -
5.9 CAN communication command description .....	- 337 -
5.9.1 Introduction to CANlink3.0 Application .....	- 337 -
5.9.2 Master-Slave network .....	- 337 -



---

5.9.3 Heartbeat mechanism.....	- 340 -
5.9.4 Offline detection.....	- 340 -
5.9.5 Use of CANlink3.0 .....	- 342 -
5.9.6 Software configuration CANlink3.0 master-slave communication configuration method .....	- 344 -
5.9.7 How to set CAN network configuration table.....	- 347 -
5.9.8 CANlink common faults and troubleshooting .....	- 364 -
Chapter 6 Expansion Modules and Expansion Card Usage.....	- 365 -
6.1 Overview of extension modules.....	- 367 -
6.1.1 Naming rules.....	- 367 -
6.1.2 Expansion model .....	- 368 -
6.1.3 H2U-CAN-BD (or H1U-CAN-BD) expansion card.....	- 369 -
6.1.4 Expansion module address number.....	- 370 -
6.1.5 Module installation dimensions and user interface table .....	- 372 -
6.2 Digital input/output expansion module .....	- 373 -
6.2.1 Digital input expansion module.....	- 373 -
6.2.2 Digital output expansion module .....	- 375 -
6.2.3 Digital input and output expansion module .....	- 377 -
6.3 Analog input module .....	- 380 -
6.4 Analog output module.....	- 387 -
6.5 Analog mixed (input/output) module.....	- 393 -
6.5.1 H2U-4AM 2-channel analog current input/2-channel analog voltage and current output hybrid expansion module.....	- 393 -
6.5.2 H2U-6AM 4-channel analog current input/2-channel analog voltage and current output hybrid expansion module.....	- 402 -
6.5.3 H2U-6CM 4-channel analog voltage input/2-channel analog voltage and current output hybrid expansion module.....	- 410 -
6.6 Temperature control module .....	- 419 -
6.7 Expansion Card Instructions.....	- 437 -
6.7.1 Instructions for H2U Series 6A/6B Expansion Card.....	- 437 -
6.7.2 Instructions for H2U Series Ethernet Communication Expansion Card.....	- 441 -
6.7.3 Instructions for H1U/ H2U Series Communication Expansion Card .....	- 447 -
6.7.4 Instructions for H1U Series 4A Expansion Card .....	- 449 -
6.8 Instructions for use of differential to collector output module.....	- 453 -

---

Chapter 7 Features.....	- 455 -
7.1 Upload is prohibited .....	- 456 -
7.1.1 Introduction to upload prohibition .....	- 456 -
7.1.2 PLC encryption settings.....	- 457 -
7.2 Project authorization protection use (firmware version 24147 and above support).....	- 459 -
7.2.1 Function introduction .....	- 459 -
7.2.2 Project authorization protection .....	- 459 -
7.3 Subroutine encryption/decryption .....	- 461 -
7.4 PLC identifier .....	- 462 -
7.4.1 Setting interface of PLC identifier.....	- 462 -
7.4.2 Modification of PLC identifier.....	- 464 -
7.5 Firmware Upgrade Step Guide.....	- 465 -
7.5.1 Preparation .....	- 465 -
7.5.2 Upgrade operation .....	- 465 -
Chapter 8 Appendix.....	- 469 -
8.1 System special soft components .....	- 470 -
8.2 System error recording function .....	- 488 -
8.2.1 Error code and content .....	- 488 -
8.2.2 PLC running time/abnormal record.....	- 494 -
8.2.3 Use Autosshop software for fault diagnosis and handling .....	- 494 -
8.3 Error code register .....	- 496 -
8.4 Software difference between H2U series MTQ type and MT type.....	- 497 -
8.5 Supplementary Instructions for H2U Series MTP Software .....	- 499 -
8.5.1 Description of DRVI and DRVA special functions .....	- 499 -
8.5.2 Description of ZRN instruction.....	- 502 -
8.6 List of commands not supported by H2U series MTQ/MTP models.....	- 503 -
8.7 Description of enhanced functions of H2U series high-speed processing instructions .....	- 504 -
8.8 Interrelationship of program flow control instructions.....	- 510 -
8.9 Function description of some special relays and registers .....	- 513 -

---

## User Manual Version

Version	Issue Date	Revision Description
V0.0	2020-08	First Release.

## Program logic instruction

Instruction Symbol	Function	Applicable Models		Page No.
		H1U	H2U	
LD	Loading of NO contacts	✓	✓	65
LDI	Loading of NC contacts	✓	✓	65
LDP	Use of rising edge pulse	✓	✓	65
LDF	Use of falling edge pulse	✓	✓	65
AND	Series connection of NO contacts	✓	✓	65
ANI	Series connection of NC contacts	✓	✓	65
ANB	Series connection of circuit blocks	✓	✓	66
ANDP	Series connection for AND rising edge pulse detection	✓	✓	65
ANDF	Series connection for AND falling edge pulse (F) detection	✓	✓	65
OR	Parallel connection of NO contacts	✓	✓	66
ORI	Parallel connection of NC contacts	✓	✓	66
ORB	Parallel connection of circuit blocks	✓	✓	66
ORP	Parallel connection for OR rising edge pulse detection	✓	✓	66
ORF	Parallel connection for OR falling edge pulse detection	✓	✓	66
OUT	Coil drive	✓	✓	67
SET	SET action storage coil instruction	✓	✓	67
RST	Contact or cache clearance	✓	✓	68
PLS	Rising edge pulse detection coil instruction	✓	✓	68
PLF	Falling edge pulse (F) detection coil instruction	✓	✓	68
MC	Coil instruction for serial contacts used by the main control	✓	✓	68
MCR	Release instruction for serial contacts used by main control reset	✓	✓	68
MPS	Stack-based storage	✓	✓	67
MRD	Stack read (current pointer unchanged)	✓	✓	67
MPP	Stack read	✓	✓	67
NOP	No action	✓	✓	69
INV	Operation result inversion	✓	✓	69
END	End of all programs	✓	✓	69
P	Indicator	✓	✓	69
I	Indicator interruption insertion	✓	✓	70

**【Note】 :**

The instructions supported by the H1S-XP model are the same as those supported by the H1U-XP;

The instructions supported by H2S-XP models are consistent with those supported by H2U-XP.

## Application instructions

Type	Instruction Symbol	Description	Applicable Models		Page no.
			H1U	H2U	
Program flow	CJ	Conditional jump	✓	✓	71
	CALL	Subroutine call	✓	✓	72
	SRET	Subroutine Reset	✓	✓	72
	IRET	Interrupt Reset	✓	✓	74
	EI	Interrupt Permit	✓	✓	74
	DI	Interrupt inhibit	✓	✓	74
	FEND	Master Program Termination	✓	✓	78
	WDT	Monitor Timer	✓	✓	79
	FOR	Begin the Cycle Range	✓	✓	81
	NEXT	Cycle Range Finish	✓	✓	80
Transmit and Compare	CMP	Compare	✓	✓	82
	ZCP	Regional comparison	✓	✓	83
	MOV	Transmit	✓	✓	84
	SMOV	Shift Transmit	✓	✓	85
	CML	Reverse Transmit	✓	✓	86
	BMOV	Both Transmit	✓	✓	89
	FMOV	Multiple Point Transmission	✓	✓	90
	XCH	Exchange	✓	✓	91
	BCD	BCD Transform	✓	✓	92
	BIN	BIN conversion	✓	✓	93
Data logical operation	ADD	BIN Addition	✓	✓	94
	SUB	BIN Subtraction	✓	✓	95
	MUL	BIN Multiplication	✓	✓	96
	DIV	BIN Division	✓	✓	97
	INC	BIN plus 1	✓	✓	99
	DEC	BIN minus 1	✓	✓	100
	WAND	Logical Character AND	✓	✓	101
	WOR	Logical Character OR	✓	✓	101
	WXOR	Logical Character XOR	✓	✓	101
	NEG	Complemental Code	✓	✓	103
Cyclical Shift	ROR	Cycle Right Shift	✓	✓	105
	ROL	Cycle Left Shift	✓	✓	105
	RCR	Carry Cycle Right Shift	✓	✓	106
	RCL	Carry Cycle Left Shift	✓	✓	106
	SFTR	Right shift	✓	✓	107
	SFTL	Left shift	✓	✓	107
	WSFR	Shift right by word	✓	✓	109
	WSFL	Character Left Shift	✓	✓	109

Type	Instruction Symbol	Description	Applicable Models		Page no.
			H1U	H2U	
	SFWR	Shift Input	✓	✓	111
	SFRD	Shift Output	✓	✓	112
Data processing	ZRST	Batch Reset	✓	✓	113
	DECO	Decode	✓	✓	114
	ENCO	Encode	✓	✓	115
	SUM	ON Median	✓	✓	116
	BON	ON Median Determination	✓	✓	117
	MEAN	Average	✓	✓	118
	ANS	Signal Alarm Set	✓	✓	119
	ANR	Signal alarm reset	✓	✓	120
	SQR	BIN Square Root	✓	✓	121
	FLT	BIN integer → Binary floating-point conversion	✓	✓	122
High-speed processing	REF	Input/output refresh Input/output refresh	✓	✓	124
	REFF	Filter Adjust	✓	✓	126
	MTR	Matrix input	✓	✓	127
	HSCS	Comparative Set (High-speed counter)	✓	✓	129
	HSCR	Comparative Reset (High-speed counter)	✓	✓	132
	HSZ	Comparative Interval (High-speed counter)	✓	✓	133
	SPD	Pulse density	✓	✓	138
	PLSY	Pulse output	✓	✓	141
	PWM	Pulse Modulation	✓	✓	144
PLSR	Pulse output with Acceleration and Deceleration	✓	✓	145	
Table operations	SER	Seek Data	✓	✓	154
	ABSD	Cam Control (Absolute)	✓	✓	156
	INCD	Cam Control (Incremental)	✓	✓	158
	TTMR	Teach timer	✓	✓	160
	STMR	Special timer	✓	✓	161
	ALT	Alternate state	✓	✓	164
	RAMP	Ramp variable value	✓	✓	165
	RTOC	Rotary workbench control	✓	✓	167
	SORT	Sort Data	✓	✓	169
I / O Peripheral	TKY	Number key input	✓	✓	171
	HKY	Hexadecimal key input	✓	✓	173
	DSW	Digital Switch	✓	✓	175
	SEGD	7-segment decoding	✓	✓	177
	SEGL	7-segment scanning display	✓	✓	178
	ARWS	Arrow switch	✓	✓	180

Type	Instruction Symbol	Description	Applicable Models		Page no.
			H1U	H2U	
	ASC	ASCII code conversion	✓	✓	182
	PR	ASCII Print	✓	✓	184
	FROM	BFM Read	✓	✓	185
	TO	BFM Input	✓	✓	186
Peripheral Equipment	RS	Serial communication instruction	✓	✓	189
	PRUN	Transmission of octal bits	✓	✓	196
	ASCI	HEX-ASCII Conversion	✓	✓	197
	HEX	ASCII-HEX Conversion	✓	✓	199
	CCD	Check parity code	✓	✓	201
	PID	PID control loop	✓	✓	203
	MODBUS	Data transmission	✓	✓	
	CANTX	CAN data transmission	✓	✓	
	CANRX	CAN data reception	✓	✓	
	RS2	Serial data transmission 2	✓	✓	
Number of floating point	MODBUS2	Data transfer 2	✓	✓	
	ECMP	Binary floating-point comparison	✓	✓	211
	EZCP	Binary floating-point interval comparison	✓	✓	212
	EMOV	Binary floating-point transfer			
	EBCD	Binary-Decimal floating-point conversion	✓	✓	213
	EBIN	Decimal-Binary floating-point conversion	✓	✓	214
	EADD	Binary floating-point addition	✓	✓	215
	ESUB	Binary floating-point subtraction	✓	✓	216
	EMUL	Binary floating-point multiplication	✓	✓	217
	EDIV	Binary floating-point division	✓	✓	218
	EXP	Binary floating-point exponential operation	✓	✓	
	LOGE	Natural logarithm operation of binary floating-point numbers	✓	✓	
	DLOG	Common logarithm operation of binary floating-point numbers	✓	✓	
	ESQR	Binary floating-point square root	✓	✓	219
INT	Binary floating-point-BIN integer conversion	✓	✓	220	
Trigonometric functions	SIN	Floating-point SIN calculation	✓	✓	221
	COS	Floating-point Cosine operation	✓	✓	222
	TAN	Floating-point Tangent operation	✓	✓	223
	ASIN	Binary floating-point ASIN operation	✓	✓	
	ACOS	Binary floating-point ACOS operation	✓	✓	
	ATAN	Binary floating-point ATAN operation	✓	✓	
	RAD	Binary floating-point angle → radians	✓	✓	
	DEG	Binary floating-point radian → angle	✓	✓	
	SIN	Hexadecimal floating-point SINH operation	✓	✓	

Type	Instruction Symbol	Description	Applicable Models		Page no.
			H1U	H2U	
	COS	Hexadecimal floating-point COSH operation	✓	✓	
	TAN	Hexadecimal floating-point number TANH operation	✓	✓	
Positioning	SWAP	Upper and lower byte conversion	✓	✓	224
	ABS	ABS position readout	✓	✓	225
	ZRN	Regression through the Origin	✓	✓	227
	PLSV	Variable pulse output	✓	✓	229
	DRVI	Relative Positioning	✓	✓	231
	DRVA	Absolute Positioning	✓	✓	234
Clock Operation	TCMP	Clock data comparison	✓	✓	237
	TZCP	Clock data region comparison	✓	✓	239
	TADD	Clock data addition	✓	✓	240
	TSUB	Clock data subtraction	✓	✓	241
	TRD	Clock data readout	✓	✓	242
	TWR	Clock data input	✓	✓	243
	HOURL	Timer	✓	✓	245
Peripheral Equipment	GRY	Grey code conversion	✓	✓	247
	GBIN	Grey code reverse conversion	✓	✓	248
Contacts comparison	LD=	(S1)=(S2)	✓	✓	252
	LD>	(S1)>(S2)	✓	✓	252
	LD<	(S1)<(S2)	✓	✓	252
	LD<>	(S1)<>(S2)	✓	✓	252
	LD<=	(S1)<=(S2)	✓	✓	252
	LD>=	(S1)>=(S2)	✓	✓	252
	AND=	(S1)=(S2)	✓	✓	254
	AND>	(S1)>(S2)	✓	✓	254
	AND<	(S1)<(S2)	✓	✓	254
	AND<>	(S1)<>(S2)	✓	✓	254
	AND<=	(S1)<=(S2)	✓	✓	254
	AND>=	(S1)>=(S2)	✓	✓	254
	OR=	(S1)=(S2)	✓	✓	256
	OR>	(S1)>(S2)	✓	✓	256
	OR<	(S1)<(S2)	✓	✓	256
	OR<>	(S1)<>(S2)	✓	✓	256
	OR<=	(S1)<=(S2)	✓	✓	256
	OR>=	(S1)>=(S2)	✓	✓	256

**【Note】 :**

The instructions supported by the H1S-XP model are the same as those supported by the H1U-XP;

The instructions supported by H2S-XP models are consistent with those supported by H2U-XP.



## Application instructions (Order by instructions symbols, quick logic instructions not included)

Class.	Instruction Symbol	Function	Applicable Models		Page no.
			H1U	H2U	
A	ABS	ABS current value readout	✓	✓	225
	ABSD	Cam Control (Absolute)	✓	✓	156
	ADD	BIN Addition	✓	✓	94
	ALT	Alternate state	✓	✓	164
	AND=	(S1)=(S2)	✓	✓	254
	AND>	(S1)>(S2)	✓	✓	254
	AND<	(S1)<(S2)	✓	✓	254
	AND<>	(S1)<>(S2)	✓	✓	254
	AND<=	(S1)<=(S2)	✓	✓	254
	AND>=	(S1)>=(S2)	✓	✓	254
	ANR	Signal Alarm Reset	✓	✓	120
	ANS	Signal Alarm Set	✓	✓	119
	ARWS	Arrow Switch	✓	✓	180
	ASC	ASCII code conversion	✓	✓	182
ASCI	HEX-ASCII Conversion	✓	✓	197	
B	BCD	BCD Conversion	✓	✓	92
	BIN	BIN conversion	✓	✓	93
	BMOV	Batch Conversion	✓	✓	89
	BON	ON Median Determination	✓	✓	117
C	CALL	Subroutine call	✓	✓	72
	CANTX	CAN data transmission	✓	✓	
	CANRX	CAN data reception	✓	✓	
	CCD	Test Code	✓	✓	201
	CJ	Conditional jump	✓	✓	71
	CML	Inverse transmission	✓	✓	87
	CMP	Compare	✓	✓	82
	COS	Floating-point Cosine operation	✓	✓	222
COSH	Binary floating-point COSH operation	✓	✓		
D	DEC	BIN minus 1	✓	✓	100
	DECO	Decoder	✓	✓	114
	DI	Interrupt Disable	✓	✓	74
	DIV	BIN Division	✓	✓	97
	DLOG	Common logarithmic operations of binary floating-point	✓	✓	
	DRVA	Absolute Positioning	✓	✓	234
	DRVI	Relative Positioning	✓	✓	231
	DSW	Digital Switch	✓	✓	175
DSZR	Return to origin with DOG search	✓	✓	175	

Class.	Instruction Symbol	Function	Applicable Models		Page no.
			H1U	H2U	
E	EADD	Binary floating-point	✓	✓	215
	EBCD	Binary-Decimal floating-point conversion	✓	✓	213
	EBIN	Decimal-Binary floating-point conversion	✓	✓	214
	ECMP	Binary floating-point number comparison	✓	✓	211
	EDIV	Binary floating-point division	✓	✓	218
	EI	Interrupt Permit	✓	✓	74
	EMUL	Binary floating-point multiplication	✓	✓	217
	ENCO	Encode	✓	✓	115
	ESQR	Binary floating-point square root	✓	✓	219
	ESUB	Binary floating-point subtraction	✓	✓	216
	EXP	Binary floating-point exponential operation	✓	✓	
	EZCP	Binary floating-point interval comparison	✓	✓	212
F	FEND	Master Program Termination	✓	✓	78
	FLT	BIN integer → Binary floating-point conversion	✓	✓	122
	FMOV	Multiple Point Transmission	✓	✓	90
	FOR	Begin the Cycle Range	✓	✓	80
	FROM	BFM Read	✓	✓	185
G	GBIN	Grey code reverse conversion	✓	✓	248
	GRY	Grey code conversion	✓	✓	247
H	HEX	ASCII-HEX Conversion	✓	✓	199
	HKY	Hexadecimal key input	✓	✓	173
	HOUR	Chronograph	✓	✓	245
	HSCR	Comparative Reset (High-speed counter)	✓	✓	132
	HSCS	Comparative Reset (High-speed counter)	✓	✓	129
	HSZ	Interval Comparison (High-speed counter)	✓	✓	133
I	INC	BIN plus 1	✓	✓	99
	INCD	Cam Control (Incremental)	✓	✓	158
	INT	Floating-point - Integer conversion	✓	✓	220
	IRET	Interrupt Reset	✓	✓	74
	IST	Status initialization	✓	✓	148
L	LD=	(S1)=(S2)	✓	✓	252
	LD>	(S1)>(S2)	✓	✓	252
	LD<	(S1)<(S2)	✓	✓	252
	LD<>	(S1)<>(S2)	✓	✓	252
	LD<=	(S1)<=(S2)	✓	✓	252
	LD>=	(S1)>=(S2)	✓	✓	252
	LOGE	Natural logarithmic operations of binary floating-point numbers	✓	✓	
M	MEAN	Average	✓	✓	118

Class.	Instruction Symbol	Function	Applicable Models		Page no.
			H1U	H2U	
	MOV	Transmit	✓	✓	84
	MODBUS	Data transmission	✓	✓	
	MODBUS2	Data transmission 2	✓	✓	
	MTR	Matrix input	✓	✓	127
	MUL	BIN Multiplication	✓	✓	96
N	NEG	Complemental Code	✓	✓	103
	NEXT	Cycle Range Finish	✓	✓	80
O	OR=	(S1)=(S2)	✓	✓	256
	OR>	(S1)>(S2)	✓	✓	256
	OR<	(S1)<(S2)	✓	✓	256
	OR<>	(S1)<>(S2)	✓	✓	256
	OR<=	(S1)<=(S2)	✓	✓	256
	OR>=	(S1)>=(S2)	✓	✓	256
P	PID	PID control loop	✓	✓	203
	PLSV	Variable speed pulse output	✓	✓	229
	PLSY	Pulse output	✓	✓	141
	PLSR	Pulse output with Acceleration and Deceleration	✓	✓	145
	PR	ASCII Print out	✓	✓	184
	PRUN	Octal transmission	✓	✓	196
	PWM	Pulse range adjustment	✓	✓	144
R	RAMP	Ramp variable value	✓	✓	165
	RCL	Carry Cycle Left Shift	✓	✓	106
	RCR	Carry Cycle Right Shift	✓	✓	106
	REF	Input/output refresh	✓	✓	124
	REFF	Filter adjust	✓	✓	126
	ROL	Cycle Left Shift	✓	✓	105
	ROR	Cycle Right Shift	✓	✓	105
	ROTC	Rotary workbench control	✓	✓	167
	RS	Serial data transmission	✓	✓	189
	RS2	Serial data transmission 2	✓	✓	189
S	SEGD	7-segment decoder	✓	✓	177
	SEGL	7-segment scanning display	✓	✓	178
	SER	Seek Data	✓	✓	154
	SFRD	Shift Output	✓	✓	112
	SFTL	Left shift	✓	✓	107
	SFTR	Right shift	✓	✓	107
	SFWR	Shift register write	✓	✓	111
	SIN	Floating-point SIN calculation	✓	✓	221
	SINH	Binary floating-point number SINH operation	✓	✓	221

Class.	Instruction Symbol	Function	Applicable Models		Page no.
			H1U	H2U	
	SMOV	Floating-point SIN calculation	✓	✓	85
	SORT	Sort Data	✓	✓	169
	SPD	Pulse density	✓	✓	138
	SQR	BIN Square Root	✓	✓	121
	SRET	Sub-program Reset	✓	✓	72
	STMR	Special timer	✓	✓	161
	SUB	BIN Subtraction	✓	✓	95
	SUM	ON Median	✓	✓	116
	SWAP	Upper and lower byte conversion	✓	✓	224
T	TADD	Clock data addition	✓	✓	240
	TAN	Floating-point Tangent operation	✓	✓	223
	TANH	Binary floating-point number TANH operation	✓	✓	223
W	WAND	Logical AND	✓	✓	79
	WDT	Monitor Timer	✓	✓	79
	WOR	Logical Character OR	✓	✓	101
	WR3A	Analog module write	✓	✓	
	WSFL	Character Left Shift	✓	✓	109
	WSFR	Character Shift Left	✓	✓	109
X	WXCR	Logical Character XOR	✓	✓	101
	XCH	Exchange	✓	✓	91
Z	ZCP	Interval Comparison	✓	✓	83
	ZRN	Regression through the Origin	✓	✓	227
	ZRST	Batch Reset	✓	✓	113

**【Note】 :**

The instructions supported by the H1S-XP model are the same as those supported by the H1U-XP;

The instructions supported by H2S-XP models are consistent with those supported by H2U-XP.

## 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	Explanation of simple logic instructions	Refer to P38~P41 instruction details
2	Explanation of application instructions	Refer P7~P10 according to the instruction name, and then check the detailed instructions
3	Selection of timer	See section 3.5
4	Selection of high-speed counter	See section 3.6
5	Use of high-speed output instructions	Consult the instructions HSCS/HSCR/HSZ/PLSY/PLSV/PWM
6	Use and setting of input interrupts	See section 3.8
7	Use and setting of timed interrupts	See section 3.8
8	Use and setting of high-speed counting interrupt	See section 3.8
9	STL/SFC programming method	See section 4.2
10	Setting method of communication format	See detailed explanation of RS instructions
11	Setting methods of various communication protocols	See detailed explanation of RS instructions
12	How to use MODBUS master instructions	See detailed explanation of RS instructions
13	How to use MODBUS to access H1U/H2U series PLC	MODBUS built-in slave protocol definition
14	How to use analog expansion card	See section 6.7.1
15	High-speed function and use of MTQ/MTP	Refer to the 8.4/8.5 appendix for high-speed instructions
16	Pulse capture function	See interrupt P description, M8170~M8175 variable description
17	Enhanced version of high-speed processing instruction parameters	See 8.5 appendix, enhanced high-speed instructions
18	Function description of some special relays and registers	See 8.7 appendix





# Chapter 1 Ladder Diagram and Ladder Diagram Program

---

## Contents

1.1 Ladder Diagram Programming Characteristics:.....	- 20 -
1.2 Component symbols used in ladder diagram programming: .....	- 21 -
1.3 PLC's operating principle .....	- 22 -
1.4 Basic Knowledge of PLC Values .....	- 25 -

## Chapter 1 Ladder Diagram and Procedure

### 1.1 Ladder Diagram Programming Characteristics

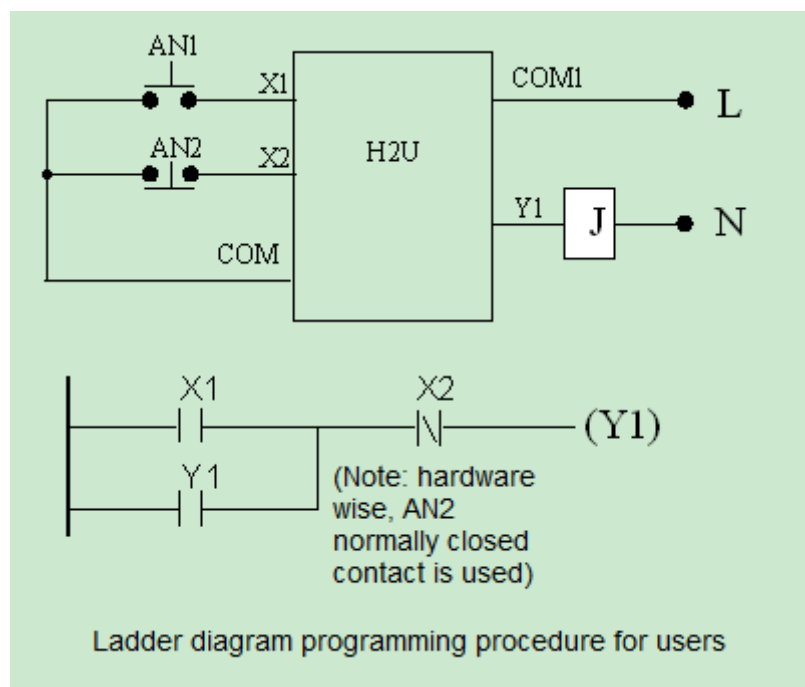
The ladder diagram programming in PLC is a designing method that adapts the electrical principle used in the conventional relay control system. The components used in a design such as switch X, intermediate relay M, time relay T, counter C, and contacts, all have similar characteristics of the actual electrical components.

In a ladder diagram, “contact” and “coil” components are often used. The contact components have two types of “normally open” and “normally closed”, which are corresponding to the “A contact” and “B contact” technical terms used in the electrical industry. The “contact” of one relay in a PLC can be used infinitely. Therefore, we can often assume one relay component (regardless of intermediate relay M, time relay T, or counter C) has unlimited “A” and “B” contact points.

For time relay, counter, existing coil (signal trigger terminal), and contacts, certain components still have the power-down and retention characteristics. Therefore, it is necessary to select appropriate components to achieve functionalities that are desired.

With the development of modern PLC, not only sequential logic control function can be accomplished, other data calculation functions can also be carried out as well. These functions are such as data comparison, four operations of arithmetic, functional operation, etc. The size of data consists of 16-bit, 32-bit, and floating-point, etc. The H1U/H2U series PLC provides a large number of D register components which can be used in a ladder diagram for calculation.

The design idea of the ladder diagram is basically the same as that of the traditional relay control system. Take the electrical principle of a common electromagnetic switch as an example:



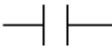

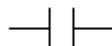
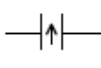
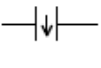
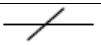
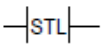

As seen in the illustration, ‘J’ is the relay or contactor. AN1 is the button that starts up ‘J’, and it is used as NO contact. AN2 is the button that disconnects ‘J’, and it is used as NC contact. In addition, ‘J’ normally open auxiliary contact is used for the purpose of status retention.




If the signal input connection and the ladder diagram programming are completed according to the design in the illustration below, identical controlling capabilities will then be able to be accomplished. (For safety consideration, "normally closed" contact is usually used for STOP button)

## 1.2 Component symbols used in ladder diagram programming

The symbols and characteristics of the components used in the ladder diagram design are described in the following table. The logical combination of the "AND" and "OR" of these "contact" components is output to the component "coil":

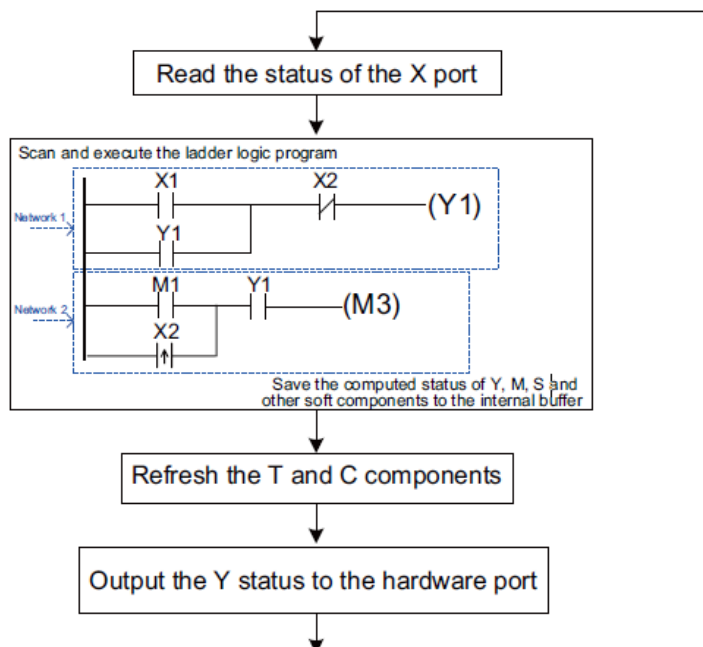
Symbol	Description	Operation Characteristic
	Contact elements, which represent normally open contacts of elements, include input X signal contacts, output Y contacts, intermediate relay M, time relay T, counter C output contacts, etc. For Y, M, T, C and other components, it is also OFF in the non-operating state.	<p>X: When the X port signal contact is closed, the state is ON; when the port signal is open, the contact state is OFF</p> <p>Y: When the "coil" of Y relay is energized, it is ON, otherwise it is OFF. The last state of Y will correspond to the state of the output Y port of the PLC.</p> <p>M: ON when the "coil" of M relay is energized, otherwise OFF</p> <p>S: When S is used as a common sign component, the "coil" of the S relay is ON when it is energized, otherwise it is OFF</p> <p>T: When the corresponding time relay coil is energized and the timing time reaches the set time, the status is ON; otherwise, it is OFF</p> <p>C: When the reading of the corresponding counter reaches the set time, the status is ON; otherwise, it is OFF</p>
	Contact element, which represents the normally closed contact of the element, has input X signal contact, output of Y contact, intermediate relay M, time relay T, counter C Contacts, etc.	The logic and state are just opposite to the signal 
	Contact element, only effective on the rising edge of contacts	When the state of the contact element (X, Y, M) changes from OFF→ON on the rising edge, the signal is valid. This contact signal is valid in one scan period. If the next state does not change, the signal will return to "OFF"
	Contact element, only effective on the falling edge of contacts	When the state of the contact element (X, Y, M) changes from the falling edge of ON→OFF, the signal is valid. This contact signal is valid in one scan period. If the next state does not change, the signal will return to "OFF"
	Inverse Status	Inverse the status of current signal point
	Step ladder diagram shows 'S' state signal	Step instruction state transition
	Coil component, is being powered under ladder diagram	When the coils of the Y and M elements are "energized", their normally open contact action is closed, and their normally closed contact action is disconnected, and the original state is restored when "power is lost"

		<p>When the coil of the T element is "energized", it starts timing, and returns to the default state when "energized". When the timing time reaches the set value, its normally open contact action is closed, and its normally closed contact action is disconnected.</p> <p>When the coil of the C element is "energized", the count value increases by 1. When the count value reaches the set value, its normally open contact is closed, and its normally closed contact is disconnected. The operation of clearing its "coil" can restore its count value and contacts to the default state.</p> <p>Note: The X input component has no coil, and the user program cannot modify its state, only the external user line determines its state.</p>
	<p>Operation instructions to operate components or coils, parameters, etc.</p>	<p>Complete many functions such as logic operation and data processing. Such as [RST Y0], [SET M2], [MOV K5 D100], [JC P1] and other commands.</p>

### 1.3 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/T/subscriber terminal can respond

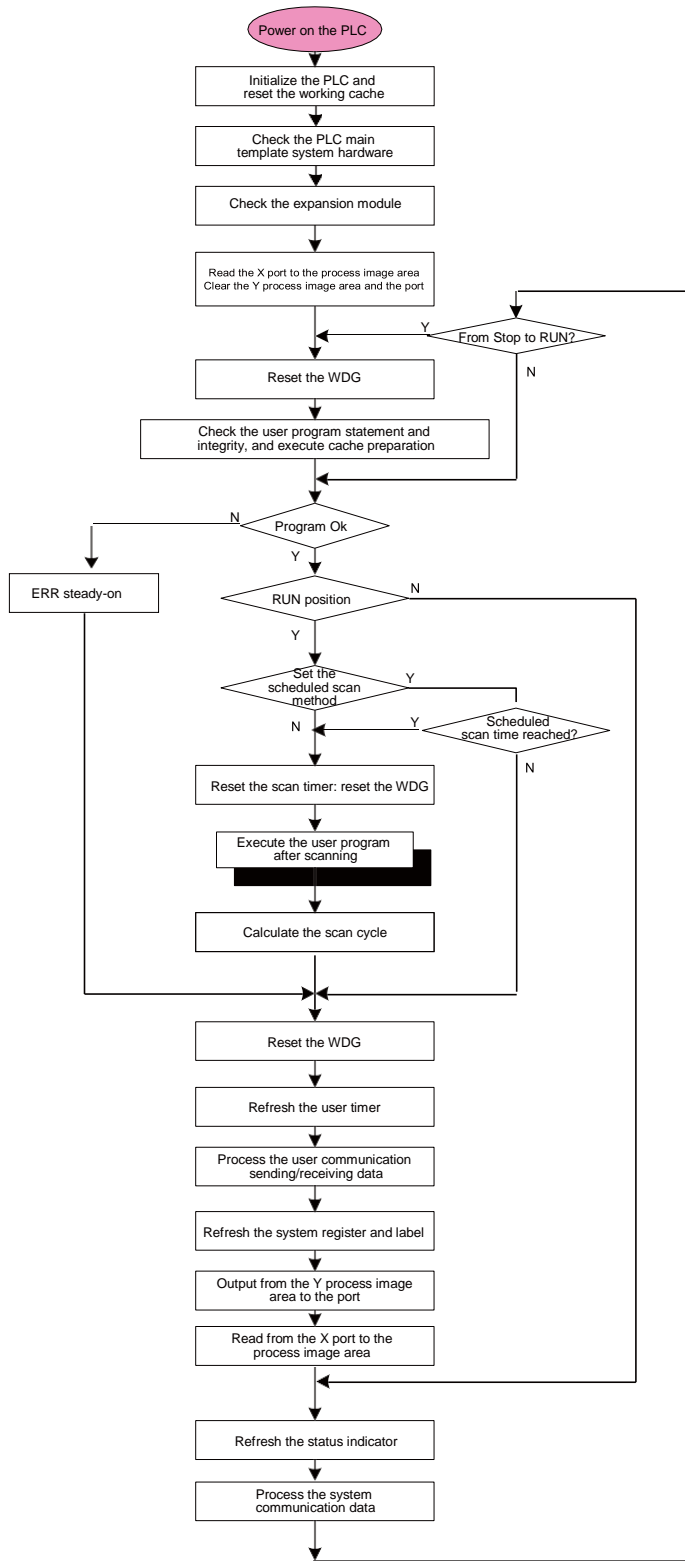
X pulse catch can be registered

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)



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 receive 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).

## 1.4 Basic Knowledge of PLC Values

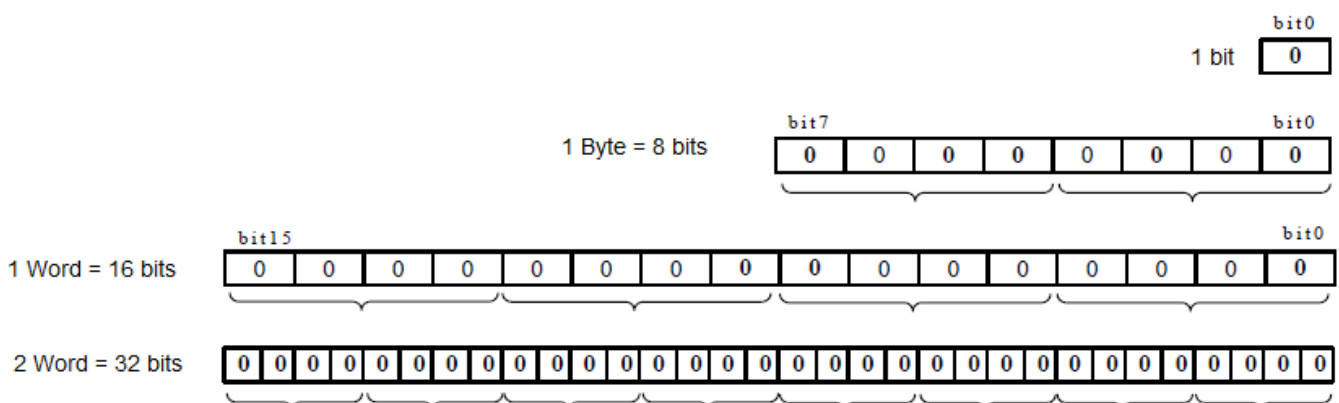
The H1U/H2U Series PLC uses high-performance 32-bit as its core processor. The operating principle is similar to that in other computers. All CPU processors use binary code as the format for their internal data processing. "Data" in a computer is processed in the method of "electric signal level". The signal level only has two statuses, "low" or "high", which correspond to "0" or "1" in the binary system. Therefore, no erroneous judgment will happen in a signal circuit, and this further ensures the accuracy of processing results.

### Binary Data

"Binary" is the most convenient and simply calculation system used in computers. Calculations that involves with single digit are:

$0+0=0$ ;  $0+1=1$ ;  $1+0=1$ ;  $1+1=10$  (a digit is carried over therefore, it requires two digits to represent).

These calculations will only require the typical "AND", "OR", or "NOT" logic circuits to combine and complete operations.



When value that requires processing gets larger, multiple binary digits are required. The greater number of digits used, the bigger the value can be expressed. The commonly used CPU bits are:

Bit Number	Maximum processing value in one time	Application
4bit	15	It is still used in simple consumer products, but it is rarely used
8bit	255	Such as 8051, often used in simple control systems
16-bit	65,535	Such as 808x, used in industrial control, less used
32-bit	4,294,967,295	Such as ARM, currently widely used in industrial control and consumer products
64bit	18446744073709551615	Used in general-purpose computers

A CPU with deficient bit number cannot process large values at once. What it requires is multiple calculations, therefore, programmers need to be familiar with the calculations. It is just like for a load a big truck carries, a small car requires several more trips to finish the load. The smaller the car, the more trips and time it takes.

In the H1U/2U Series PLC component, the commonly used data bandwidth is 1 Word (16-bit); some counters are 2 Word (32-bit). As for the symbol-free data in 16-bit, the maximum value expressed in binary format is 1111, 1111, 1111, 1111. If converted to decimal format then it becomes 65, 535.

#### Hexadecimal

It is still readable when the binary value is small. But when the number of digits gets more and more, readability becomes an issue. Therefore, if every four binary values are grouped together and represented by one number, then it becomes a hexadecimal (HEX). The readability becomes much greater when one 16-bit binary value is expressed in four hexadecimal numbers. Numbers from 10~15 (decimal) under the hexadecimal format are expressed using characters A~F.

#### Octal

Due to conventional practices in computers, values with 8-bit bandwidth and the using of hardware ports are most commonly used. 8-bit is defined as 1 Byte (1 Word). In PLCs, 8 hardware ports are also grouped together for the convenience in communicating operation (read or write), such as the serial numbers of input X port and output Y port still use the octal format.

Octal is composed of three binary numbers ranges from 000~111, or 0~7. There is no 8 or 9. Although CPU is usually in 8, 16, or 32-bits, hexadecimal is still used when comes to data calculation.

#### Decimal

Decimal system is mostly used during our daily lives. The ten numbers are from 0 to 9. When calculating "9" + "1", the process will carry over to next digit and obtain "10".

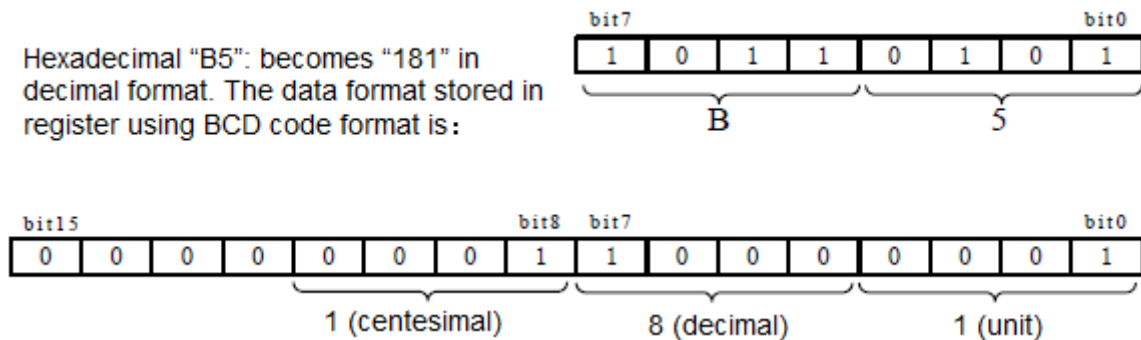
There is other calculation system in life, such as Sunday, Monday, etc. They are represented by seven numbers from 0 to 6 and understood as "heptagon". However, heptagon system is not easy to calculate therefore it is not used often.

#### BCD Code

The numerical format that suits people's reading habit the most is the decimal format. Decimal format is usually adopted to display data when monitoring or configuring work parameters and yet, HEX format is used in

the computer. Therefore, every four binary values are used as a foundation to compose one digit, and every digit can only be in decimal format from 0 to 9. The encoding of numbers composed in the register is called BCD code (Binary-Coded Decimal).

Inside a PLC, in principle four binary numbers are used to represent one decimal number and in every BCD code, there is no A~F character from the HEX format. For a 8bit as for a register unit that has 8-bit bandwidth, the maximum BCD number that can be stored is 99. Therefore, after converting HEX format to BCD code, it will occupy more storage space.



Inside a PLC, data calculation is always done using HEX format. Before initiating and displaying data on a mis-intelligent equipment (such as numeral tubes), HEX formatted data must be converted to BCD codes first inside the PLC in order to perform display output. Before users saving the parameters configured in decimal format into the PLC cache, it is often required to convert BCD codes to HEX format first.

In the H1U/H2U Series PLC, it provides commands for converting between HEX and BCD coding formats. Whenever necessary, the commands can be executed to perform display output or configuration read switches.

The decimal readouts people see on a computer monitor was only displayed after the computer has completed BCD conversion. The parameters modified during monitoring are written in after the computer software performed the HEX conversion. There is no manual interference required.

Comparison Chart of the Types of Carry-Over Systems:

Binary BIN	Octal OCT	Decimal DEC	Hexadecimal HEX	BCD Code	Binary BIN	Octal OCT	Decimal DEC	Hexadecimal HEX	BCD Code
0000	0	0	0	0000	1000	10	8	8	1000
0001	1	1	1	0001	1001	11	9	9	1001
0010	2	2	2	0010	1010	12	10	A	00010000
0011	3	3	3	0011	1011	13	11	B	00010001
0100	4	4	4	0100	1100	14	12	C	00010010
0101	5	5	5	0101	1101	15	13	D	00010011
0110	6	6	6	0110	1110	16	14	E	00010110
0111	7	7	7	0111	1111	17	15	F	00010101

### Conversion of carry-over systems:

Conversions between binary, octal, and hexadecimal systems are very simple. For instance, 8 digits of

binary are expressed as “10110101”. When it converts to hexadecimal, every four digits are grouped together from right to left as “1011, 0101”, and are expressed as “B5”.

When converting to octal, every 3 digits are grouped together from right to left as “10, 110, 101”, and are expressed as “265” in octal.

To convert binary numbers to decimal numbers requires much complicated calculations. The most used method is through power-weight accumulation method, starts from the rightmost digit:

When first digit (bit 0) is 1, 0 to the power of 2 ( $2^0$ ) is 1, otherwise it is 0;

When second digit (bit 1) is 1, 1 to the power of 2 ( $2^1$ ) is 2, otherwise it is 0;

When third digit (bit 2) is 1, 2 to the power of 2 ( $2^2$ ) is 4, otherwise it is 0;

When fourth digit (bit 3) is 1, 3 to the power of 2 ( $2^3$ ) is 8, otherwise it is 0;

When fifth digit (bit 4) is 1, 4 to the power of 2 ( $2^4$ ) is 16, otherwise it is 0;

When sixth digit (bit 5) is 1, 5 to the power of 2 ( $2^5$ ) is 32, otherwise it is 0;

When seventh digit (bit 6) is 1, 6 to the power of 2 ( $2^6$ ) is 64, otherwise it is 0;

When eighth digit (bit 7) is 1, 7 to the power of 2 ( $2^7$ ) is 128, otherwise it is 0;

Therefore, when converting “10110101” into decimal format, the result is  $(128+0+32+16+0+4+0+1) = 181$ .

When converting from 16-bit into decimal format, such as “B5” in the example, to the power accumulation in hexadecimal is also adopted, starting from the rightmost digit:

When first HEX value, 1 to the power of 16, ( $16^0$ ) is 1, actual value of the digit x 1;

When second HEX value, 2 to the power of 16, ( $16^1$ ) is 16, actual value of the digit x 16;

When third HEX value, 3 to the power of 16, ( $16^2$ ) is 256, actual value of the digit x 256;

When fourth HEX value, 4 to the power of 16, ( $16^3$ ) is 4096, actual value of the digit x 4096;

Therefore, when converting “B5” into decimal format, the result is  $(B \times 16 + 5 \times 1) = (11 \times 16 + 5) = 181$ .

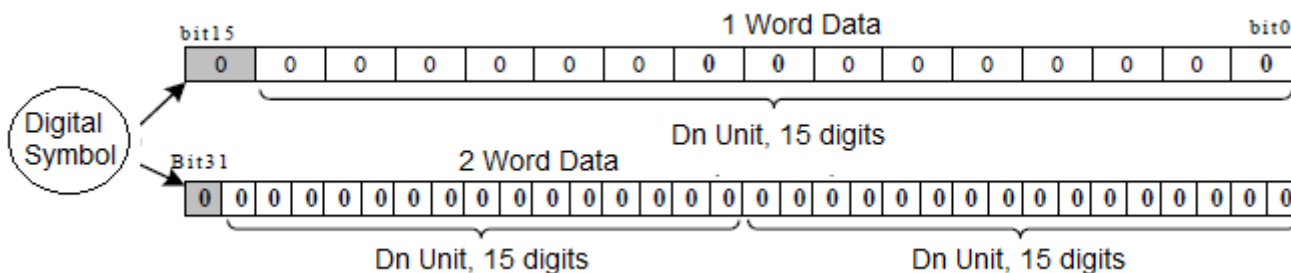
After familiarizing with the conversion from HEX to decimal, users may convert binary or octal value to hexadecimal format first (group every 4-bit), and then convert it to decimal format. The calculation will be much simpler and faster.

### Signed and unsigned numbers

Four fundamental operations of arithmetic can be performed inside a PLC, and the calculation results may produce negative numbers. Such results produced “signed numbers”. In fact, the data in register D inside the H1U/H2U Series PLC and in the 32-bit counter C can all be calculated using the four operations and function commands in the format of “signed numbers”.

The highest digit (bit 15) in the 16-bit D register is used as the representative symbol. Therefore, the value-selecting range of a D register is -32,768~32,767. When using two characters (32-bit, 2 continuous D register) to represent one data, the highest digit (bit 31) should be used as the representative symbol. Therefore, the value-selecting range of a D register is -2,147,483,648~2,147,483,647. The symbol digits are as following:





When symbol digit is 0, it means it is positive. Therefore, 1 Word’s positive value in the maximum HEX format is H7FFF, which is 32767; 2 Word’s positive value in the maximum HEX format is H7FFFFFFF, which are 2,147,483,647.

When symbol digit equal 1 it means it is negative, which is the make-up code for the value. The absolute value is calculated as “inverse every symbol digit individually, and then plus 1”. Take HFFFF of HEX format as an example, the absolute value is =H0000+1=1, in which “HFFFF” represents -1; another example of H8000 in HEX format, its absolute value =H7FFF+1=32768, in which the signed value H8000 is -32768 and it is the smallest negative value in 1 Word register. Using the same principle, the smallest negative value of 2 Word is the signed value of H80000000, which is -2, 147, 483, and 648.

When performing addition or subtraction calculations for larger values, be cautious of the sign. It is especially necessary when performing carrying or borrowing operations. “Digits borrowing label” and “digits carrying label” must be carefully diagnosed and processed, otherwise calculation may result in errors.

Unsigned symbol value means there is no symbol sign, and all is defaulted as positive value. For 1 Word register, the value-selecting range is from 0 to 65535. There is only positive value under some application circumstances, such as timing and counting circumstances. There is no need for signed symbol processing. When performing addition or subtraction operation, it is necessary to avoid calculation overflow, or it may result in calculation error.

When performing logic operations (such as “AND”, “OR” operation commands), operands are processed as unsigned values. Bit15 and other equivalent digits are calculated logically together.

**Floating points**

Floating-point numbers are used to approximate any real number in PLC (or computer). The specific format is an integer or fixed-point number (that is, the mantissa) multiplied by a base number (usually 2 in a computer). This representation method Similar to scientific notation with base 10.

A floating-point number can be represented by  $m \times b^e$ . where, m is the mantissa, in the form of  $\pm d.ddd\dots dd$ ; b is the base; e is the exponent. For example, 988436216 can be represented by a decimal floating-point number as  $9.8844 \times 10^8$ . Because the mantissa is rounded, the precision is reduced. But you can see that using floating-point numbers can represent a larger range of values.

A floating-point number is represented in a computer, and its structure is as follows:

Odd amount (fixed point decimal)		Step codes (fixed point round number)	
Symbol, ±	Odd amount, m	Step codes, ±	Step codes, e

Floating points used in a PLC or a computer, all use the international standard format. For the convenience of computing, instruments all adopt the binary floating-point format. One floating-point occupies 32-

bit of register unit. When using floating-point, users do not need to have thorough understanding of the floating-point format. The computer will automatically process and format the entered number as per the standards.

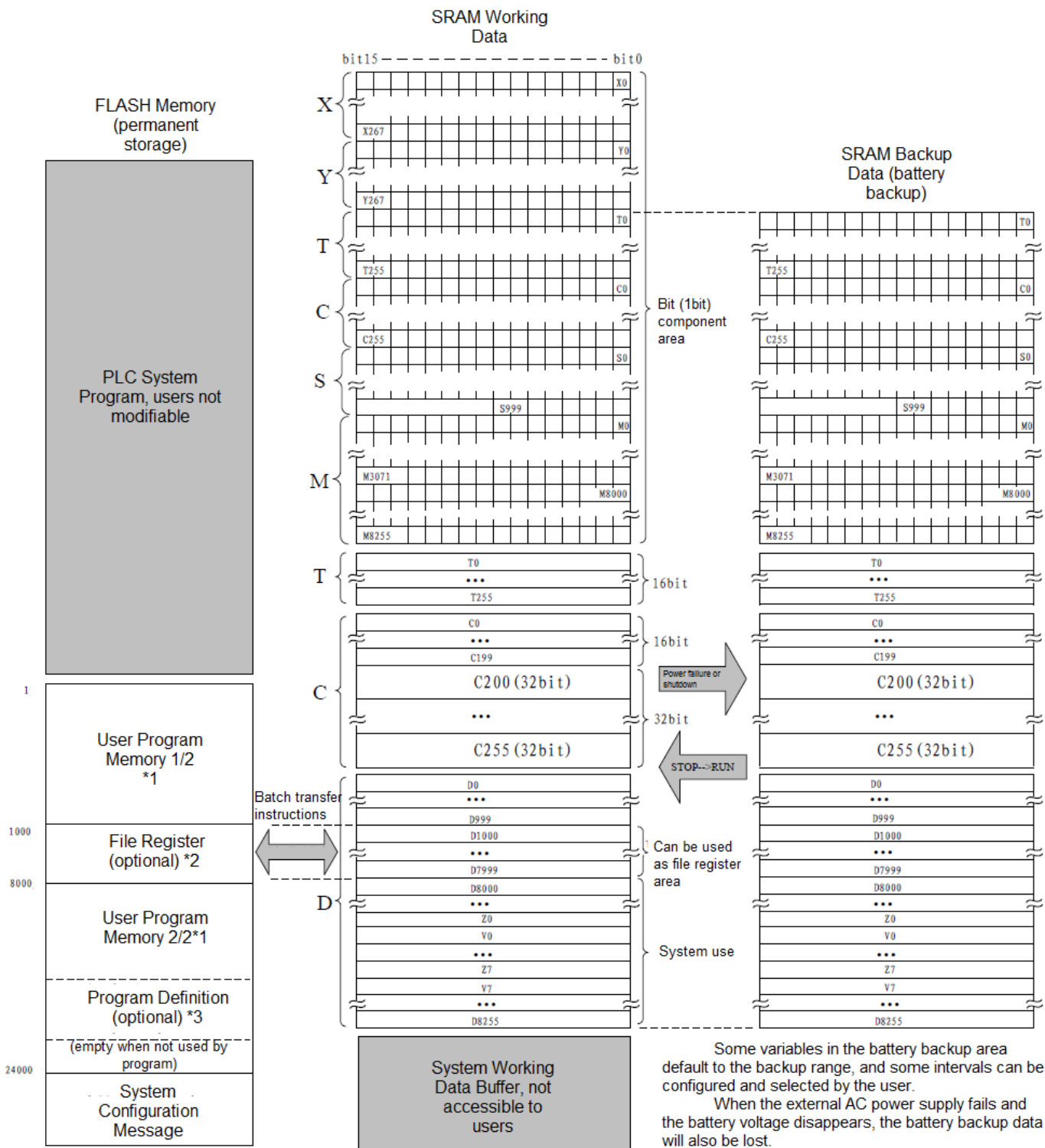
Floating-point calculation refers to the operation that involves floating points. This type of operation is often accompanied with approximation or rounding-up due to inability to express accurately. When there are simulating signal processes and operations in a PLC, floating-point may be utilized.

### Memory structure of H1U/H2U series PLC

For the system of a microcomputer or a single-chip computer, in addition to the CPU core, various memories are the main components. There are several kinds of memory in H1U/H2U series PLC:

Type	Purpose	Characteristics
FLASH	Save system program	Permanent storage
	Save user program	Permanent storage, unless deleted manually or updated by downloads
	File register data	Permanent storage, unless deleted or edited manually
SRAM	Used to store PLC soft components and working data	When is charged by battery, data will remain even when there is an external power failure

As described earlier, soft components in a PLC consist of "bit element" (contact component), "word element" in 16-bit (such as in register D, counter C, timer T, and etc), and "2 word elements" in 32-bit (some high-speed counter C). The organization in a PLC is as follows:



Some variables in the battery backup area default to the backup range, and some intervals can be configured and selected by the user.  
 When the external AC power supply fails and the battery voltage disappears, the battery backup data will also be lost.

\*1 User program memory has the maximum of 24K Word. File register can be automatically avoided when saving user programs.  
 \*2 File register memory can be defined with a maximum memory of 7K (Word), which will occupy user program space.  
 \*3 User program definition memory is closely connected to the ladder diagram program. The size of space is determined by definition contents, and it shares user program space



## Chapter 2 How to use H1U/H2U series PLC

---

### Contents

2.1 PLC System Requirements .....	- 33 -
2.2 Programming and User Program Download .....	- 34 -
2.3 Connecting to HMI.....	- 34 -

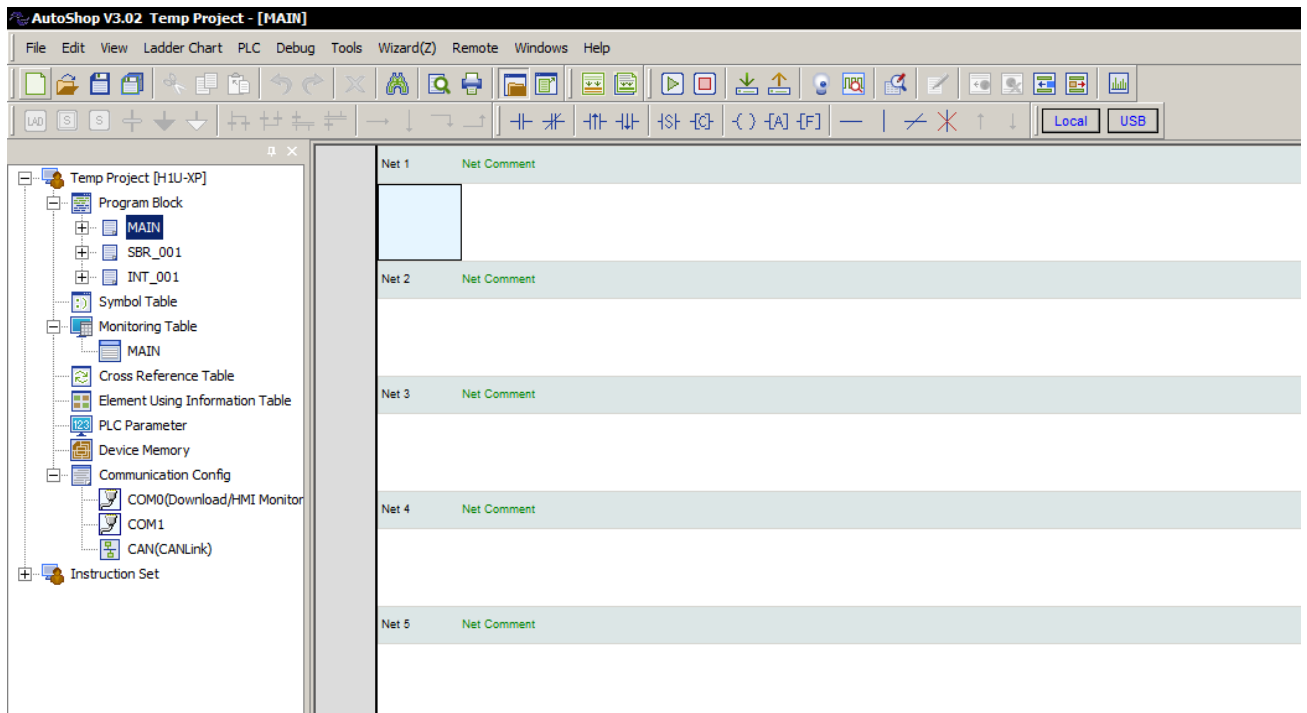
## Chapter 2 How to use H1U/H2U series PLC

### 2.1 PLC System Requirements



Item	Configuration Requirements
Personal computer x 1	Main frequency of PENTIUM is 100 MHz or above; the memory is above 256MB; the PC should have: DB9 type RS232 serial communication port or USB port Windows 7\8\10 operating system free hard disk space is not less than 200MB
AutoShop programming software	The AutoShop software developed by Inovance Control Technology Co., Ltd. is installed on PC and used for writing, downloading, monitoring and debugging of user programs. The programming environment of other compatible models can also be used.
PLC main module	One H1U/H2U Series PLC main module. Expansion modules may be prepared if required
Download cable x 1	A cable with RS232-Mini DIN8 pin, used for downloading, debugging, and monitoring user programs. It may also be used for connecting HMI. For PC without the DB9 RS232 serial communication port, USB-Mini DIN dedicated download cable may be prepared.
Power supply wiring and others	For the power for PLC power supply, common tools such as wires, switches, screwdrivers, etc. can be prepared as needed.







AutoShop programming software is a programming software written and developed by Inovance. Under the programming environment, the functions of programming, downloading, and monitoring user programs can be performed in the H1U\H2U Series PLC.



## 2.2 Programming and User Program Download

The AutoShop environment provides programming languages such as ladder diagram, SFC, instruction list, etc. Users can choose to use their familiar programming language for programming and design the program according to the control technology requirements of the PLC application system. During the programming process, you can press   to compile at any time to check and correct programming errors in time.

After the program design is completed, when the PLC and the computer are connected normally, and the power is on, press  to download the user program. After the program is downloaded, turn the RUN/STOP switch on the PLC to the "RUN" position, and the PLC can start running User program.

When the PLC is running the user program, press the   key to stop and run the command operation; press  to monitor the status and readings of various relays and register D in the PLC, which are displayed on the current programming screen, which facilitates program debugging.

## 2.3 Connecting to HMI

H1U/H2U series PLC provides MODBUS protocol, and supports the monitoring protocol of FX2N/3U series PLC. Therefore, currently commercially available HMI products can basically be used with H1U/H2U series PLCs, including connecting cables, which can be purchased from the market.

For a detailed description of the types and usage of protocols supported by H1U/H2U series PLC's, please refer to the explanation of RS communication commands.



---

## Chapter 3 Device elements description

### Contents

3.1 Input Relay X .....	- 36 -
3.2 Output Relay Y .....	- 37 -
3.3 Auxiliary Relay M .....	- 38 -
3.4 Status Relay S .....	- 40 -
3.5 Timer T .....	- 40 -
3.6 Counter C .....	- 42 -
3.7 Register D .....	- 49 -
3.8 Sub-program and Interruption Indicator P, I .....	- 53 -
3.9 Constant K, H .....	- 57 -
3.10 Controller soft component specifications .....	- 58 -

## Chapter 3 – Device elements description

The following table lists types of elements supported by H1U\H2U:

SN.	Component Type	Features and classification
1	Input Relays X	Corresponding to the PLC's bit element X for input
2	Output Relays Y	Corresponding to the PLC's bit element Y for output
3	Auxiliary relays	Ordinary auxiliary relay M
		System special auxiliary relay M
		System special auxiliary relay SM
4	Status relay	Status flag bit element S for step control
5	Timers T	16-bit timers T with a step length of 1 ms, 10 ms, 100 ms
6	Counters C	16-bit/32-bit incremental/decremental counter C
		32-bit high-speed counter C
7	Data registers D	Data register D
		Indirect addressing data registers V and Z
8	Pointers P, I	Marker/Jump pointer P
		Subprogram SBR
		Interrupt subroutine I, with high-speed input, timing, counting and other interrupts
9	Constants K, H	Binary, decimal, hexadecimal, floating point, etc.

### 3.1 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.

The controller's counter signal, external interrupt signal, pulse capture and other functions are input through the X0 ~ X7 ports. The input and output points of some product models are shown in the following table:

Model	Input	Output
H2U-1616MR/T	X000-X017	Y000-Y017
H2U-3624MR/T	X000-X043	Y000-Y027
H2U-3232MR/T	X000-X037	Y000-Y037
H2U-6464MR/T	X000-X077	Y000-Y077
H1U-0806MR/T	X000-X07	Y000-Y005
H1U-1410MR/T	X000-X015	Y000-Y011

When the local expansion module is connected, the number of the X port on the expansion module is numbered backwards according to the number of the X port on the main module. For example, when the main module is H2U-1616MR, it is necessary to connect the H2U-1600EX expansion module. Because the last X port number of the main module is X17, the access number of X of the expansion module during programming is X20~X37.



Note that the number of the expansion module always starts from 0 in the octal unit. For example, when the main module is H2U-3624MR, the last X port number is X43, and the access number of the X of the expansion module during programming is X50~X67, that is, the vacant port numbers of X44~X47 on the main module are discarded. The Y port on the expansion module also takes the same approach.

The remote digital expansion module must use FROM/TO instructions to access point X. For details, please refer to the introduction of FROM/TO instructions.

## 3.2 Output Relay Y

Output relay is a soft element directly connected to the hardware port of the external user control device, and logically corresponds to the physical output port of the PLC. After the PLC scans the user program each time, the component status of the Y relay will be transmitted to the hardware port of the PLC. 0 means the output port is open; 1 means the output port is closed.

Y relay number is identified by Y0, Y1...Y7, Y10, Y11... and other symbols, and its serial number is numbered in octal. Y relay components can be used unlimited times in the user program; In terms of hardware, according to different output components, it can be divided into relay type, transistor type, etc.; if there are output expansion module ports, they are numbered in sequence starting from the main module.

When the local expansion module is connected, the number of the Y port on the expansion module is numbered backwards according to the number of the Y port on the main module. For example, when the main module is H2U-1616MR, now it is necessary to connect the H2U-0016EYR expansion module. Since the last Y port number of the main module is Y17, the access number of X of the expansion module during programming is Y20~Y37.

Note: The port number of the expansion module always starts from 0 in the octal unit.

The remote digital expansion module must use FROM/TO instructions to access point Y.

### 3.3 Auxiliary Relay M

The auxiliary relay M element is used as an intermediate variable in the execution of the user program, just like the auxiliary relay in the actual electronic control system, used for the transmission of status information, and multiple M variables can also be used as word variables, M variables and external ports are not directly related, but you can copy X to M through program statements, or copy M to Y to connect with the outside world. M variable can be used unlimited times.

The auxiliary relay M is identified by symbols such as M0, M1...M8255, and its serial number is numbered in decimal system. The variables above M8000 are system-specific variables, used for the interaction between PLC user program and system status; some M variables also have the feature of power-off storage.

H1U Model:

Total number of M register	General	Latched	Special
1792 points	M0-M383 384 points (*1)	M384-M3071 2688 points (*3)	M8000-M8511 512 points

H2U Model:

Total number of M register	General	Latched	Latched dedicated	Special
3328 points	M0-M499 500 points (*1)	M500-M1023 524 points (*2)	M1024-M3071 2048 points (*3)	M8000-M8511 512 points

(\*1) Volatile areas. Using parameter settings, it can be changed to a non-volatile area.

(\*2) Non-volatile areas. Using parameter settings, it can be changed to a volatile area.

(\*3) The non-volatile area cannot be changed with parameters.

The area allocation of the general auxiliary relay and the auxiliary relay for power failure in the programmable controller can be adjusted by parameter setting.

There are many special auxiliary relays in the programmable controller. (See the table of system special components). Each of these special auxiliary relays has its own specific function, which can be divided into the following two categories:

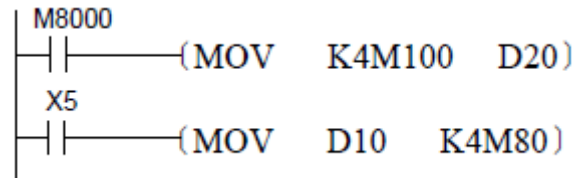
- The special auxiliary relay of the contact utilization type automatically drives the coil for the PLC system, and the user program can only be read and used, such as:
  - M8000: Run monitor (turned on during running), often used before commands that require drive signals.
  - M8002: Initial pulse (only turned on at the beginning of operation), often used to execute the initialization command only once.
  - M8012: 100ms clock pulse, used to generate a signal that flips at a fixed interval.
- Coil-driven special auxiliary relay, which drives the coil for the user program, used to control the working status and execution mode of the PLC, such as:
  - M8030: Battery LED lights off instruction
  - M8033: Keep output when stopped

M8034: All outputs are prohibited

M8039: Constant scan

Please note that there are two situations that are valid during driving and valid after the END instruction is executed; users cannot use special auxiliary relays that have not been defined.

- The continuous M variables can be accessed (read or written) by byte or word, for example:



K4M100 will group these 16 units from M100, M101, M102, ..., to M115 into a unit of 1 Word to perform readout operation (M100 will become bit0...M115 will become bit15). This can help to increase programming efficiency.

## 3.4 Status Relay S

The state relay S is used for the design and execution of the step program, and the STL step instruction is used to control the transfer of the step state S, simplifying the programming design.

If the STL programming method is not used, S can be used as an ordinary bit component, just like an M variable. State S variables are identified by symbols such as S0, S1...S999, and their serial numbers are numbered in decimal system. Some S variables have a power-down save function. The following table:

H1U Model:

For retentive use
S0-S999
1000 points

H2U Model:

General use Latched	Latched	Alarm Used
S0-S499 500 points (*1)	S0-S9 10 points	S500-S899 400 points (*2)
		S900-S999 100 points (*3)

(\*1) No power failure retention areas. Using parameter settings, it can be changed to a power failure retention area.

(\*2) Power failure retention areas. Using parameter settings, it can be changed to a non-latched area.

(\*3) Power failure areas, cannot be modified with parameters.

## 3.5 Timer T

The timer is used to complete the timing function. Each timer contains a coil, a contact, and a counting time value register. When the timer coil is "powered" (energy flow is valid), the timer starts timing. If the timing value reaches the preset time value, its contact will act, NO contact is closed, and NC contact is disconnected. If the coil is "lost" (power flow is invalid), the contact of the timer returns to the initial state, and the timer value is automatically cleared. There are also some timers with features such as accumulation and non-volatile, and they will maintain the value before power-off after power-on again.

Timer T is identified by symbols such as T0, T1...T255, and its serial number is numbered in decimal system. The timer has different timing steps, such as 1ms, 10ms, 100ms, etc., some of which have power-down retention characteristics. H1U and H2U are described in the following table:

100ms type 0.1~3276.7s	10ms type 0.01~327.67s	1ms accumulative type 0.001~32.767s	100ms accumulative type 0.1~3276.7s
T0~T199, 200pts in total Where T192~T199 can be used for interruption/subroutine	T200~T245 46pts in total	T246~T249 4pts in total Execution interrupt retention	T250~T255 6 points Used as retentive

- A timer number that is not used as a timer can also be used as a data register for storing values.
- The timer accumulates 1ms, 10ms, 100ms, etc. clock pulses in the programmable controller. When the timing time reaches the set value, its contacts can only operate when the coil instruction or END instruction

is executed.

- Use the constant (K) in the program memory as the set value. The contents of the data register (D) can also be used for indirect designation. Note that the content of D must be set before starting the timing. After the count starts, the data change of D can only take effect the next time the timing is started.
- From the start of the coil driving the timer to the contact action of the timer, the possible timing lengths are described as follows:

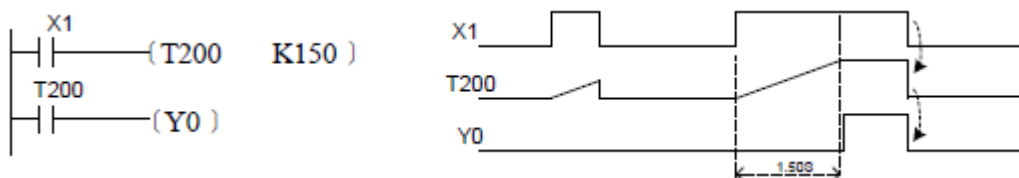
The longest case is  $(T+T_0+a)$ , where: T is the set timing time;  $T_0$  is the program scan execution time; a is the timing step of the timer.

The shortest case is  $(T-a)$ .

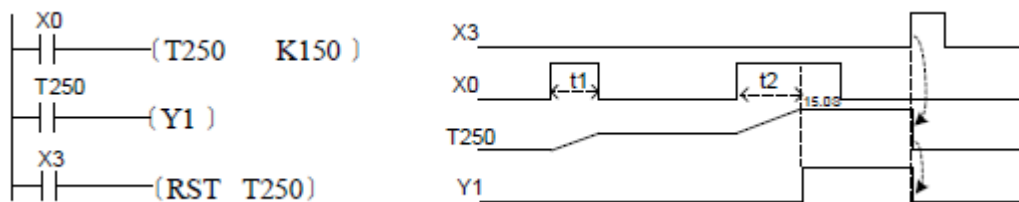
If the timer's contact command is before the coil command, the least ideal timing length is  $(T+2T_0)$ .

- Using the b contact of the timer, it is possible to realize delayed disconnection and self-oscillation output signals.
- PLC also provides special timer instructions, such as TTMR, STMR, etc., please refer to the description of the corresponding instructions.

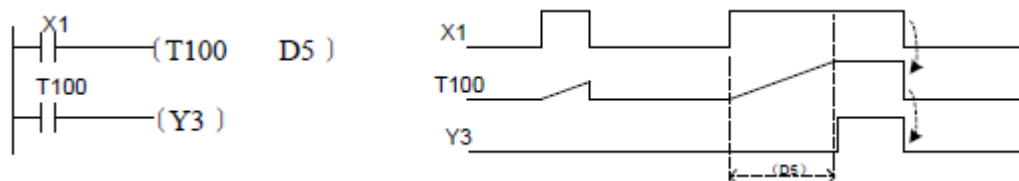
[Use example 1]: The ordinary timer T200 is a counter with a step length of 10ms, and the actual action delay is  $150 \times 10\text{ms} = 1500\text{ms}$ , that is, 1.50s, and the action principle is:



[Example 2]: For the timer T250 with power failure retention, the drive signal is OFF, or when the PLC is powered off, its internal count value remains unchanged, and the next time the drive signal is ON, the timing will continue until the time is met. When the value is set, the output contact is closed. When the timer coil is reset, the timer value is cleared and the output contact is disconnected, as shown in the figure below. Because the counter T250 has a step length of 100ms, the actual action delay accumulates to  $150 \times 100\text{ms} = 15000\text{ms}$ , that is, 15.0s, that is, the time  $(t_1 + t_2)$  in the figure:



[Example 3]: The set action value of the timer can be set by register D, as shown in the figure below. (During counter timing, if the value in register D changes, it will take effect the next time the timer starts)



## 3.6 Counter C

The counter is used to complete the counting function. Each counter contains coils, contacts, and timing data value registers. Whenever the driving signal of the counter coil changes from OFF→ON, the counter reading increases by 1, if the timing value reaches the preset time value, the NO contact is closed, and the NC contact is opened; if the timer value is cleared, the output a contact will be opened and the NC contact will be closed. Some counters have the characteristics of power-down retention, accumulation, etc., and maintain the value before power-down after power-on again.

The counters are identified by C0, C1, ... C255, and are numbered sequentially in decimal.

The counters have 16-bit and 32-bit widths; there are one-way counting type, up-down counting type, two-phase counting type, etc. The count value of some counters also has the feature of power-down retention. Select the appropriate counter according to your needs.

H1U Model:

16-bit sequence/counter 0 ~ 32,767 counts		32-bit sequence/counter -2,147,483,648 ~ +2,147,483,647		
General	Latched	General	Latched dedicated	High-speed counter
C0~C15 16 points (*1)	C16~C199 184 points (*3)	C200~C219 20 points (*1)	C220~C234 15 points (*3)	C235~C255 (*3)

H2U Model:

16-bit sequence/counter 0 ~ 32,767 counts		32-bit sequence/counter -2,147,483,648 ~ +2,147,483,647		
General	Latched	General	Latched dedicated	High-speed counter
C0~C99 100 points (*1)	C100~C199 100 points (*2)	C200~C219 20 points (*1)	C220~C234 15 points (*2)	C235~C255 (*2)

(\*1) Non-maintenance area. It can be changed to the latched area by setting parameters.

(\*2) Power failure latched area. It can be changed to non-latched area by setting parameters.

(\*3) Power failure latched area. It cannot be changed by parameter setting.

The counter number that is not used as a counter can be used as a data register for data memory. For 32-bit counters C200~C234, special auxiliary relays M8200~M8234 are used as up/down counter switching control, see the following table:

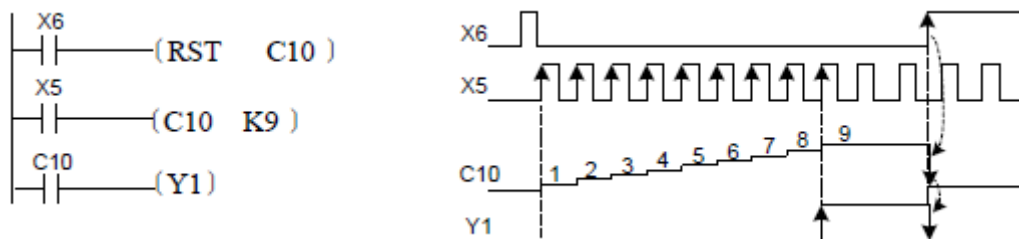
Counter No.	Direction Switch	Counter No.	Direction Switch	Counter No.	Direction Switch	Counter No.	Direction Switch
C200	M8200	C209	M8209	C218	M8218	C226	M8226
C201	M8201	C210	M8210	C219	M8219	C227	M8227
C202	M8202	C211	M8211	—	—	C228	M8228
C203	M8203	C212	M8212	C220	M8220	C229	M8229
C204	M8204	C213	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

16-bit counter and 32-bit counter's characteristics are illustrated in the chart below. It can be used separately according to the switching of counting directions and the counting range requirements.

Item	16-bit counter	32-bit counter
Counting direction	In sequence	Sequence/Reverse (see above)
Setting	1 ~ 32,767	-2,147,483,648 ~ +2,147,483,647
Designated setting	Constant K or data register	Constant K, 2, D data registers may be used
Current value changes	No change sequence	Changes after sequence (loop counter)
Output contact	Remain action after sequence	Remain action in sequence, and reset when reversed
Reset	When executing the RST command, the counter's current value returns to zero and the output contact resets	
Current value register	16-bit	32-bit

### 16-bit counter

- The distribution between common counter and the power failure retention status can be modified and configured through system parameter settings.
- For a 16-bit counter, the valid setting is between K1 ~ K32,767 (decimal constant); Setup value K0 and K1 carry the same effect, where the output contact will activate when first counting starts. See following illustration:



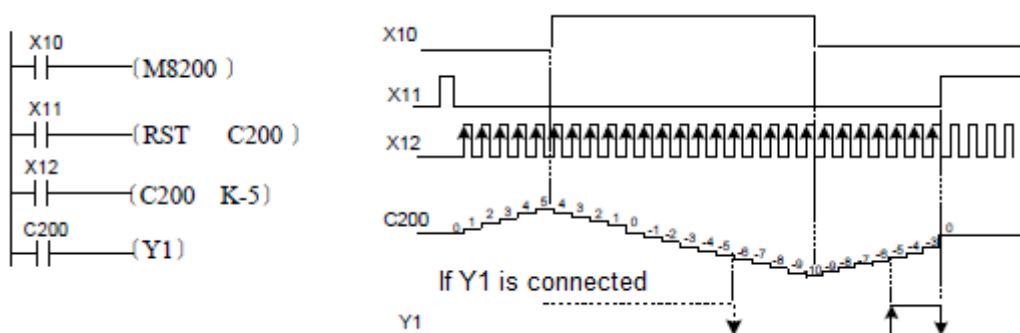
Every time when C10 coil is activated by the counting input X5, the counter's present value increases. When carrying out the ninth coil command, output contact activates. Even if the counting input X5 activates again afterward, the present value of the counter remains the same. If reset input X6 is ON, the RST command will be executed, count's present value will be cleared to 0, and output contact resets.

- Counting settings: Not only it uses the above-mentioned K constant, it can also be assigned by the data register. Use the above illustration as an example, D20 is assigned. If D20's content is 9, it has the same setting as K9.
- When using the MOV command to write these data into the current value register, the value saved in the register will become the new setting value after the output coil has been connected.
- For other commonly used counter, count value will be cleared if the programmable controller's power was disconnected. Counters with the power failure preservation feature will be able to store the count value before the power failure. Therefore, the counter can continue counting from the last accumulated value.

### 32-bit counter

- For 32-bit counters, the valid range of up/down counting setting value is -2,147,483,648~+2,147,483,647 (decimal constant), which can be set with the content of constant K or data register D. Use special

auxiliary relays M8200~M8234 to specify the direction of up/down counting. If CXXX drives M8XXX to set 1, it will count down, and if it is cleared when it is not driven, it will count up.



Increase or decrease of the current value has nothing to do with the action of the output contact, but if the counting starts from 2,147,483,647 and after another pulse is input, it becomes -2,147,483,648. Similarly, if you start counting down from -2,147,483,648 and then input a pulse, it will become 2,147,483,647. (This kind of action is called ring counting); if the reset input X11 is ON, the RST instruction is executed, the current value of the counter becomes 0, and the output contact is also reset.

- When using a counter for retention of power failure, the current value of the counter, output contact action, and reset status are retained.
- The 32-bit counter can also be used as a 32-bit data register. However, a 32-bit counter cannot be used as a device in a 16-bit application instruction.
- When data above the set value is written into the current value data register with the DMOV instruction, etc., the counting can be continued during the subsequent counting input, and the contact will not change.
- For a 16-bit counter, the highest bit (bit 15) is the sign bit, and the processed data is in the range of 0 to 32,767, that is, it can only be a positive number.
- For a 32-bit counter, the highest bit (bit 31, the highest bit of the high byte) is the sign bit, and the processed data range is from -2,147,483,648 to 2,147,483,647.

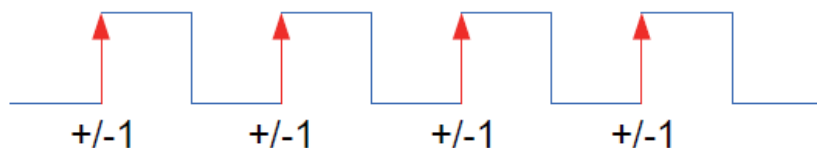
### High-speed counter

H1U/H2U Series PLC's built-in high-speed counter is illustrated as the following chart. The counter's serial numbers (C) are assigned in inputs X000~X007.

For X input ports that are not being used by high-speed counter can be used as a regular input relays in sequential control programs. Besides, high-speed counter serial numbers that are not being used by the high-speed counter can also be used by the 32-bit data registers for the purpose of storing data.

- High-speed counters have the following types:

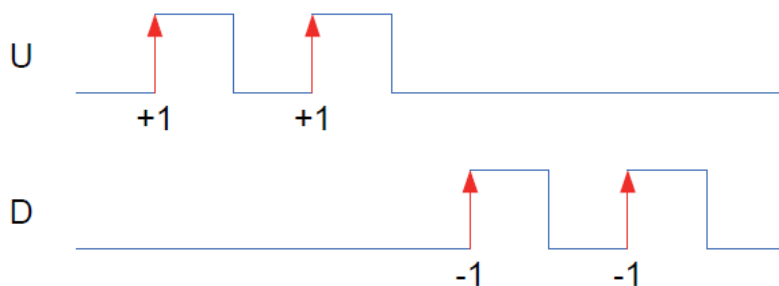
1) Single-phase unidirectional counter: With only one pulse signal counting input, the corresponding M element for up/down counter control decides on counting up/down. Some counters also have hardware reset and start up signal input ports.



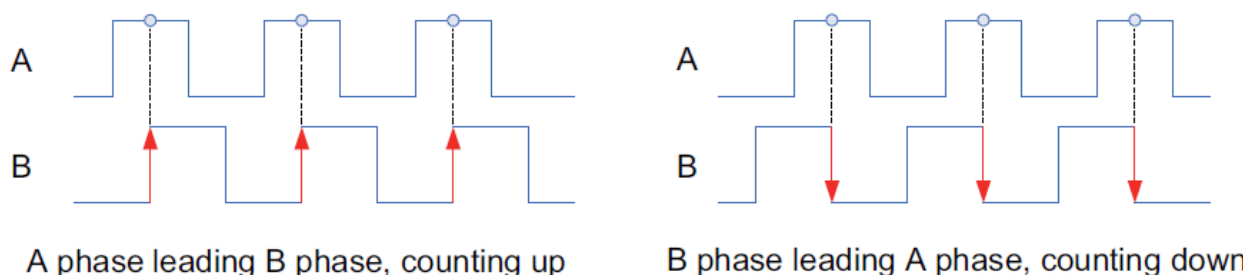
2) Single-phase bidirectional counter: Two pulse signal counting inputs function as an up/down counter



pulse input respectively. Some counters also have hardware reset and start up signal inputs. The up/down counting state of a counter is monitored by reading the special M element for up/down counting state.



3) 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.



Two-phase encoder exports A phase and B phase that has a 90° phase difference from each other. Base on this, the high-speed counter automatically performs the up/down counting.

- Through special variables setting, AB phase counting with 4 times of frequency can be carried out. It can also provide counting accuracy.
- Some counters still have signal input port for hardware reset, start, and stop.

Item	Single-phase single counting input	Single-phase double counting input	Two-phase single/double counting input
Designating counting direction	C235-C245 performs up/down counting based on M8235-M8245's start status.	In response to up counting input or down counting input, the counter automatically up/down counts.	When the A phase input is ON, the B phase input will count up when it is OFF→ON, and it will count down when it is ON→OFF.
Counting direction monitoring	-	Through monitoring M8246-M8255, one can observe the up (OFF)/down (ON) count status.	

## Auxiliary relay switch for up/down counts

Types	Counter	UP/DN designation
Single-phase single counting input	C235	M8235
	C236	M8236
	C237	M8237
	C238	M8238
	C239	M8239
	C240	M8240
	C241	M8241
	C242	M8242
	C243	M8243
	C244	M8244
	C245	M8245
	C235	M8235

## Auxiliary relay for counting direction monitoring

Types	Counter	UP/DN Monitoring
Single-phase dual counting input	C246	M8246
	C247	M8247
	C248	M8248
	C249	M8249
	C250	M8250
Double-phase dual counting input	C251	M8251
	C252	M8252
	C253	M8253
	C254	M8254
	C255	M8255

- High-speed counter's serial numbers should be used in pairs with the corresponding X ports. It means that after assigning high-speed counter CXXX, the corresponding X input port should also be designated. So that X ports will not be used repeatedly when programming. Definitions are as follows:

Assigned Input	Single-phase single counting input										
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
X000	U/D						U/D			U/D	
X001		U/D					R			R	
X002			U/D					U/D			U/D
X003				U/D				R			R
X004					U/D				U/D		
X005						U/D			R		
X006										S	
X007											S

## Double count and A/B phase counter:

Assigned Input	Single-phase single counting input					A/B Phase Count				
	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X000	U	U		U		A	A		A	
X001	D	D		D		B	B		B	
X002		R		R			R		R	
X003			U		U			A		A
X004			D		D			B		B
X005			R		R			R		R
X006				S					S	
X007					S					S

**[Note]**

1) M8166 is the flag of prohibiting to reset high-speed counter via X port (except edge interrupt reset), which is "Do not prohibit" by default.

2) M8166 is the flag of prohibiting to reset high-speed counter via X port (no matter it is level reset or edge interrupt reset), which is "Do not prohibit" by default.

U: UP counter input; D: DOWN counter input; A: A-phase pulse input; B: B-phase pulse input; R: reset counter input; S: start counter input

The X0 and X1 input points of H1U-XP model single-phase input can reach 60kHz, the X2~X5 input points can reach 10kHz, and the total frequency does not exceed 70kHz.

The X0~X5 input points of H2U-XP model single -phase input can reach 100kHz.

The maximum input frequency of the AB phase should be reduced by half, if the quadrature frequency is used, it will be reduced to a quarter. For example, the C251 of the H2U model is the AB phase count, occupying X0 and X1, and the maximum input frequency is 30kHz when the frequency is doubled, and the maximum input frequency is 15 kHz at quadrature frequency (M8195 is ON).

The input terminal that is not used as a high-speed counter can be used as a general input.

**[Example 1 of reading table]**

C235 in the table is a single-phase single-input count, using X0 input port, no need to interrupt reset and interrupt start port;

If you use the C235 counter, the X0 input port is used by default, you can no longer use C241, C244, C246, C247, C249, C251, C252, C254 and interrupt I00 port or M8170 (pulse capture), for counters, interrupts or pulse captures, because X0 port is used for pulse capture, which creates a port conflict.

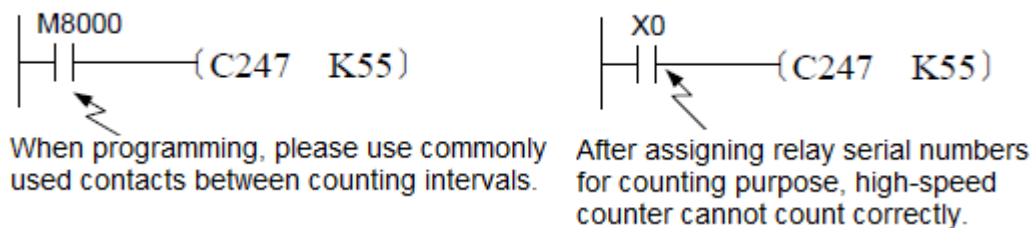
**[Example of reading table 2]**

In the table, C254 is a double-phase dual-input counter, namely AB-phase counter, X0 port is used as A-phase pulse input, X1 port is used as B-phase pulse input, X2 port is used as interrupt reset input, and X6 port is used as interrupt start input, if you use the C254 counter, that is, the X0, X1, X2, and X6 input ports are used by default. Counters, interrupt ports, or pulse captures related to these ports can no longer be used.

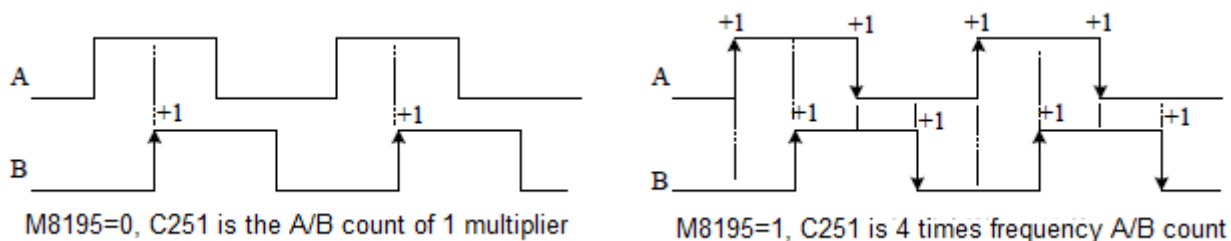
Counter instructions:

- The high-speed counter executes actions based on specific inputs and uses interrupt processing to perform high-speed counting actions on the transition edges of related signals, so it has nothing to do with the PLC scan time.
- When the current value of the high-speed counter reaches the set value, if you want to perform output processing immediately, please use high-speed pulse comparison commands HSCS, HSCR, HSZ and other application instructions. For details, see the instruction explanation.
- When the current value of the high-speed counter reaches the set value, if you want to perform some logic processing immediately, you can use the high-speed counting interrupt, use the high-speed pulse comparison instruction HSCS, and specify the operation of the instruction as the I0X0 interrupt (where X = 1~6 interrupt number ), of course, the subroutine corresponding to the interrupt number must be written.

- For the coil drive contact of the high-speed counter, use a contact that is always on during high-speed counting.



- When programming the coil for the high-speed counter, the corresponding input relay's input filter will automatically become 20s (X000, X001) or 50s (X002-X005) (initial value is 10ms). Moreover, the input filter of the input relay that is not being used by high-speed counter input will remain its initial value of 10ms.
- A/B phase high-speed counter C251~C255 has two frequency modes: double and four times. They can be configured separately by the special register M8195~M8199. See following illustration:



- High-speed counters all use hardware counting method, which has no soft component limit over the input pulses overall frequency. Two-phase high-speed counters signal occupies two pulse input ports, and its equivalent pulse number toward PLC is calculated twice. If C251~C255 counter's A/B input uses four times of frequency, it is in the soft component counting mode and the high-speed input frequency drops to 25kHz.
- Since both high-speed X and Y counting pulse output are processed in interrupt mode, when the number of signal channels are large, the execution speed of the program may be affected. When the counter inputs the signal, the frequency used is lower than the above frequency. If a signal exceeding this frequency is input, a watchdog timer (WDT) error may occur.
- When multiple high-speed counters are used at the same time or high-speed counters and (SPD), (HSCS), (HSCR), (HSZ), (PLSY), (PLSR) and other high-speed outputs are used simultaneously, the high-speed counter needs de-rating, the frequency range of single-phase counters up to 10KHz and dual-phase counters of 5kHz.

## 3.7 Register D

### Data Register D

Registers are used for the calculation and storage of data, such as timer, counter, simulative variable parameters, etc. Every register's bandwidth is 16-bit. If 32-bit command is used, it will automatically combine two neighbouring registers into 32-bit register. Registers with lower addresses are low bytes, higher addresses are high bytes.

Among the commands of H2U Series PLC, data are processed according to its signs. For 16-bit registers, bit 15 is the sign digit. 0 represents positive while 1 represents negative values. For 32-bit registers, bit 15 in high byte is the sign digit. The value ranges are -32,768 ~ +32,767.

When 32-bit data are processed, two neighbouring D registers can be combined into 32-bit 2 words. For example, when accessing D100 in 32-bit format, turn the high address D101 register into high byte. Meanwhile, use high byte, bit15 as the sign digit for the 2 Word, so that the values that can be processed are -2,147,483,648 ~ 2,147,483,647.

Registers use D0, D1...D9, 999 as their symbol and the serial numbers are assigned using the decimal system.

H1U Model:

General	Latched	Special	For address change
D0~D127 128 points (*1)	D128~D7999 7872 points (*3)	D8000~D8511 512 points	V0~V7 Z0~Z7

H2U Model:

General	Latched	Latched dedicated	Special	For address change
D0~D199 200 points (*1)	D200~D511 312 points (*2)	D512~D7999 7488 points (*3)	D8000~D8511 512 points	V0~V7 Z0~Z7

(\*1) No power failure retention areas. It can be changed to the latched area by setting of parameters.

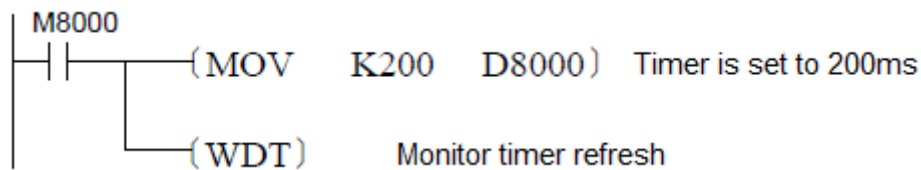
(\*2) Power failure retention areas. It can be changed to non-latched area by setting of parameters.

(\*3) It is not possible to change the characteristics of power failure retention by setting of parameters.

Note: H1U-XP (version 26128 and above) and H2U-XP (version 24137 and above) with CANlink3.0 function, because this function occupies the D component of D7000-D7999, users are not allowed to use D7000-D7999

- Express 32-bit data using two neighbouring data registers. (High bits are bigger numbers; low bits are smaller numbers. In the address-changeable registers, V is high bit and Z is low bit). If 32-bit is assigned as low bit (i.e. D0), the high bit serial numbers (i.e. D1) that are assigned after, will automatically be occupied. Low bit can be assigned with any odd or even soft component serial numbers. When taking into consideration for peripheral device monitoring, it is recommended to use even soft component serial numbers with low bit.
  - Once data is written into the data register, if there is no other data entry, then there will be no change. However, when switching from RUN to STOP, or when there is a power failure, all data will be cleared to 0. (data may be retained if auxiliary relay M8033 is activated). Also, data registers with power failure preservation feature can also retain data contents under the circumstances of RUN/STOP and power outage.

- Using the system's parameter configuration function to change D register's use between common usage and power failure preservation usage; When turning the data register with power failure preservation into a common register, please use RST or ZRST command in the initial stage of program start to clear the content.
- Under the situations where PLC quick connect or online connect is used, a part of the data registers will be used by default.
- The special purpose data register refers to the data written for a specific purpose, used to realize some special functions of the controller, and can be regarded as a special unit for data interaction between the user program and the PLC system program. For example, in D8000, the time of the watchdog timer is initially set by the system ROM. To change it, use the MOV transfer command to write the target time in D8000.



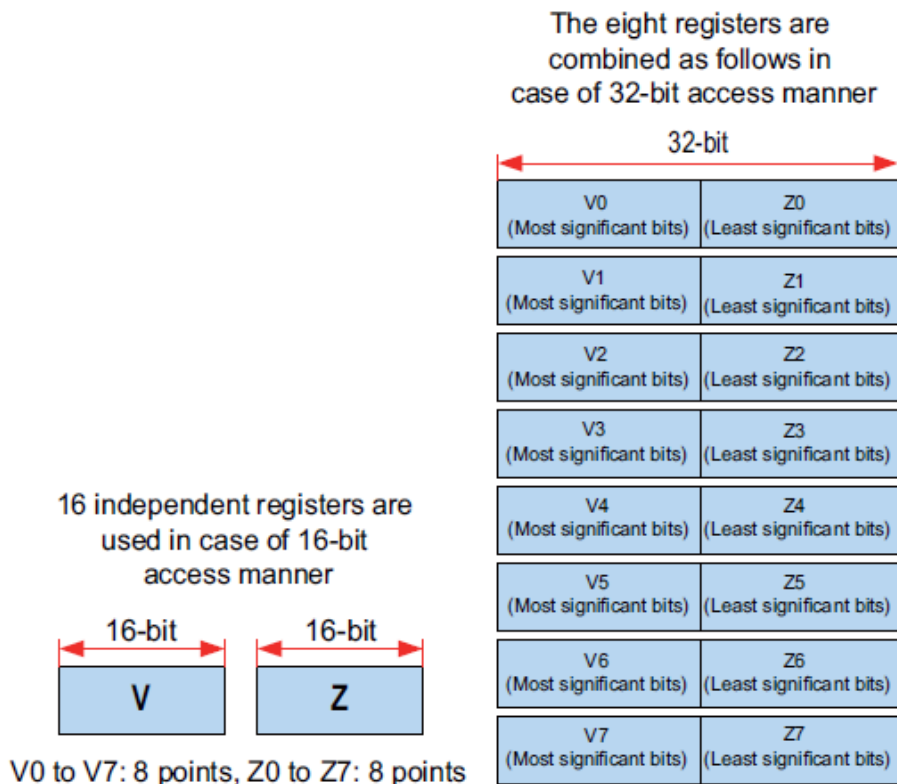
There are also other special D registers that are used as parameter memory for system working status. Refer to these registers to determine operation parameters.

- About the power failure preservation feature in the special data registers, please refer to “Special Register Descriptions”.
- Data registers can be used to process all kinds of numerical data. It is through these registers all kinds of controls can be implemented. For instance, it can be used to configure timer and counter settings, or all types of data calculations. Commands that support the usage of D registers will be described and explained later on in details.

### Data register V, Z

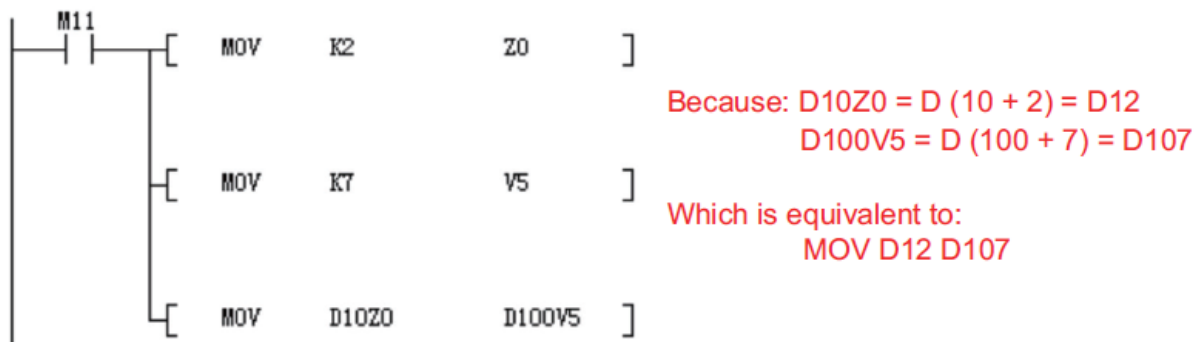
Index register V, Z is the same as the common data register, which is a 16-bit data register for read/writing numerical data. There are a total of 16 data registers: V0~V7, Z0~Z7. The index register has the same use as the common data register, and it also can be used with other soft components (as the address offset variable). However, it should be noted that the soft component's serial number of basic sequential control commands or ladder diagram commands such as LD, AND, and OUT, cannot be combined and used with address-changeable registers.

V, Z registers can gain access use both 16-bit and 32-bit methods. See explanation below:

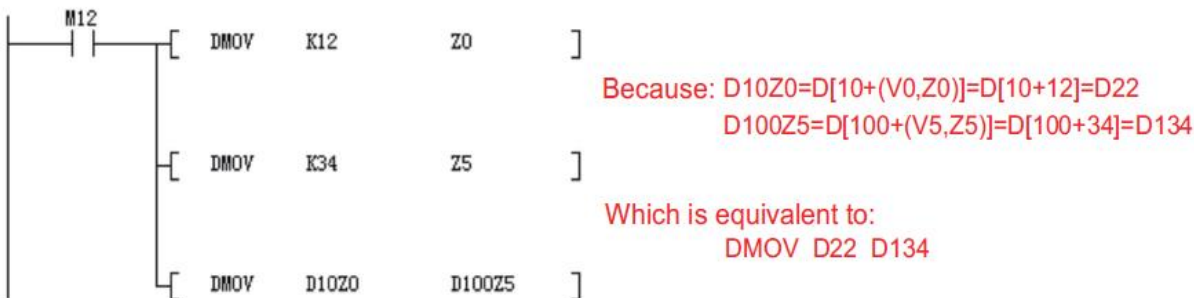


As usual, when processing the soft components in 32-bit application commands or data exceeding the 16-bit range, (32-bit register method), V (high bit), Z (low bit) will be accessed simultaneously. The assigned register names must be Z0~Z7. Even if V0~V7 on the high bit end is assigned, address change will not be allowed.

16-bit address change illustration:



32-bit address change illustration:



Special example of constant-based addressing index:



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

When V and Z intermediate address-searching method is used in loop instructions (V, Z variables change with the loop), the operation can be done for several data ranges or it may be used for search operation to simplify programming and increase command efficiency.



## 3.8 Sub-program and Interruption Pointer P, I

Pointer (P) is used to indicate the entry address of start program and the initial address of a sub-program; Pointer (I) is used to indicate the initial address of the interruption program. It's serial number is assigned using the decimal system. It is illustrated in the chart below:

H1U Model:

Branch	To end the jump	Input interrupt	Timing interrupt	High-speed counter interrupt	Multi-user interrupt (XP model does not have this function)
P0~P62; P64~P127 127 points  P128~P131: 4 points, encryption subroutine	P63 Total 1 point	I00x(X0) I10x(X1) I20x(X2) I30x(X3) I40x(X4) I50x(X5) x=1, rising edge interrupt x=0, falling edge interrupt 12 points (*1)	I010 I040 I020 I050 I030 I060 6 points	I010 I020 I030 I040 I050 I060 6 points	I507~I530 24 points

H2U Model:

Branch	To end the jump	Input interrupt	Timing interrupt	High-speed counter interrupt	Pulse complete interrupt	Multi-user interrupt (XP model does not have this function)
P0~P62; P64~P127 127 points	P63 1 point	I00x(X0) I10x(X1) I20x(X2) I30x(X3) I40x(X4) I50x(X5) x=1 rising edge interrupt x=0 trailing edge interrupt 12 points(*2)	I600 I700 I800 3 points	I010 I020 I030 I040 I050 I060 6 points	I502(Y0) I503(Y1) I504(Y2) I505(Y3) I506(Y4) 5 points	I507~I530 24 points

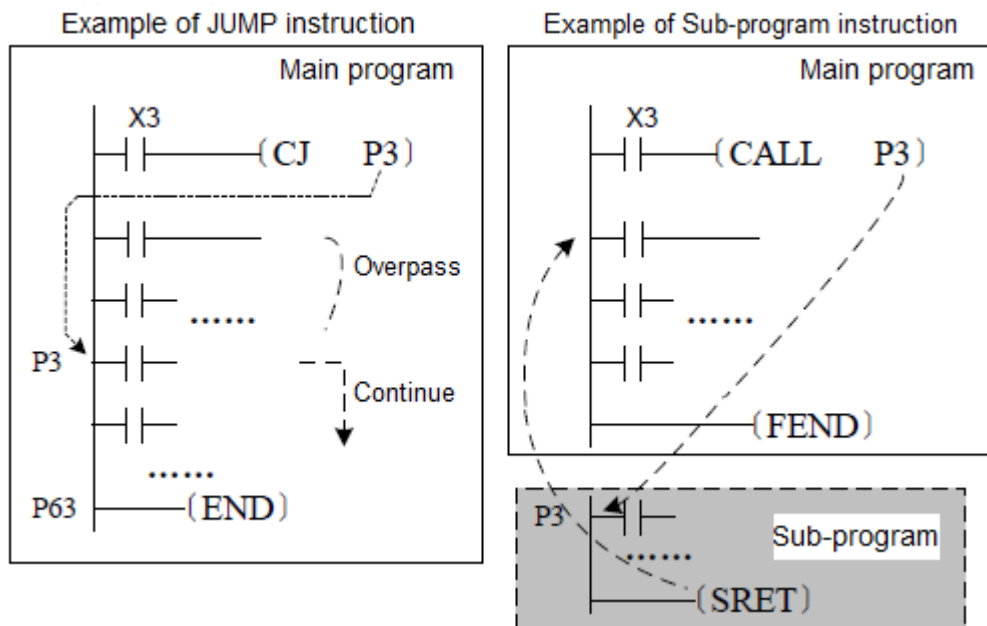
\*1: The number of allowable input interrupts of the enhanced function version has been expanded, please refer to Appendix 8.6 Enhanced Function Description.

\*2: Support 4 encrypted subroutines, subroutines can be encrypted individually, each subroutine has a maximum of 480 steps.

Because external input interruption, high-speed counting, pulse frequency measurement and other functions are input through the X0~X7 ports, the X ports used by these functions cannot be reused. Therefore, when using the input interrupt pointer, pay attention to the port function arrangement, check the input port number used by high-speed counter and pulse density command.

The use and difference of jump pointer (P) and subroutine pointer (P) are shown in the figure below. The instruction statement guided by the jump pointer (P) is still in the main program. It is only used to cross a part of

the instruction statement when the condition is met; but the subroutine pointer (P) is used for a section of subroutine. If the condition in the main program is met, CALL the subroutine. After the subroutine is executed (SRET), return to the next step of the original call (CALL) instruction to continue execution.



- One serial number is used for two types of P indicators. Do not duplicate when defining P indicator;
- P63 indicator is an indicator assigned with a specific number and used to point to the program's termination state END. Do NOT program P63.

Indicator (I) is used to indicate the starting address of the interrupt, and interrupt sub-program is used under the situation of "interrupt enable". The moment when signal requirement is satisfied, the PLC system will pause the main program's operation (record the pause point). Interrupt sub-program will be executed starting from the address indicated by the specific 'I' indicator, until the IRET command has been executed and the system has returned to the pause point in the main program and continue its operation. Because the PLC system processes the interruption signals based on their priorities, it will not be affected by the scanning time. The PLC system provides two types of interrupt for H1U. They are X input and counter interrupt. Five types of interruptions are provided for the H2U model. They are introduced as below (the first two include for H1U):

- 1) X input interrupt: the controller's X0~X5 can be configured individually as interrupt input ports. Under every port there are rising and falling edge interruptions which are recognized by their interrupt signals. For instance, "I100" signal represents the falling edge interrupt at X1 port, while "I101" signal represents the rising edge interrupt at X1 port.
- 2) Counter interrupt: Based on the comparison results (HSCS) of the built-in high-speed counter in the programmable controller, interrupt sub-program is executed to process counting results. When HSCS command output target is set for I010~I060, high-speed counter interrupt is applied. Corresponding sub-program must be properly programmed, and the corresponding interrupt symbol must be turned ON in order to proceed the responding interrupt.
- 3) Timer interrupt use: execute interrupt sub-program under each designated interrupt cycle time (1ms-99ms). Used in the loop interrupt processing control of the programmable controller's calculation cycle.

System provides 3 timer interrupts. Timer interrupt cycle can be programmed. Timer interruption uses the timer inside the system; therefore, it does not occupy T0~T255.

- 4) Using the special bit M8090~M8094 (corresponds to Y0~Y4 respectively) as ON, one-time user interrupt can be realized after pulse output completes; after the designated pulse number has been transmitted, user interrupt I502~I506 can be implemented immediately.
- 5) In order to support multiple free high-speed tasks when the high-speed counter is still in operation, high-speed counter multi-user interrupt has been realized (support up to 24 users, and all of them are expanded interrupt signals). It is defined using the configuration and comparison data chart.

Codes that correspond to "Interrupt enable" are as follows. Each code can be configured individually:

Interrupt enable/disable setting			
M8050	I00 port interrupt disabled	X input interrupt, total 12 interrupts. Each corresponds to the rising/falling edge interrupt respectively to X0~X5 ports. Port: 1= rising edge interrupt; 0= falling edge interrupt	Each code corresponds to one external interrupt control; When the M code is OFF, allow corresponding X to interrupt; When the M code is ON, disabled corresponding X to interrupt
M8051	I10 port interrupt disabled		
M8052	I20 port interrupt disabled		
M8053	I30 port interrupt disabled		
M8054	I40 port interrupt disabled		
M8055	I50 port interrupt disabled		
M8056	I600 port interrupt disabled	Timer interrupt 0	
M8057	I700 port interrupt disabled	Timer interrupt 1	
M8058	I800 port interrupt disabled	Timer interrupt 2	
M8059	Drive counter interrupt disabled	High-speed counter interrupt, total 6.	When it's ON, I010~I060 interrupts are disabled

Port Number	Special bit used	Corresponding user interruption
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

Code	Description
M8084	When ON, allow high-speed counter to perform multiple user interrupt
D8084	High-speed counter serial number 235~255
D8085	Number of corresponding user interrupt, maximum 24, I507~I530
D8086	Correspond to multiple comparative data serial number. Can only be D component at two-word bandwidth i.e. 200 is two word starting from D200

After the corresponding code for "interrupt enable" are ON, "overall interrupt disable" will need to be activated as well. EI command must be activated in order to use all interrupt functions. Overall interrupt disable DI command is used to prohibit correspondence to all interrupt. When the interruption permit setting code with input serial number is activated and the input signal satisfies interruption settings, corresponding sub-program will be executed.

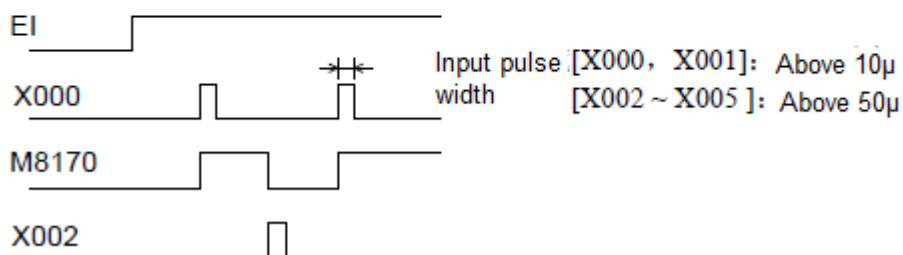
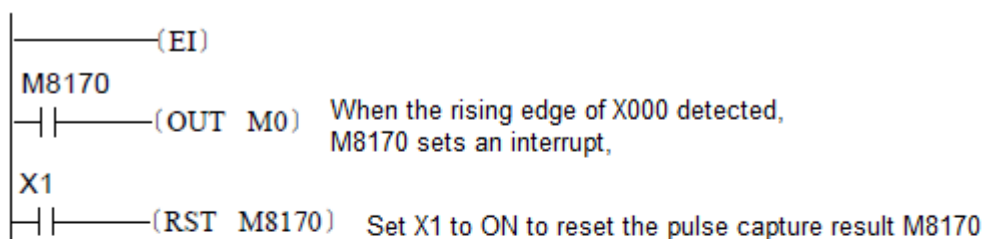
At the end of every interrupt sub-program, user must use IRET command to represent the completion of interrupt sub-program. After PLC executes the command, it will return to the point before the interrupt program

was initiated. (AutoShop interrupt program does not require IRET command)

The “pulse capture” function can be used when response to instantaneous pulse signal at X0~X5 ports is needed without special requirement of response time. PLC will store the rising edge signal of X0~X5 ports to M8170~M8175 which can be used by main program to judge and process and can be cleared manually after response.

Instruction for M8170~M8175:

After executing EI command, when input relay X000~X005 OFF to ON changes, auxiliary relay M8170~M8175 will reset to perform interruption process. Reset to preset components must be carried out by program to acquire pulse again. Pulse capturing action has no relation to individual M8050~M8055 auxiliary relays.



About the descriptions for sub-program and interrupts, please refer to 4.3.2.1 and appendix 5.8 for detailed instructions.

## 3.9 Constant K, H

H1U/H2U series programmable controller uses 7 types of data based on any specific purposes or applications. Their role and functions as follow:

Type	Application Notes in Programming
Decimal DEC	Timer and counter setting value (K constant) auxiliary relay (M), timer (T), counter (C), status S, etc. number (device number) Specify the value and command action in the operand of the application instruction (K constant)
Hexadecimal HEX	Same as decimal number, used to specify the operand and specified action in the application instruction (H constant)
Binary BIN	Use decimal numbers or hexadecimal numbers to specify the values of timers, counters, or data registers, but within the programmable controller, these numbers are all processed with binary numbers. Moreover, when monitoring on peripheral devices, these devices will automatically be converted to decimal numbers as shown in the figure (also can be switched to hexadecimal)
Octal OCT	The device numbers of input relays and output relays are assigned in octal numbers. Therefore, the carry of [0-7, 10-17...70-77, 100-107] can be carried out, in the octal number, there is no [8, 9]
BCD	BCD is a 4-bit binary method that represents the values of 0-9 in decimal numbers. It's easy to process each bit, so it can be used for BCD output type digital switch or seven-segment display control.
BIN float	The programmable controller has the function of high-precision floating-point operation, and internally uses binary (BIN) floating-point numbers for floating-point operations
Decimal floating point	The decimal floating-point value is only used for monitoring and is easy to read.

### Constant K

[K] is the symbol that expresses the 10-decimal integer. It is used to set the value of timer, counter, and value in the application instruction. In 16-bit commands, the value range for constant K is -32,768 ~ 32,767; in 32-bit commands, the value range for K constant is -2,47,483,648, 2,147,483,647.

### Constant H

[H] is the symbol that expresses the 16-decimal integer. It is used to set the values in the application instruction. Constant H's value range is 0000~FFFF; in 32-bit commands, constant K's value range is 0000,0000~FFFF, FFFF.

### 3.10 Table of elements

The power failure retention areas and none-power failure retention areas of H1U model cannot be changed while some areas of H2U can be changed.

H1U Model:

Input port X	X0~X377, maximum 256 points; XY sum up to 256 points		X0~X5: with interruption function;		Octal naming rules		
Output port Y	Y0~Y377, maximum 256 points; XY sum up to 256 points		Transistor type has 5 high-speed pulse output function, please refer to user manual for specific specifications		Octal naming rules		
Auxiliary relay M	[M0~M383] 384 points, General		[M384~M3071] 2688 points, for storage, power-off latched		M8000~M8511 512 points, Special, power-off latched		
Status S	[S0~S999] 1000 points, All for storage use, power-off latched						
Timer T	T0~T199 200 points, 100ms General		T200~T245 46 points, 10ms General		[T246~T249] 4 points, 1ms accumulative		[T250~T255] 6 points, 100ms accumulative
Counter C	16-bit up counter		32-bit up/down counter		High-speed counter		
	C0~C15 16 points General	[C16~C199] 168 points, power-off latched	C200~C219 20 points General, power-off latched	[C220~C234] 15 points power-off latched	C235~C245 Single- phase single counting	C246~C250 Single- phase dual counting	C251~C255 Two-phase counter
Data register D, V, Z	D0~D127 128 points, General	[D128~D7999] 7872 points, power-off latched	[D1000~D7999] Maximum 7000 points	[D8000~D8255] 256 points, Special	V7~V0, Z7~Z0 16 points, For address change		
Nested pointers	N0~N78 points, for main control	P0~P127 128 points, jump to subroutine; P128~P131, 4 points, for encrypt subroutine	I010~I060 total 6 points, counting interrupt pointer	I6**~I8** total 3 points, timer interrupt pointer	I00*~I50* 6-point input interrupt pointer		
Constant	K	16-bit, -32,768~32767		32-bit, -2,147,483,648~2,147,483,647			
	H	16-bit, 0~FFFFH		32-bit, 0~FFFFFFFFH			
	E (floating-point)	-		32-bit, 1175×10-41~3402×1035			

H2U Model:

Input port X	X0~X377, maximum 256 points; XY sum up to 256 points	X0~X5: with interrupt function; X0~X7: filtering time configurable;		Octal naming rules	
Output port Y	Y0~Y377, maximum 256 points; XY sum up to 256 points	Transistor possesses 5-way high-speed pulse output function. Details please refer to user manual			Octal naming rules
Auxiliary relay M	M0~M499 500 points, general (*1)	[M500~M1023] 524 points, storage (*2) relay	[M1024~M3071] 2048 points, storage (*3)	M8000~M8511 512 points, special	
Status S	S0~S499, total 500 points (*1) Initialization use S0~S9	[S500~S899] total 400 points, power-off latched (*2)			[S900~S999] total 100 points, alarm use (*2)
Timer T	[T0~T199] total 200 points, 100ms. Sub-routine: T192~T199	T200~T245 total 46 points, 10ms	[T246~T249], 4 points, 1ms accumulative (*3)	[T250~T255], 6 points, 100ms accumulative (*3)	
16-bit up counter C	[C0~C99] 100 points, general (*1)			[C100~C199] 100 points, storage (*2)	
32-bit counter C	32-bit up/down counter		32-bit high-speed counter		
	C200~ C219 20 points, general use (*1)	[C220~ C234] 15 points, storage (*2)	[C235~C245] Single-phase single counting (*2)	[C246~C250] single-phase dual counting (*2)	[C251~C255] Two-phase counting (*2)
Data register D, V, Z	D0~D199, 200 points, general (*1)	[D200~ D511] 312 points, storage (*2)	[D512~ D7999] Total 7488 points, storage (*3)	[D8000~D8255] total 256 points, special	V7~V0, Z7~Z0, 16 points, for indexing
Nested pointers	N0~N78 points, for main control	P0~P127 total 128 points, jump to subroutine; P128~P131, 4 points, for encryption subroutine	I00*~I50* total 6 points, input interrupt pointer	I6**~I8** total 3 points, timer interrupt pointer	I010~I060 total 6 points, counting interrupt pointer
Constant	K (Decimal)	16-bit, -32,768~32,767		32-bit, -2,147,483,648~2,147,483,647	
	H (Hexadecimal)	16-bit, 0~FFFFH		32-bit, 0~FFFFFFFFH	
	E (floating-point)			32-bit 1175×10-41~3402×1035	

The components in [ ] are the battery storage area

\*1: Non-battery storage area. According to the parameter setting, it can be changed to the battery storage area.

\*2: Battery storage area. According to the parameter setting, the non-battery storage area can be changed.

\*3: The battery is stored in a fixed area, and the area characteristics cannot be changed.

Note 1: H1U-XP (version 26128 and above) and H2U-XP (version 24137 and above) have CANlink3.0 function, because this function occupies D7000-D7999

D Component, D7000-D7999 is not allowed to be used by users after using CANlink3.0 function; D7000-D7999 users can use it normally when CANlink3.0 function.

Note 2: The system has 56 units of high-speed input and output resources, and the usage is as follows;

Item	Number of occupied units
High-speed output	7
Single phase single counter	6
Single phase dual counter	13
AB Phase counter	17
SPD counter	4
External interrupt	1
Pulse capture	1

For example, the system usage: 3 channels of pulse output, 2 channels of single-phase counting (the remaining 4 channels are pulse capture); the resources used are  $3*7+2*6+4*1=37$  units.





## Chapter 4 Instructions

### Contents

4.1 Basic instructions.....	- 63 -
4.2 STL/SFC instruction.....	- 70 -
4.2.1 STL Programming instruction .....	- 70 -
4.2.2 SFC Sequential Command Diagram Programming .....	- 72 -
4.3 Application instruction table.....	- 82 -
4.3.1 Program flow (00~09) .....	- 86 -
4.3.2 Transmission and comparison (10~19).....	- 97 -
4.3.3 Four logical arithmetic operations (20~29).....	- 107 -
4.3.4 Cyclic shift (30~39) .....	- 117 -
4.3.5 Data processing (40~49) .....	- 124 -
4.3.6 High-speed processing (50~59) .....	- 134 -
4.3.7 Convenient instructions (60~69).....	- 152 -
4.3.8 Peripheral I/O (70~79).....	- 167 -
4.3.9 Peripheral SER equipment (80~93) .....	- 187 -
4.3.10 Floating number (110~147) .....	- 212 -
4.3.11 Location control (155~159).....	- 237 -
4.3.12 Clock operation (160~169) .....	- 258 -



## Chapter 4 - Instructions

### 4.1 Basic instructions

Among the basic instructions, some instructions use the "function number" coding method. If the program is input by a handheld programmer, it can be input by using the corresponding command button in the keyboard or by using the function number.

Instruction	FUN NO.	Function	Operand Type	Instruction size
LD		Loading of NO contacts	S、X、Y、M、T、C	1
LDI		Loading of NC contacts	S、X、Y、M、T、C	1
LDP	90	Use of rising edge pulse	S、X、Y、M、T、C	1
LDF	91	Use of falling edge pulse	S、X、Y、M、T、C	1
AND		Series connection of NO contacts	S、X、Y、M、T、C	1
ANI		Series connection of NC contacts	S、X、Y、M、T、C	1
ANB		Series connection of circuit blocks	None	1
ANDP	92	Series connection for AND rising edge pulse detection		3
ANDF	93	Series connection for AND falling edge pulse (F) detection		3
OR		Parallel connection of NO contacts	S、X、Y、M、T、C	1
ORI		Parallel connection of NC contacts	S、X、Y、M、T、C	1
ORB		Parallel connection of circuit blocks	None	1
ORP	94	Parallel connection for OR rising edge pulse detection		3
ORF	05	Parallel connection for OR falling edge pulse detection		
OUT		Coil drive	S、Y、M	1
SET		SET action storage coil instruction	S、Y、M	1
RST		Contact or cache clearance	S、Y、M、T、C、D	3
PLS		Rising edge pulse detection coil instruction		
PLF		Falling edge pulse (F) detection coil instruction		
MC		Coil instruction for serial contacts used by the main control	N0~N7	3
MCR		Release instruction for serial contacts used by main control reset	N0~N7	3
MPS		Stack-based storage	None	1
MRD		Stack read (current pointer unchanged)	None	1
MPP		Stack read	None	1
NOP		No action	None	1
INV	98	Operation result inversion	None	1
END		End of all programs	None	1
P		Indicator	0~127	1
I		Indicator interruption insertion	I101/I201/301 etc.	1

## Basic instruction explanation

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

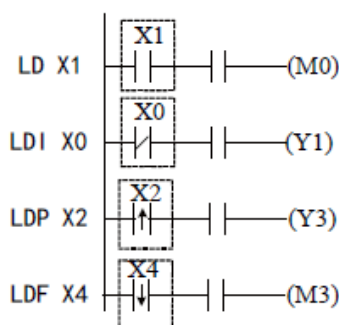
Instruction	Operand						
LD LDI LDP LDF	X0~X377	Y0~Y377	M0~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255
	✓	✓	✓	✓	✓	✓	

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

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



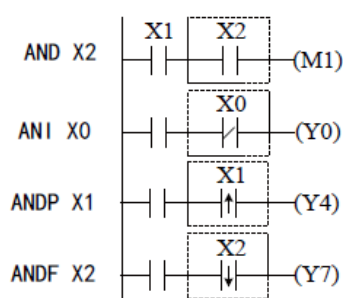
Instruction	Operand						
AND ANI ANDP ANDF	X0~X377	Y0~Y377	M0~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255
	✓	✓	✓	✓	✓	✓	

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 contact 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.



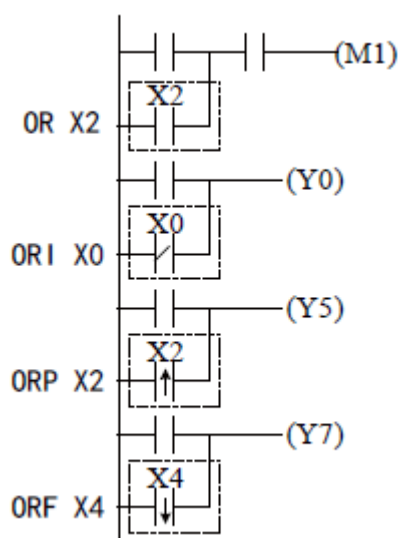
Instruction	Operand						
	X0~X377	Y0~Y377	M0~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255
OR ORI ORP ORF	✓	✓	✓	✓	✓	✓	

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 contact A and B for an OR operation.

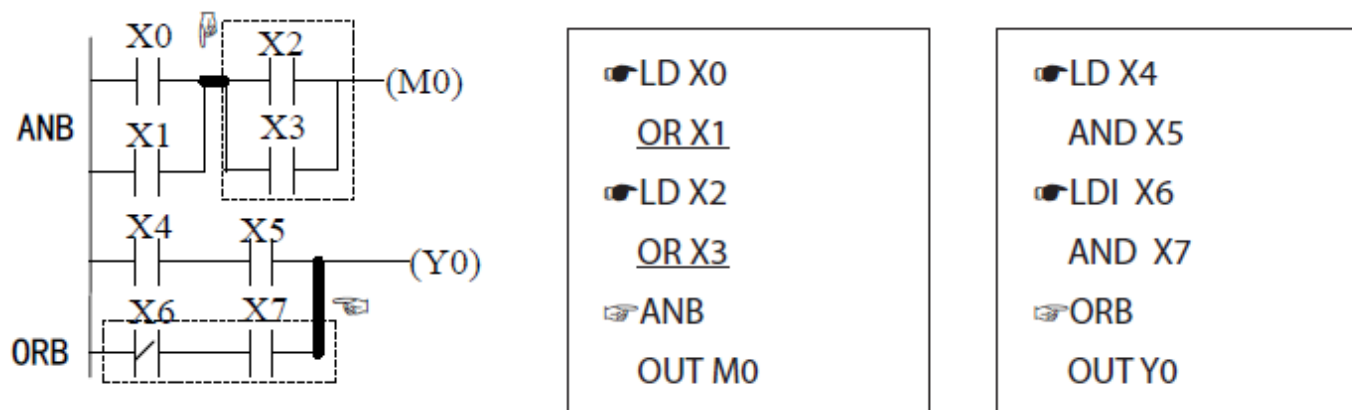
Use the ORP instruction to acquire the rising edge jump status of a contact for an OR operation.

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



Instruction	Operand
ANB	None
ORB	The object of 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" and "OR" operation on previously saved logic operation result and the content of the cumulative cache.



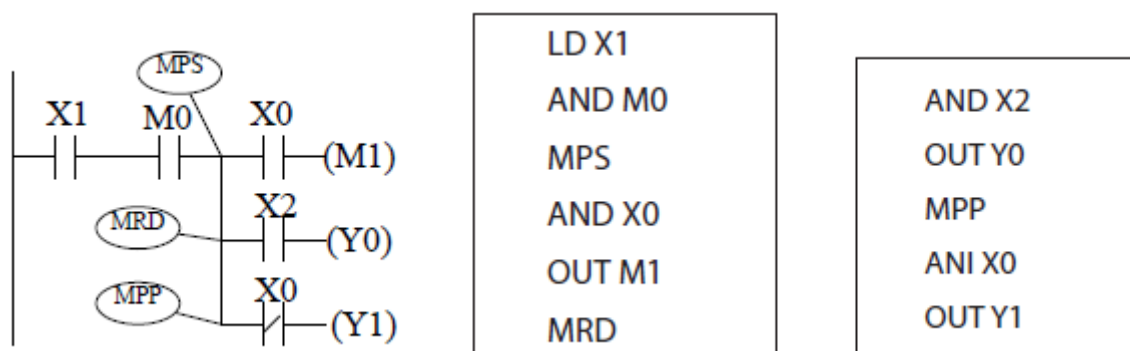
## MPS stack-based storage, MRD stack read, and MPP stack read

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.)

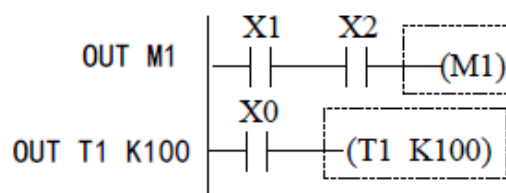
Use the MPP instruction to retrieve the previously saved logical operation result from the stack and store it in the cumulative cache. (The stack pointer is decremented by 1.)



## OUT, SET, RST, PLS, and PLF

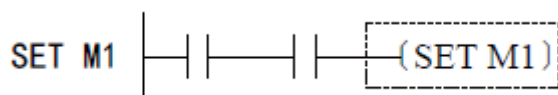
Instruction	Operand						
OUT	X0~X377	Y0~Y377	M0~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255
		✓	✓	✓	✓	✓	

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



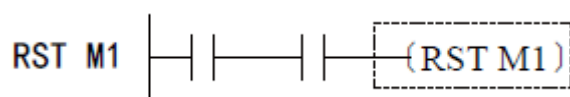
Instruction	Operand						
SET	X0~X377	Y0~Y377	M0~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255
		✓	✓	✓			

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~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255	V, Z
		✓	✓	✓	✓	✓	✓	✓

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.



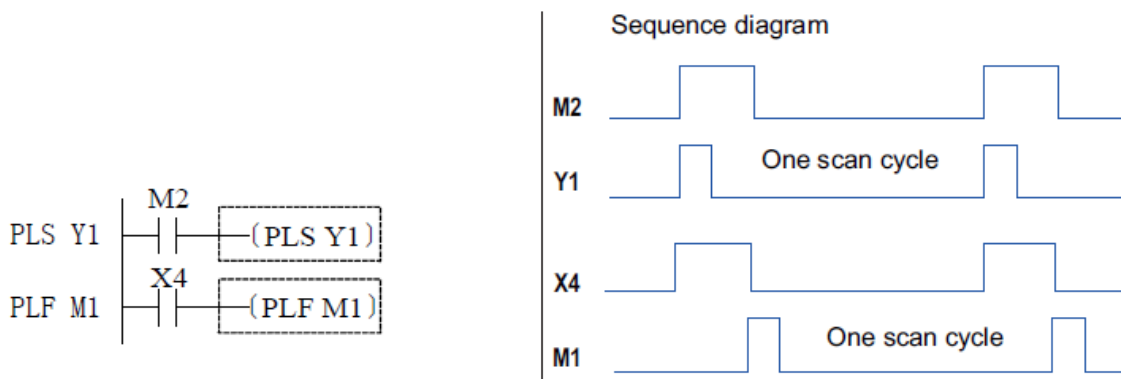
Instruction	Operand result
S, M, and Y	Coils and contacts are set to OFF.
T and C	The timer or counter in use is set to 0, and coils and contacts are set to OFF.
D, V, and Z	The values of the elements are cleared.

Instruction	Operand						
PLS PLF	X0~X377	Y0~Y377	M0~M3071 M8000~M8255	S0~S999	T0~T255	C0~C255	D0~D8255
		✓	✓				

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:



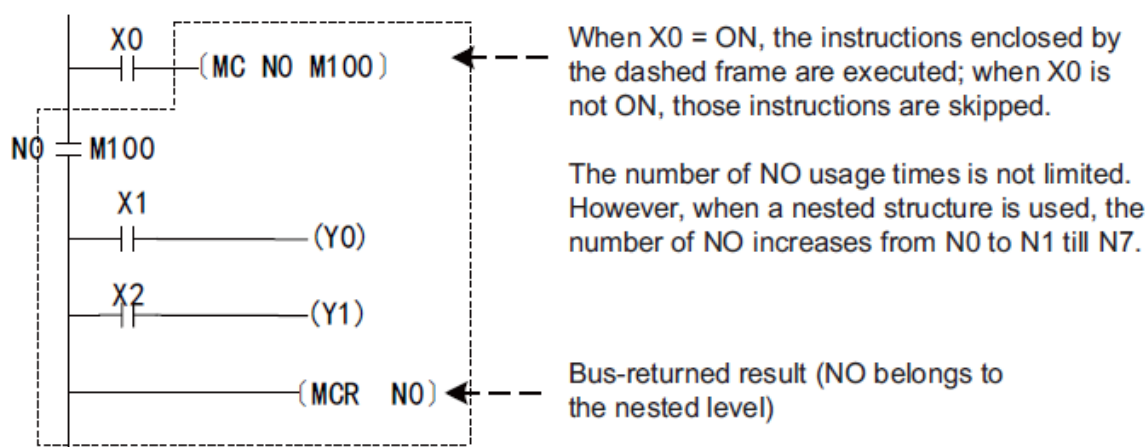
## Main Control Instructions

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.

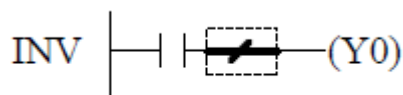
MCR is the main control ending instruction and located at the end of the main control program. This instruction must not be preceded by contact instructions. The MC and MCR instructions of the main control program support the nested program structure with a maximum of eight layers, which are numbered from N0 to N7 and used in sequence.



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.





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.

Instruction	Operand
FEND END	None

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 FEND 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 FEND 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.

Instruction	Operand
P	P0~P127 Used to mark the start of the jump address in the main program, where P63 is a dedicated address that points to END. Used to mark the starting address of the subroutine, each subroutine ends with SRET.
I	I00*~I50*, 6 points, input interrupt pointer; I6**~I8**, 3 points, timer interrupt pointer; I010~I060, 6 points, counting interrupt pointer

#### Pointer (P)

Pointer P is used for jump instruction CJ and subroutine call instruction CALL. It does not need to start with number 0, but the number cannot be used repeatedly, otherwise unexpected errors will occur. The timing of use is as follows:

- 1) Used in instruction CJ to indicate the destination address of the program execution jump and input the pointer P with the same number at the beginning of the target program.
- 2) Used in the instruction CALL to indicate the destination address of the subroutine and input the pointer P with the same number at the beginning of the subroutine.

## 4.2 STL/SFC instruction

### 4.2.1 STL Programming instruction

STL	Program return to secondary bus	S	1
RET	Program return to primary bus	None	1

#### 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.

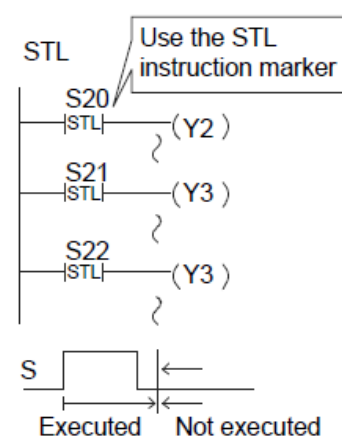
STL and ladder charts differ in programming.

- 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:**

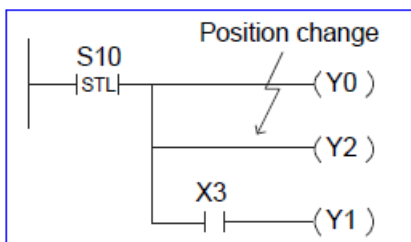
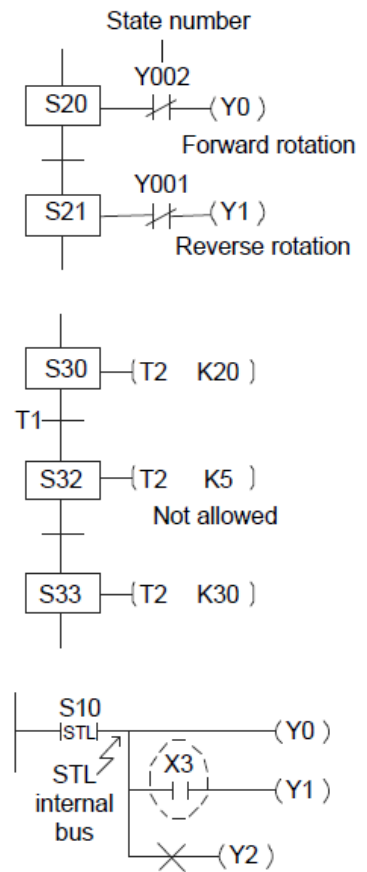
Two states are both connected for a period (one scan cycle) during state transition. To prevent simultaneous connection of a pair of outputs that cannot be connected at the same time, configure external interlock for the PLC and configure interlock for the corresponding program.

▪ **Repeated use of timers**

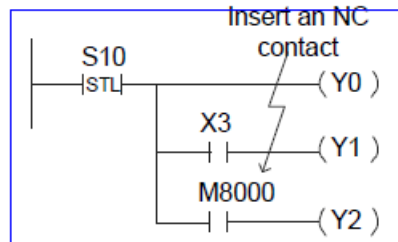
Like output coils, timer coils can be used to program the same element in different states, but programming in adjacent states is not allowed; otherwise, timer coils are not disconnected during process transfer and the current value cannot be reset.

▪ **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.

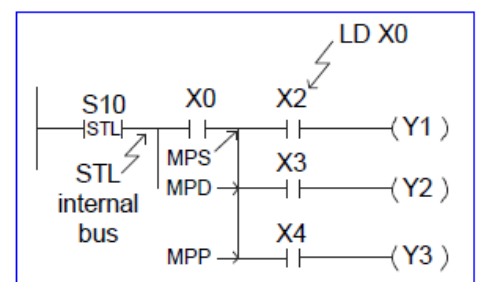


Or



▪ **Locations of MPS, MRD, and MPP for stack operation**

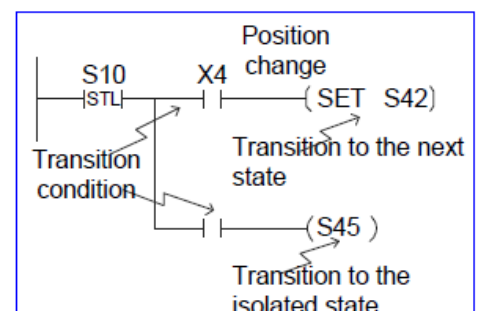
The MPS, MRD and MPP instructions cannot be used directly in the STL internal bus in intra-state mode. A program must be compiled after the LD or LDI instruction, as shown in the figure on the left.



▪ **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:

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

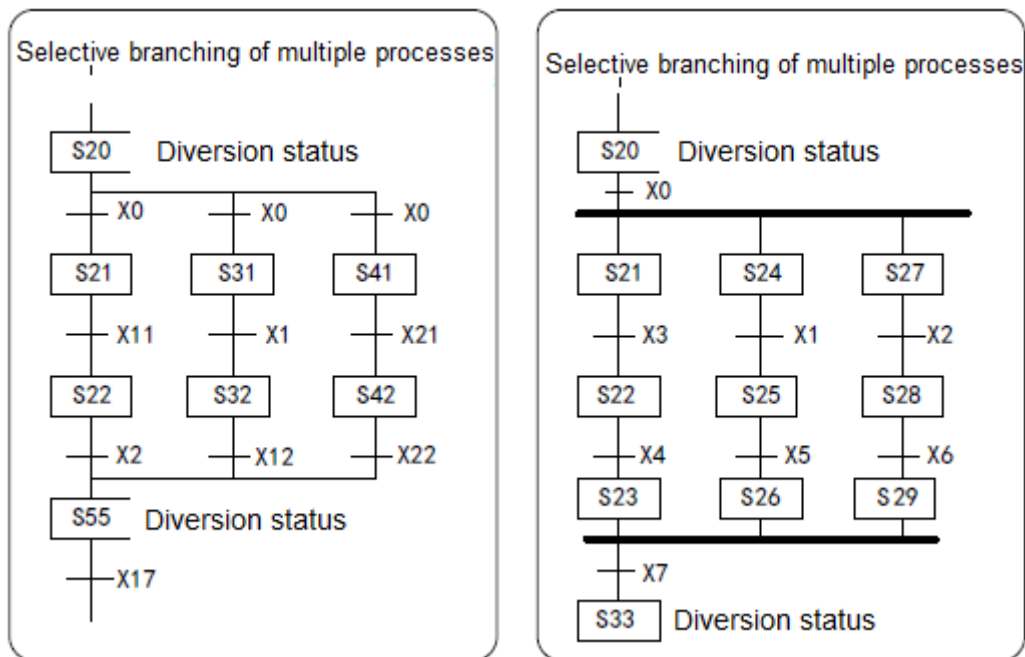
#### 4.2.2 SFC Sequential Command Diagram Programming

H1U/H2U Series Programmable Controller uses the sequential control function in the SFC diagram. SFC adopts the concept of a flow chart to demonstrate the control process so that the testing and maintenance of the programming becomes much simplified. Symbol definitions in SFC diagram design:

	Start-up stepper point graphics, which is used as the initial programming point for S0~S9 status. One starting symbol is used in one user program.
	Trapezoidal block graph used to illustrate that the internal programs are stepping ladder diagrams. Constant trapezoidal block graph serial number, such as LAD0, LAD1, etc.
	Normal stepping trapezoidal block graph may use S10~S889 status variables
	Status transition condition diagram, used to indicate conditions required between neighbouring statuses.
	Status separation diagram, used to indicate jumps between two non-neighbouring statuses.
	Upward status transition diagram, used to indicate upward transition status.
	Status reset diagram, reset program status to start status S0
	Select division diagram, under different condition, the transition happens from same stepping point to other corresponding stepping point.
	Select concurrence diagram, more than two stepping points are transferred to the same stepping point through corresponding transition conditions.
	Parallel division graph, one stepping point transfers the integrated body to more than two stepping points using one transition condition.
	Parallel division concurrence graph, when there are two different stepping points, under one transition condition they are transferred to the same stepping point.

#### 1) SFC's Programming Characteristic:

- In the ladder diagram block **LAD0** when the programmable controller changes from STOP→RUN, the auxiliary relay M8002, which acts instantaneously, sets the initial state S0 (ON);
- S0-S9 in the programmable controller are the initial state soft elements;
- S20-S889 and other states are assigned to each operation process. There is also a state for power failure retention, and its operating state can be saved even during a power failure. In addition, S10-S19 can be used for special purposes;
- The timers, counters and auxiliary relays in the programmable controller can be used at will;
- When there are multiple process selections or multiple processes that need to be performed at the same time, the following methods are used:



In the SFC diagram, the actions of the equipment in each process are clear and easy to understand, and its sequence control design is easy to facilitate debugging and maintenance.

- SFC diagram and step ladder diagram instructions are programmed according to certain rules, can be converted to each other, and their contents are the same. The familiar relay ladder diagram can also be used.

## 2) SFC programming method:

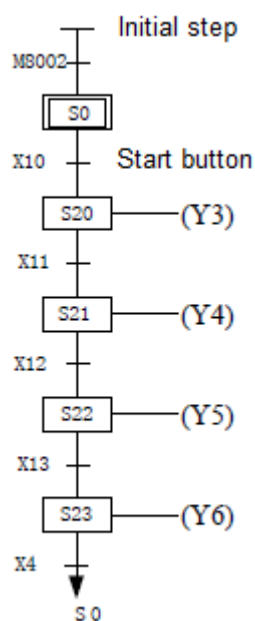
The following uses examples to illustrate the SFC programming method item by item.

### The role of the initial state

- The initial state is at the top of the SFC diagram, and state numbers S0-S9 can be used.
- When the initial state is also driven by other states (as shown in the example S23 in the above figure), it is necessary to use other methods to drive in advance at the start of operation.
- The example shown in the figure below is to use the special auxiliary relay M8002 that only acts instantaneously to drive when the programmable controller is switched from STOP→RUN.
- General states other than the initial state must be driven by STL instructions from other states, and

cannot be driven from states other than the initial state.

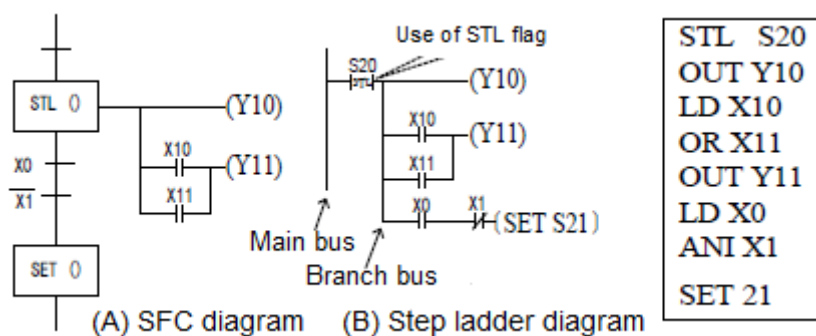
- The state driven by contacts other than the STL instruction is called the initial state. It must be stated at the beginning of the process. In addition, the STL instruction corresponding to the initial state,
- It must be programmed before a series of STL instructions after it.



### There is no general process of branching and confluence

Figure below is a typical SFC diagram. Each state has three functions: drive load, specify transfer destination, and specify transfer conditions. If using relay sequence control to show SFC diagram, the step ladder diagram in the figure (B) below.

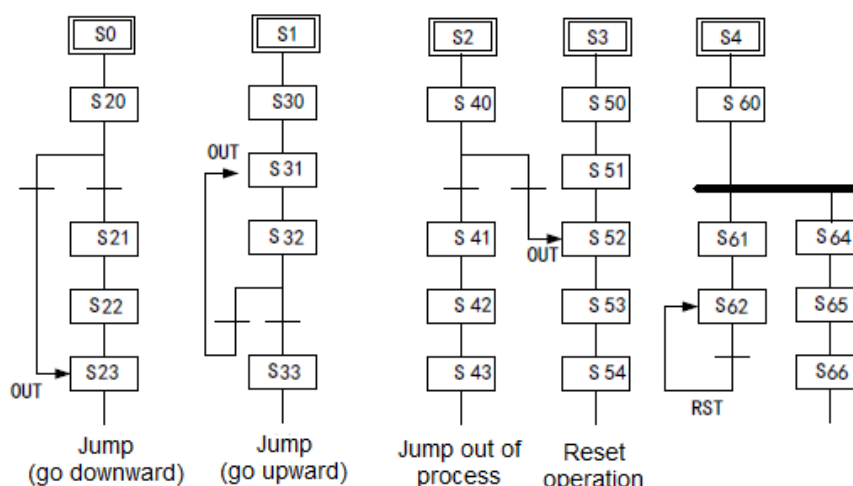
The program can be written with SFC diagram or ladder diagram. The programming sequence is to first carry out the drive processing of the load, and then carry out the transfer processing.



The above step ladder diagram also can be described using instruction list equivalently which is shown in right figure. The STL instruction is normally open contact instruction connected to main bus, and then coils can be connected to branch bus directly and drive coils through contacts.

Initial state must be placed before all STL instructions and RET instruction must be written at the end of program.

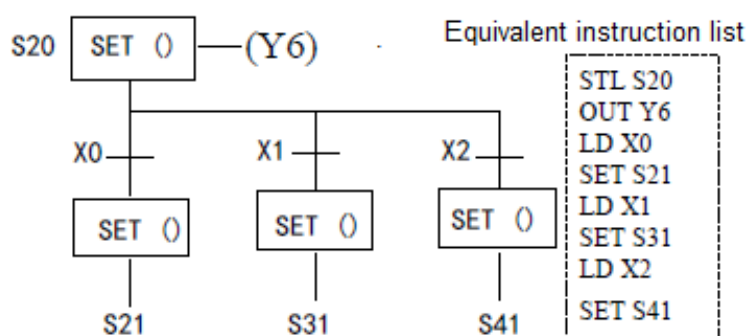
### General state with jump and loop



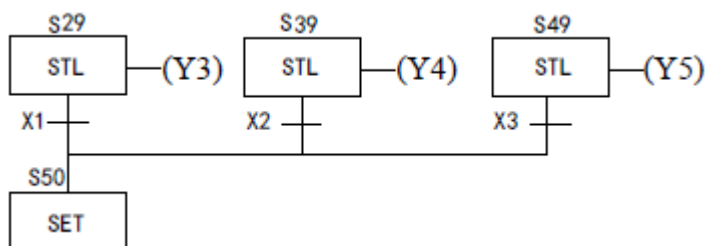
As shown in the figure above, the separate state transitions such as the transition to the lower state (jump), the transition to the upper state (repeat), and the transition out of the flow are programmed with the OUT instruction.

### Selective branch and converge status

Firstly, process of driving output and then process of state transferring which is the same as the process of general status. The equivalent instruction list is shown in right figure:



Equivalent instruction table:

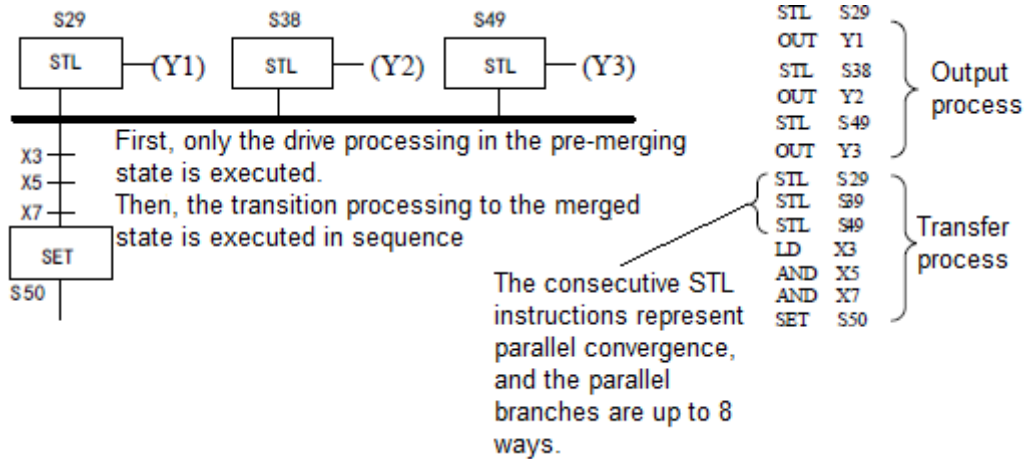
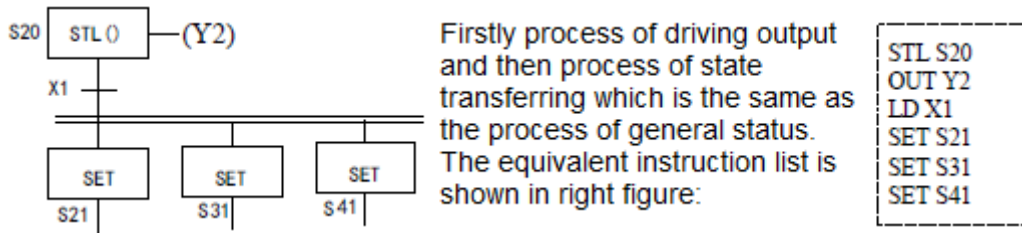


STL S29
OUT Y3
STL S39
OUT Y4
STL S49
OUT Y5
STL S29
LD X1
SET S50
STL S39
LD X2
SET S50
STL S49
LD X3
SET S50

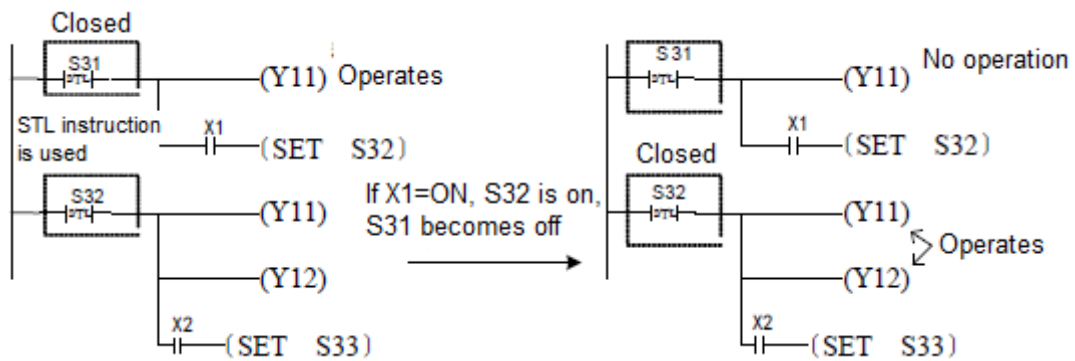
Above example is a typical example of branch converge, and the equivalent instruction table is shown in right side.

Following instructions can't be used in the transferring process program of branch and converge: MPS,MRD,MPP,ANB and ORB; On the other side, MPS instruction can't be used after STL instructions directly even in the load driving circuit.

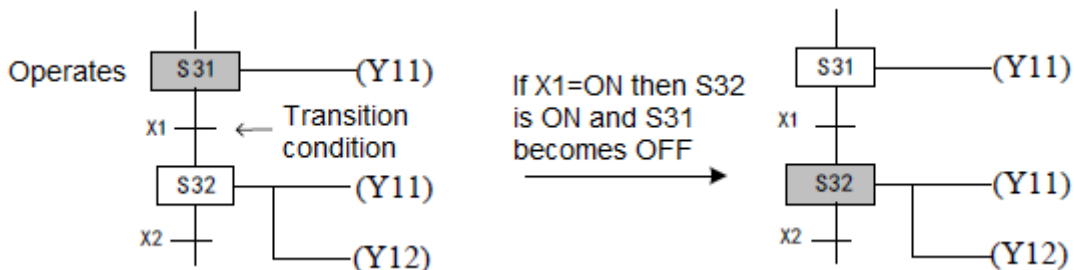
Status of parallel branch and converge



Ladder diagram and corresponding actions are shown in following figures:

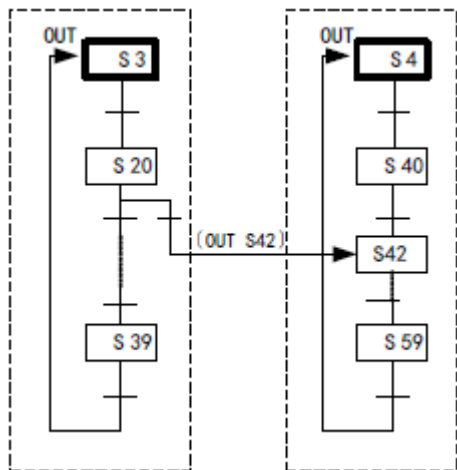


If the ladder diagram circuit is represented by SFC diagram then the representation is shown in the following figure:



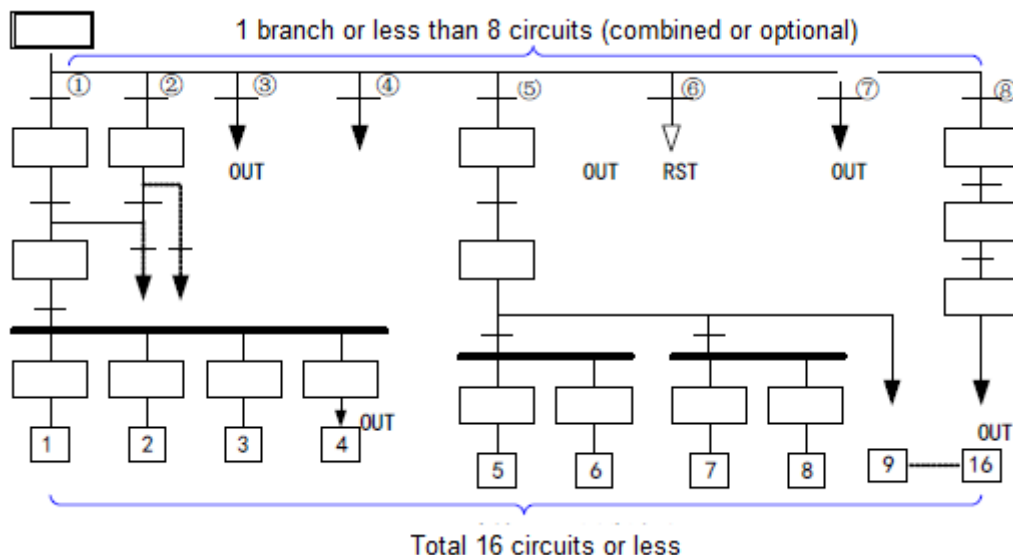


The SFC diagram with multiple initial states can be programmed individually for each state.



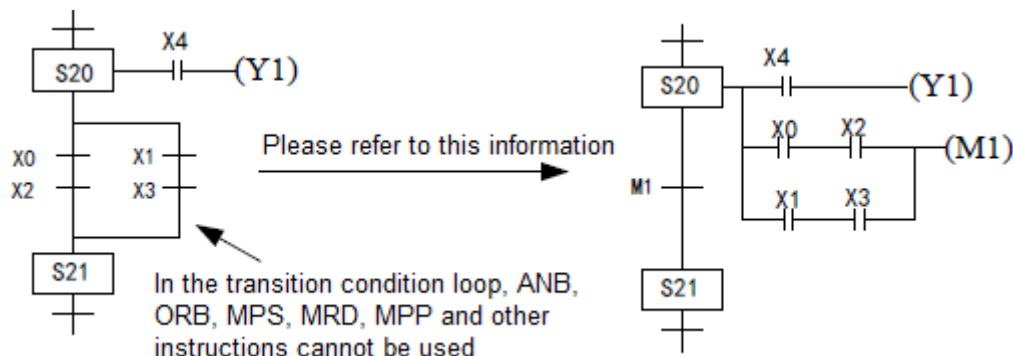
Just as shown in the left figure, initial state S3 corresponds to its program of STL instructions and initial state S4 corresponds to another program; Instructions other than STL can use the state number of the opposite side. OUT S42 instruction is included in the program of initial state S3 just as shown in the left figure. LD S39 instruction is included in the program of initial state S4. STL instructions can't be mixed together, which is very important.

Circuit number of one parallel branch or selective branch must be less than 8; But circuit number of each initial state must be less than 16 if multiple parallel branch or selective branch exist. As follow:



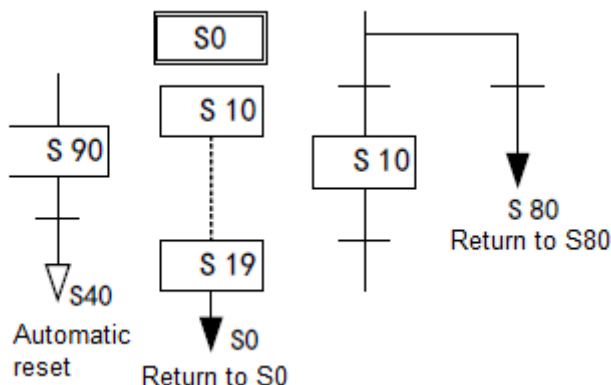
Transferring or reset from converge line or status before converge to separated status can't be executed. Void status should be set to execute transferring and reset process from the branch line.

Simplified process is recommended if the condition of status transferring is complicated. For example:



### Tranferring and reset of state:

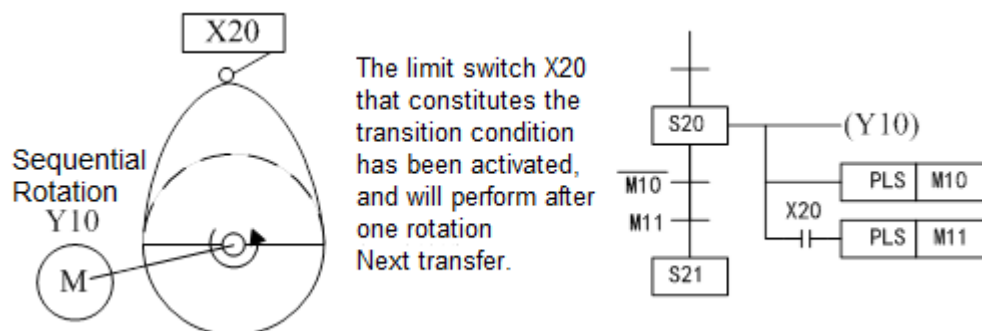
- In the process, the ↓ symbol indicates the repetition of the transition to the upper state or the transition (jump) to the lower state, or the state transition to another separate process.↴The symbol indicates the reset processing of the state.
- The status flag S can also be used to reset the flags in a section in batches using the ZRST instruction.
- When writing (SFC) diagram programs, you can use the following special auxiliary relays to improve programming efficiency, as shown in the following table



Soft element No.	Description	Function and purpose
M8000	RUN monitor	PLC needs relay which is always closed during operation. This component can be used as input condition of driven program or display of PLC operation status.
M8002	Initial pulse	Instantaneously (within 1 period) closed relay while PLC changing from STOP to RUN. This component can be used for initial setting or the set of initial state in program.
M8040	Transferring disabled	If this relay is driven then all the states are forbidden to be transferred. Since the program still operates even the state transferring is forbidden, the output coil will not be disconnected automatically.
M8046	STL operation	M8146 will be connected automatically if any state is ON. Used to avoid the same time start with other processes or be as the action flag for work process.
M8047	STL monitor enabled	If this relay is driven then the operating states can be read automatically and displayed by the programming function. Please refer to manuals of external devices for details.

- States which will be maintained during power failure use battery to maintain its operation status. These states can be used when powered again and continue to operate after power failure during mechanical operations.
- The RET instruction must be programmed at the end of STL instructions, or [program error] may occurs and the PLC will not run. The execution of this instruction will stop the step ladder diagram circuit. If interruption of a series of working procedure is needed in the main program then RET instruction is also needed, and multiple RET instruction can be programmed.

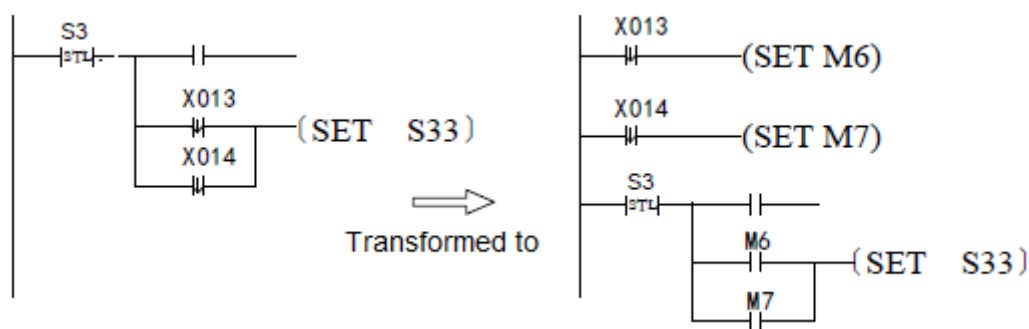
- State process when the transferring condition is met.



In the application shown in the figure above, the transition condition is pulsed, S20 acts for the first time, and the transition does not occur through M10.

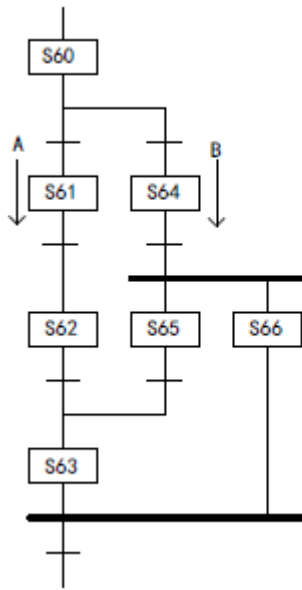
- Cautions when using rising/falling edge detection contacts:

When using the rising/falling edge detection side contact of LDP, LDF, ANDP, ANF, ORP, ORF in the state, the contact that changes when the disconnected state is detected after the state is re-connected. For the conditions that change when the state is disconnected, when the rising edge/falling edge must be detected, please modify the program as shown in the figure below:

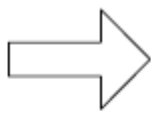
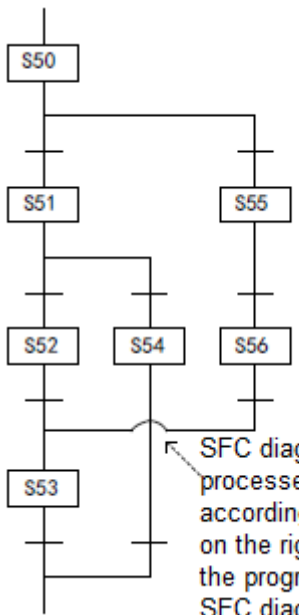
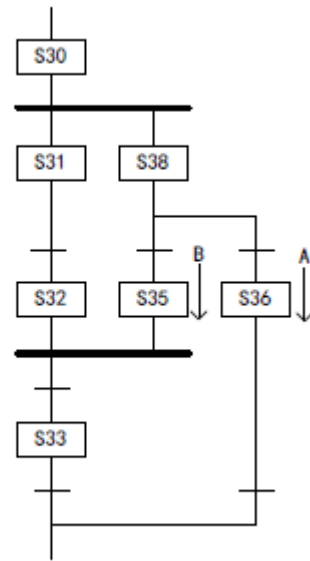


After transferring to S33 through the falling edge of X013, if X014 falls. At this time, because S3 is off, the falling edge of X014 cannot be detected. When S3 is on again, it is detected. Therefore, when S3 operates for the second time, immediately shift to S33.

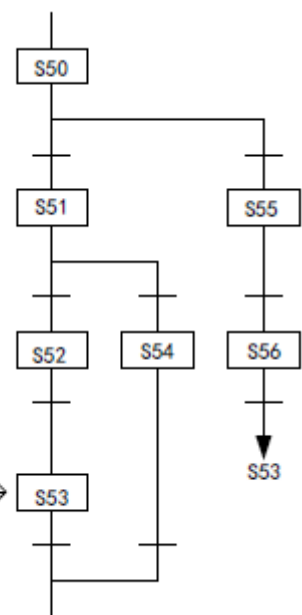
Composition process of branch and converge



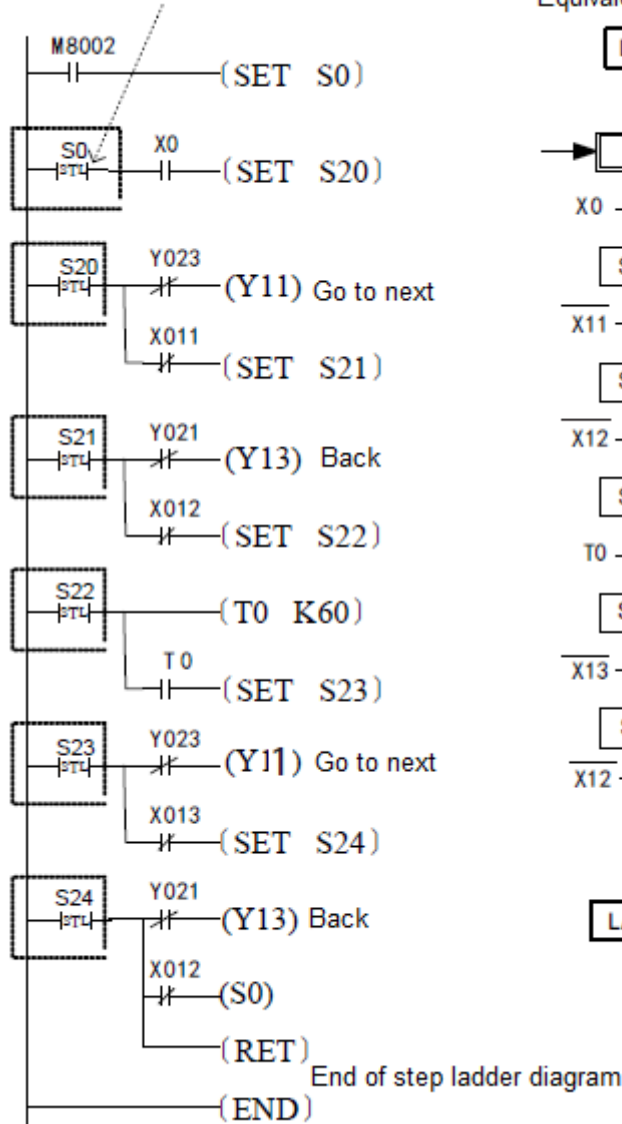
Notice:  
There is no problem with the process of B, but in some cases, the process of A has a state of waiting for action at the confluence.



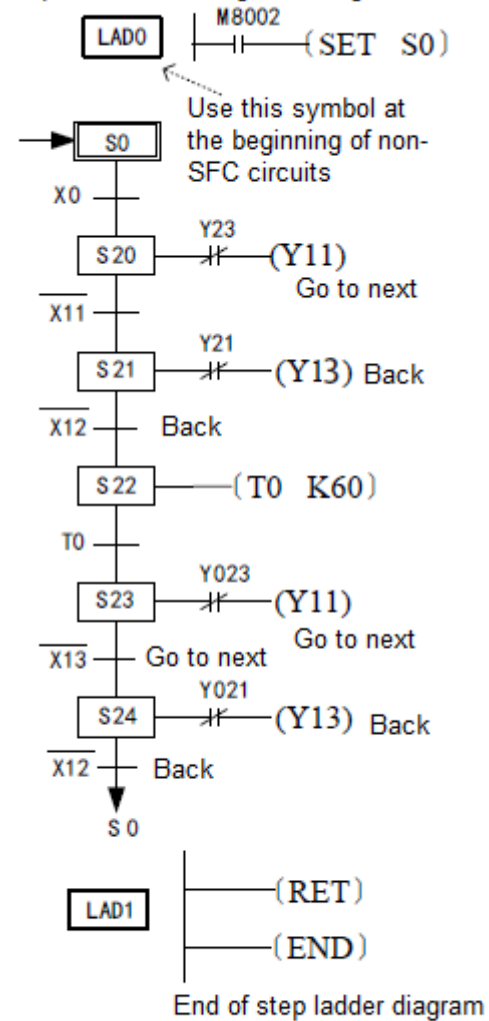
SFC diagrams cannot have processes that overlap. Program according to the procedure shown on the right. Only in this way can the program be reversed to the SFC diagram.



STL diagram design method



Equivalent SFC diagram design method



### 4.3 Application instruction table

Application instruction means the program instructions other than logical process instructions including program flow control, numerical calculation, high-speed signal processing, communication and extension and special control functions. These instructions not only have an instruction designation but also have a “function number” which is convenient for programming with handheld programming device. These instructions are list in the following table according to their function categories, and the function and usage of each instruction is described in detail in this chapter.

Class	Instruction symbol			Instruction function
	16-bit	32-bit	P instruction	
Program flow control	CJ	–	✓	Conditional jump
	CALL	–	✓	Subroutine call
	SRET	–	–	Sub-program Reset
	IRET	–	–	Interrupt Reset
	EI	–	–	Enable interrupt
	DI	–	–	Disable interrupt
	FEND	–	–	Master Program Termination
	WDT	–	✓	Monitor Timer
	FOR	–	–	Start of loop range
	NEXT	–	–	End of loop range
Transmit and Compare	CMP	✓	✓	Compare
	ZCP	✓	✓	Zone comparison
	MOV	✓	✓	Transmit
	SMOV	–	✓	Shift Transmit
	CML	✓	✓	Reverse Transmit
	BMOV	–	✓	Both Transmit
	FMOV	✓	✓	Multiple Point Transmission
	XCH	✓	✓	Exchange
	BCD	✓	✓	BCD conversion
	BIN	✓	✓	BIN conversion
Four arithmetic operations	ADD	✓	✓	BIN addition
	SUB	✓	✓	BIN subtraction
	MUL	✓	✓	BIN multiplication
	DIV	✓	✓	BIN division
	INC	✓	✓	Increase by 1 (BIN)
	DEC	✓	✓	Decrease by 1 (BIN)
	WAND	✓	✓	Logical Character AND
	WOR	✓	✓	Logical Character OR
	WXOR	✓	✓	Logical Character XOR
	NEG	✓	✓	Complemental Code
	ROR	✓	✓	Cycle Right Shift
	ROL	✓	✓	Cycle Left Shift

Class	Instruction symbol			Instruction function
	16-bit	32-bit	P instruction	
Data shift	RCR	✓	✓	Carry Cycle Right Shift
	RCL	✓	✓	Carry Cycle Left Shift
Data shift	SFTR	✓	✓	Right shift
	SFTL	-	✓	Left shift
	WSFR	-	✓	Shift right by word
	WSFL	-	✓	Character Left Shift
	SFWR	-	✓	“FIFO” write
	SFRD	-	✓	“FIFO” read
Data processing	ZRST	-	✓	Interval reset
	DECO	-	✓	Encode
	ENCO	-	✓	Decode
	SUM	✓	✓	Count of ON
	BON	✓	✓	Determination of ON count
	MEAN	✓	✓	Average
	ANS	-	-	Signal Alarm Set
	ANR	-	✓	Signal alarm reset
	SOR	✓	✓	BIN square root
	FLT	✓	✓	floating-point to decimal conversion
High speed processing	REF	-	✓	Input/output refresh Input/output refresh
	REFE	-	✓	Filter Adjust
	MTR	-	-	Matrix input
	HSCS	✓	-	Comparative Set (High-speed counter)
	HSCR	✓	-	Comparative Reset (High-speed counter)
	HSZ	✓	-	Comparative Interval (High-speed counter)
	SPD	-	-	Pulse density
	PLSY	✓	-	Pulse output
	PWM	-	-	Pulse width modulation
	PLSR	✓	-	Pulse output with Acceleration and Deceleration
Other peripheral instructions	IST	-	-	State initialization
	SER	✓	✓	Data search
	ABSD	✓	-	Absolute cam control
	INCD	-	-	Incremental cam control
	TIMR	-	-	Teach timer
	STMR	-	-	Special timer
	ALT	-	-	Alternate state
	RAMP	-	-	Ramp variable value
	ROTC	-	-	Rotary table control

Class	Instruction symbol			Instruction function
	16-bit	32-bit	P instruction	
	SORT	–	–	Sort of list data
External device I/O	TKY	✓	–	Decimal number input 0-9
	HKY	✓	–	Hexadecimal number input
	DSW	–	–	Digital Switch
	SEGD	–	✓	7 segments coding
	SEGL	–	–	7 segments display with latch
External device I/O	ARWS	–	–	Vector switch
	ASC	–	–	ASCII code conversion
	PR	–	–	ASCII code print
	FROM	✓	✓	Special function block reading
	TO	✓	✓	Special function block writing
External device SER	RS	-	-	Serial communication instruction
	PRUN	✓	✓	Parallel operation
	ASCI	-	✓	HEX→ASCII conversion
	HEX	-	✓	ASCII→HEX conversion
	CCD	-	✓	Check sum code
	PID	-	-	PID calculation
	MODBUS	-	-	Serial communication instruction
	CANTX	-	-	CAN transmission
	CANRX	-	-	CAN receive
	RS2	-	-	Serial communication 2
	MODBUS2	-	-	Serial communication 2
Floating points	ECMP	✓	✓	Binary floating-point number comparison
	EZCP	✓	✓	Binary floating-point zone comparison
	EBCD	✓	✓	Binary-decimal floating-point number conversion
	EBIN	✓	✓	Decimal-binary floating-point number conversion
	EADD	✓	✓	Binary floating-point addition
	ESUB	✓	✓	Binary floating-point subtraction
	EMUL	✓	✓	Binary floating-point multiplication
	EDIV	✓	✓	Binary floating-point division
	ESOR	✓	✓	Binary floating-point square root
	INT	✓	✓	Binary floating-point – BIN integer conversion
	SIN	✓	✓	Floating-point SIN operation
	COS	✓	✓	Floating-point COS operation
	TAN	✓	✓	Floating-point TAN operation
	SWAP	✓	✓	Upper and lower byte conversion



Class	Instruction symbol			Instruction function
	16-bit	32-bit	P instruction	
Locate Instruction	ABS	✓	-	ABS position readout
	ZRN	✓	-	Regression through the Origin
	PLSV	✓	-	Variable pulse output
	DRVI	✓	-	Relative position control
	DRVA	✓	-	Absolute position control
Clock operation	TCMP	-	✓	Clock data comparison
	TZCP	-	✓	Clock data region comparison
	TADD	-	✓	Clock data addition
	TSUB	-	✓	Clock data subtraction
	TRD	-	✓	Clock data readout
	TWR	-	✓	Clock data input
	HOUR	✓	-	Timer
	GRY	✓	✓	Grey code conversion
	GBIN	✓	✓	Grey code inverted conversion
Contacts comparison	LD=	✓	-	(S1) = (S2)
	LD>	✓	-	(S1) > (S2)
	LD<	✓	-	(S1) < (S2)
	LD<>	✓	-	(S1) <> (S2)
	LD<=	✓	-	(S1) <= (S2)
	LD>=	✓	-	(S1) >= (S2)
	AND=	✓	-	(S1) = (S2)
	AND>	✓	-	(S1) > (S2)
	AND<	✓	-	(S1) < (S2)
	AND<>	✓	-	(S1) <> (S2)
	AND<=	✓	-	(S1) <= (S2)
	AND>=	✓	-	(S1) >= (S2)
	OR=	✓	-	(S1) = (S2)
	OR>	✓	-	(S1) > (S2)
	OR<	✓	-	(S1) < (S2)
	OR<>	✓	-	(S1) <> (S2)
	OR<=	✓	-	(S1) <= (S2)
OR>=	✓	-	(S1) >= (S2)	

### 4.3.1 Program flow (00~09)

#### CJ: Conditional jump

16-bit	32-bit	P	CJ P*** CJP P***	Conditional jump
✓		✓		
3 steps		3 steps	P000~P127	

- 1) If the power flow is valid, then program jumps from the address of CJ (or CJP) to the address specified by P\*\*\* and continue execution, and the instruction with intermediate address will be jumped over and not be executed.
- 2) If the power flow is invalid, then program continue execution and CJ (or CJP) instruction will not be executed.

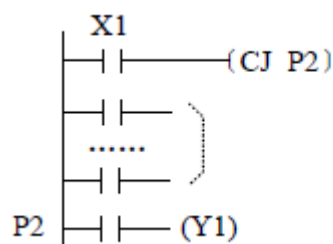
If TMR timer or counter exist in the program which is jumped over and the timer or counter have been driven, then the corresponding operation will be as follows:

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	

Requirements of P\*\*\* address pointer:

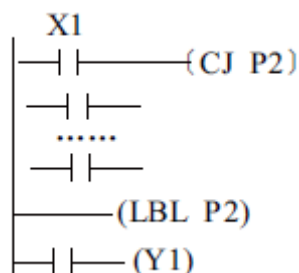
- The address pointer used by CJ (or CJP) must be before the termination of main program (FEND instruction);
- The address of END which is pointed by P63 can't be defined to other program steps;
- The program starting from P\*\*\* doesn't need SRET instruction for termination, which is different from subroutines;
- The definition address of P\*\*\* should not be duplicated;
- This instruction can be used when user wants part of the program not to be executed to shorten the scan period or wants to avoid dual-coils when output through two coils;
- CJ instruction can specify the same pointer P repeatedly, but CJ can't specify the same pointer P as the one of CALL instruction, or errors will occur.

Example:



If X1 is ON, the program in the middle will be skipped and will not be executed, and jump to P2 to continue execution.  
If X1 is OFF, the program is executed normally as a normal program, and the CJ instruction is not executed

The CJ instruction is used as follows in AutoShop:



And the subroutine and interrupt program are written in separate windows, so there is no need to pay attention to FEND and other matters. CJP63 is CJEND in the AutoShop programming environment.

16-bit	32-bit	P	CALL P*** CALLP P***	Conditional jump
✓		✓		
3 steps			P000~P127	

If the power flow is effective then program will call the subroutine specified by P\*\*\*. The program will return to the next statement of this CALL (or CALLP) instruction and continue for running.

Requirements of address pointer P\*\*\*:

- The subroutine starting from P\*\*\* must be before the termination of main program (FEND instruction);
- The subroutine must be terminated by SRET instruction;
- The subroutine starting from P\*\*\* can be called for several times and called by other subroutines, but the nest level should not exceed 5 levels;
- The subroutine can't call itself within itself, which can avoid endless loop and operation overtime of program.
- T192~T199 or T246~T249 can be used as timer in subroutines.

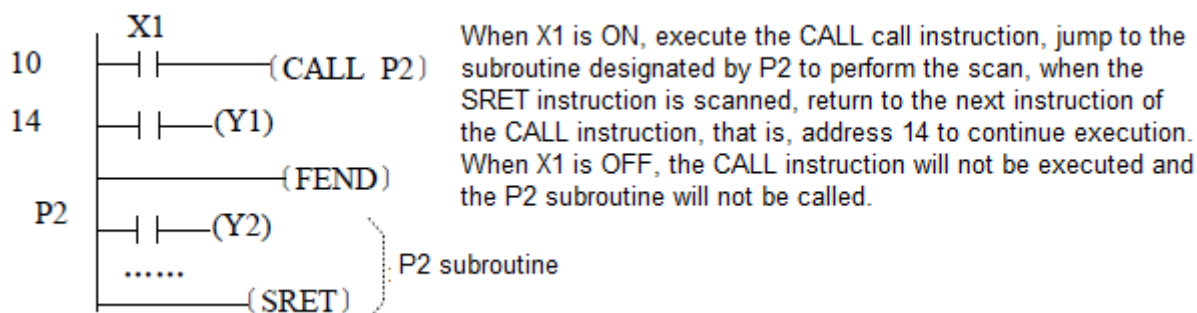
In the AutoShop programming environment, subprograms are written in a separate window, so there is no problem with FEND, SRET and other instructions, and the names of subprograms can be modified.

16-bit	32-bit	P	SRET	Conditional jump
✓				
1 step			No need for contact, no separate instructions for operands	

SRET instruction locates at the end of the subroutine. The program will return to the statement which calls this subroutine and continue running after the execution of this instruction.

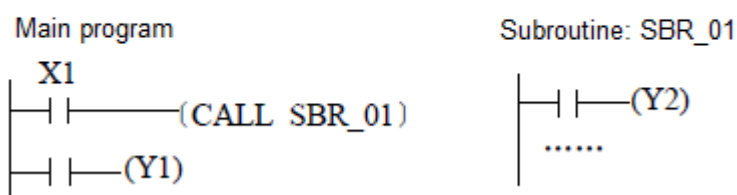
It's not necessary to write SRET at the end of subroutine in AutoShop programming environment.

Example 1:



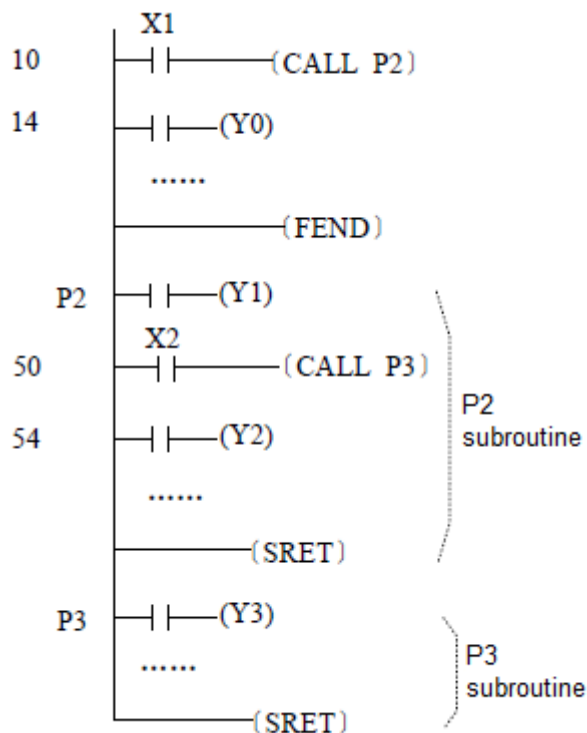
Instruction format of “instructin example 1” in AutoShop programming environment is shown below:

Right-click the subroutine SBR\_01 of the program block and select properties. You can modify 01 to the name you want.



Example2:

When X1 is ON, execute the CALL instruction and jump to the subroutine designated by P2 to perform the scan. There are two cases: if X2 is OFF when the scan reaches address 50, continue scanning until the SRET instruction, and return to the main program. Address 14 continues to execute down; if X2 is ON when address 50 is scanned, CALL P3 is executed, and the subroutine specified by P3 is executed for scanning. When SRET instruction is executed, it returns to P2 subroutine address 54 and continues down carried out

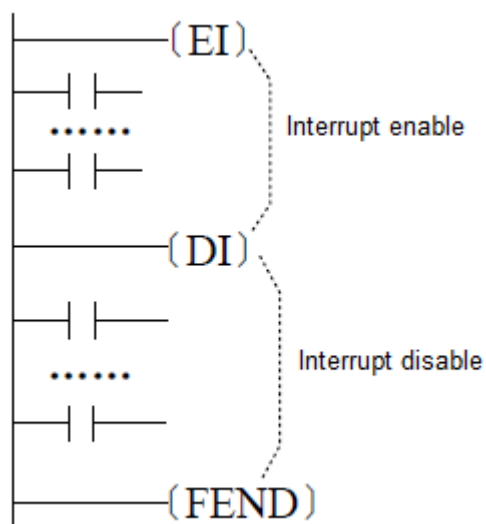


16-bit	32-bit	P	IRET	Interrupt program completed	No need for contact, no separate instructions for operands
✓			EI	Interrupt enable	
1 step			DI	Interrupt disable	

IRET instruction is placed at the end of the interrupt routine. The program will return to the statement before calling this interrupt routine and continue to operate after the execution of this instruction. Interrupt routine is programmed in a separate window in the AutoShop programming environment, so IRET instruction is not necessary to be programmed in the interrupt routine.

The interrupt is disabled when the PLC starts to run; The interrupt will be enabled after the execution of EI instruction; If the interrupt has been enabled and DI instruction is executed, then interrupt will be disabled. If there is no space for inserting interrupt disable instruction in the program, then DI instruction could not be used.

Types and settings of interrupts:



- 1) External signal input interrupt: The rising or falling edge of X0~X5 input signal can be defined to interrupt. For X signals that do not require immediate response, the pulse capture function can also be used;
- 2) High-speed counting interrupt: Comparing with FNC53 (DHSCS) and using the set instruction, when the current value of high-speed counting reaches the set value, an interrupt is generated;
- 3) Timer interrupt: an interrupt that occurs at a fixed cycle of 1ms~99ms;
- 4) Pulse completion interrupt: After the specified number of pulses are sent, the interrupt will be executed immediately;
- 5) Multi-user interrupt (XP does not have this function): A high-speed counter can be arbitrarily taken to perform up to 24 interrupts.

External signal input interrupt pointer and settings (H1U-XP, H2U-XP):

Input number	Pointer number		Interrupt disable register
	Rising edge interrupt	Falling edge 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 pointer and settings (H1U-XP, H2U-XP):

Input number	Interrupt period MS	Interrupt disable register
I6xx	Enter a number from 1 to 99 in the "XX" command port, such as I605, execute a timer interrupt every 5ms	M8056
I7xx		M8057
I8xx		M8058

High-speed counting interrupt pointer and settings (H1U-XP, H2U-XP):

Input number	Interrupt disable register
I010	M8059
I020	
I030	
I040	
I050	
I060	

Pulse output completion interrupt pointer and setting: (This function needs to start M8090~M8094 to generate an interrupt after the pulse output is completed) (H1U-XP, H2U-XP).

Port No.	Using special bit	Corresponding user interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

The interrupt subroutine uses different numbers, that is, different ports and interrupt trigger edges are selected;

For the same X input in the external input interrupt, the rising interrupt and the falling interrupt cannot be numbered at the same time. For an X input port, only one trigger edge can be used, and the trigger edge is set by the pointer number;

External input interrupt: If M8050-M8055 is "ON" during program execution, the interrupt function of the corresponding X port is prohibited.

Timing interrupt: If M8056-M8058 is "ON" during program execution, the corresponding timing interrupt function is prohibited.

High-speed counting interrupt: If M8059 is "ON" during program execution, all high-speed counting interrupt functions are prohibited.

Interrupt programming regulations and execution characteristics:

- When an interrupt occurs between DI-EI instructions (interrupt prohibition interval), it can also be memorized and executed after the EI instruction.
- The interrupt subroutine must be written after the FEND instruction, and the end of the subroutine must end with IRET. In the AutoShop environment, do not write in the main program. The end of the subroutine can omit IRET
- The pointer number cannot be reused
- When multiple interrupts occur in sequence, the one that occurs first has priority. When it happens at the same time, the one with the highest priority is given priority. The priority levels from high to low are high-speed counter interrupt, external interrupt, time interrupt, and pulse output completion interrupt.
- During the execution of the interrupt routine, other interrupts are prohibited.

- When controlling input relays and output relays during interrupt processing, use the input and output refresh command (REFF) to achieve high-speed control by reading the latest input state or outputting the calculation result immediately;
- The number of the input relay used as the interrupt pointer should not overlap with the number of application commands such as [High Speed Counter] and [Pulse Density (FNC56)] that use the same input range.
- For timers in subroutines and interrupt routines, please use routine timers T192-T199; if you use general timers, in addition to not being able to keep time, you also need to pay attention when using the 1ms cumulative timer
- If input interrupt pointer I port and O port are designated, the input filter feature of the input relay is automatically closed. Therefore, there is no need to use REFE (FNC51) instructions and special data register D8020 (input filter adjustment). In addition, the input filter of the input relay that is not used as an input interrupt pointer can maintain 10ms (initial value).

In order to meet the requirements of high-speed counter operation, we have added 24 new high-speed counting interrupts in the H2U model. Any high-speed counter can be designated to generate 24 interrupt responses. This function is named High-speed counter multi-user interrupt (XP does not with this feature). The user setting has the following rules;

Flag bit	Description
M8084	Multiple interrupt of high speed counter when this flag is ON
D8084	Sequence number of high speed counter C235~C255
D8085	Number of corresponding user interrupts from I507~I530, maximum 24
D8086	Sequence number corresponding to multiple comparison data, which only can be D component. For example 200 means double words starting from D200.

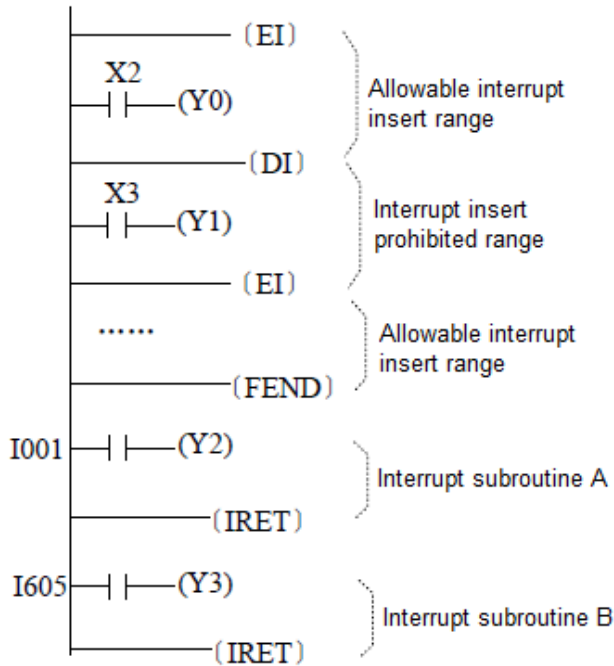
Example of the Comparison Point Data Storage:

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

Data of C235	Recorded cell	Store unit value	Corresponding user interruption	Value of D8131
100	D200, D201	=100	I507	0
200	D202, D203	=200	I508	1
300	D204, D205	=300	I509	2
400	D206, D207	=400	I510	3
500	D208, D209	=500	I511	4 → 0 (M8133=ON)

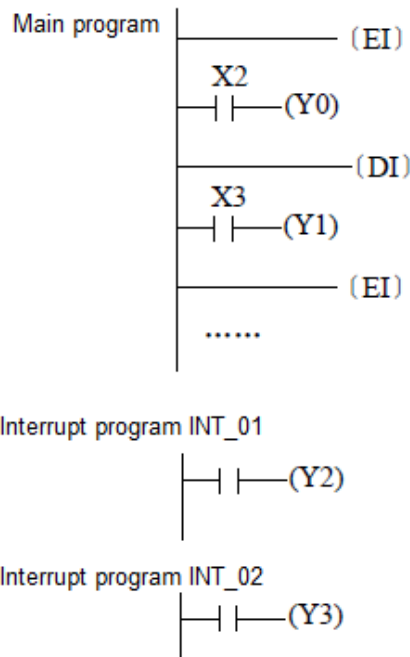
Each interrupt can be generated by the value of the high-speed counter and the value of the recording unit.

Example:



When the PLC executes, when the program scans between the EI instruction and the DI instruction, X0=On or when the timing time 5ms expires, the interrupt insertion subroutine A or B will be executed, and when the subroutine is executed to IRET, it will return to the main program and continue execute down.

The program on the left in AutoShop is as follows: (right-click the interrupt program INT\_01 or INT\_02 of the program block, and select properties. You can modify 01 or 02 to the any name you want)



Right-click the interrupt programs INT\_01 and INT\_02 of the program block respectively, Select attribute-interrupt items: set to I001 and I605 respectively

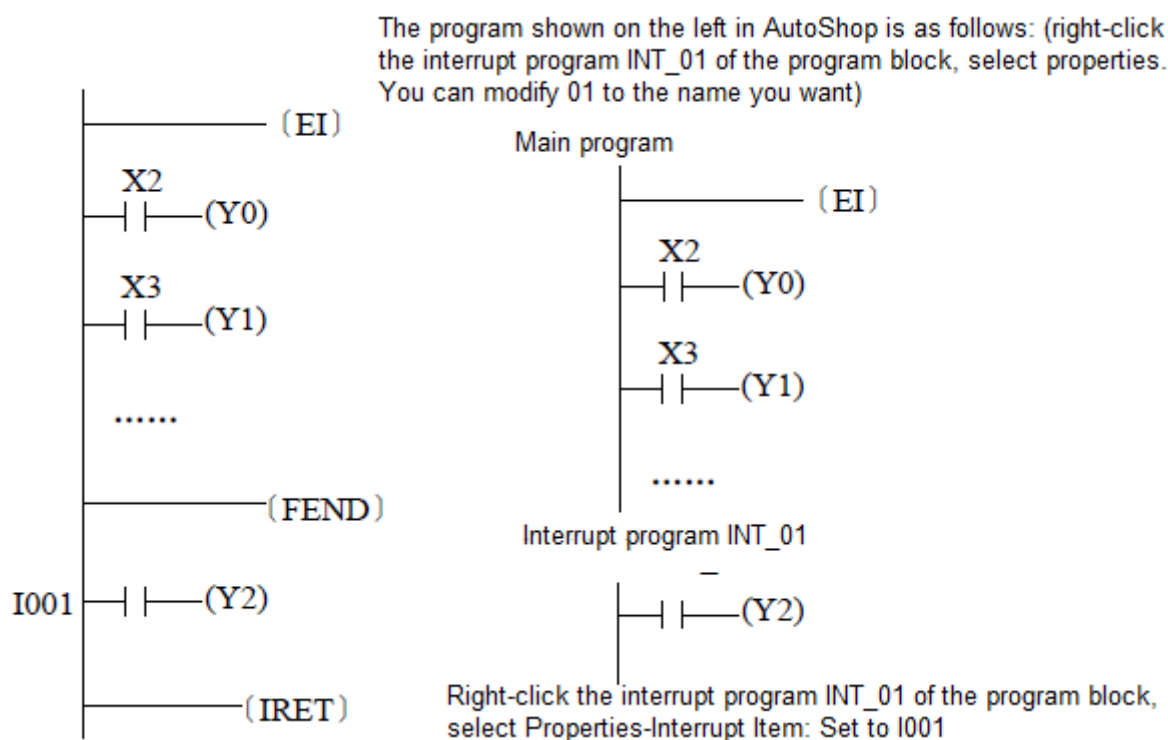


16-bit	32-bit	P	FEND	The main program is completed.
✓				
1 step			No need for contact, no separate instructions for operands	

The FEND statement is located at the end of the main program. After the instruction is executed, the scan of the user program is ended, the program returns to step 0, and the program is scanned from the beginning again.

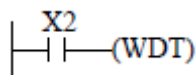
The subprogram called by the CALL command must be written after the FEND command, and the SRET command must be added at the end of the subprogram; the interrupt subprogram must also be written after FEND, and the IRET command must be added at the end of the interrupt program. In AutoShop, the main program, subprogram or interrupt program must be written in a separate window. There is no need to add FEND at the end of the main program, and there is no need to add SRET or IRET at the end of the subroutine or interrupt program.

Example:



16-bit	32-bit	P	WDT	The watchdog timer is reset.
✓		✓		
1 step		1 step	Separate instruction without operand	

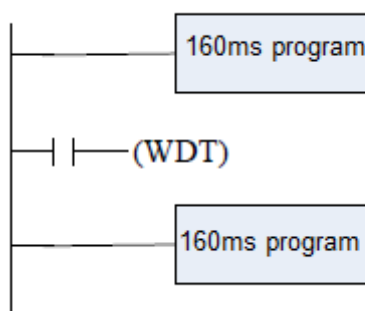
There is a timer in the PLC system to monitor whether the user program is executed once or not. If it times out, it will stop the execution of the user program and give an alarm. Executing the WDT instruction can reset the watchdog timer and restart the watchdog timer. Avoid timeout errors.



If the operation performed by the user program is too complicated (for example, too many loop calculations), a running timeout error may occur during execution. If necessary during programming, WDT instruction can be used (for example, insert this instruction between FOR~NEXT instruction);

If the scan time of the program is greater than the value of D8000 (default 200ms), you can insert a WDT instruction between the programs to divide each program into scan time less than 200ms or you need to modify the set value of D8000 as required.

Example:



The scan time of this program is 320ms, and the program is divided into 2 parts with the WDT instruction, so that the scan time of each part of the program is below 200ms.

16-bit	32-bit	P	FOR	Start of loop
✓				
3 steps			Instruction format: FOR S1	

Instruction	Bit elements				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The FOR instruction is used at the beginning of a loop and indicates the number of loop executions. It must be used in conjunction with the NEXT instruction. where:

S1 is the control variable of the number of cycles.

See the explanation and example of the NEXT instruction.

16-bit	32-bit	P	NEXT	End of loop
✓				
1 step			Instruction format: NEXT (no operand)	

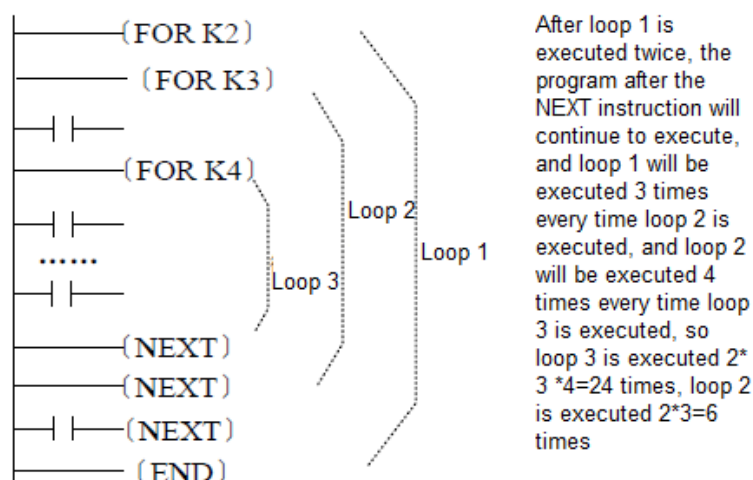
The NEXT instruction is used to indicate the end of the loop area. The FOR~NEXT loop is specified by the FOR instruction to execute N times back and forth, and then it jumps out of the FOR~NEXT loop to continue execution.

In the loop interval of the FOR~NEXT instruction, another FOR~NEXT loop can be embedded, but it is stipulated that from the outermost FOR~NEXT calculation, up to 6 levels of FOR~NEXT loops can be embedded. When running, PLC will analyse and execute according to each FOR~NEXT layer. However, it should be noted that when the number of cycles is too large, the PLC scan period will be prolonged, which may cause the overtime monitoring timer to operate and cause errors. It can be improved by using WDT instruction between FOR~NEXT instruction.

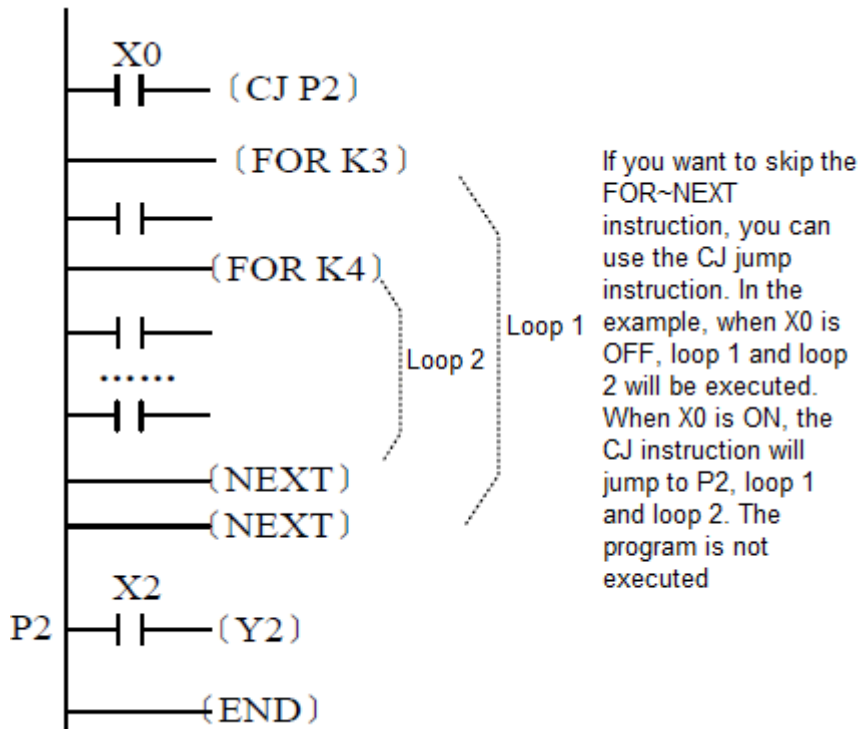
There will be errors in the following situations:

- The NEXT instruction comes before the FOR instruction;
- There is a FOR instruction but no NEXT instruction;
- There are NEXT instructions after FEND and END instructions;
- The number of FOR instructions and NEXT instructions does not match.

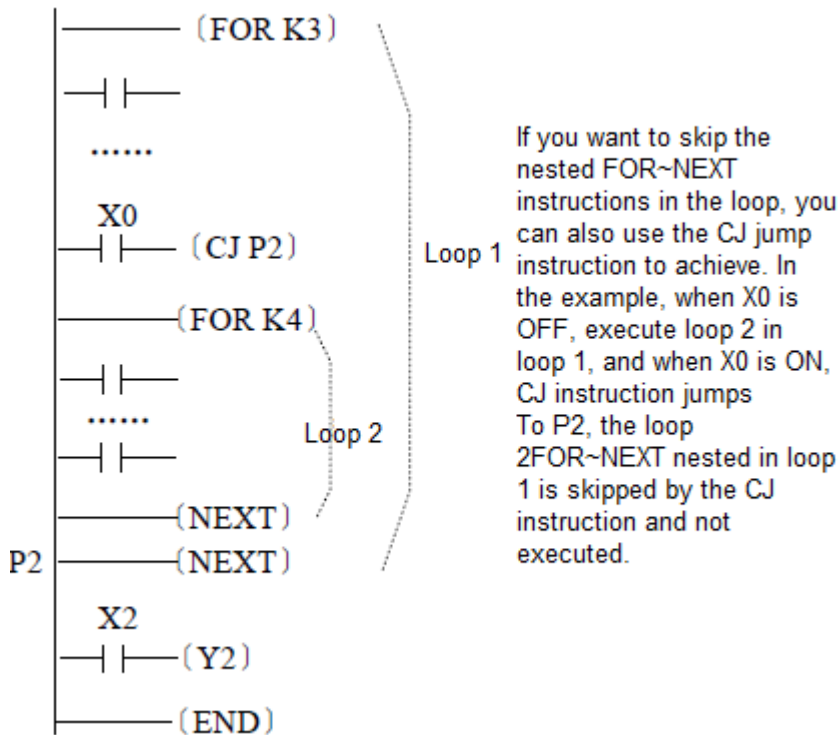
Example 1:



Example 2



Example 3:



## 4.3.2 Transmission and comparison (10~19)

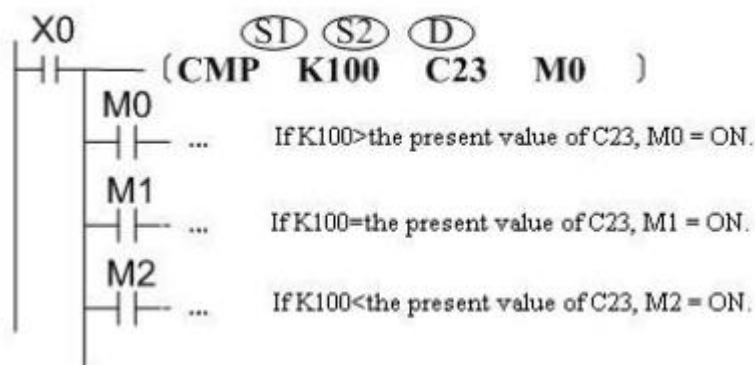
16-bit	32-bit	P	CMP	Data comparison
✓	✓	✓		
7 steps	13 steps		Instruction format: CMP S1 S2 D	

Operand	Bit elements				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D		✓	✓	✓												

This instruction compares two operational variables and outputs the comparison result to a specified bit variable. The operands are all algebra compared according to signed data.

Where, D will occupy consecutive 3 address bit variables.

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  $\geq$ ,  $\leq$  and  $\neq$  results.

16-bit	32-bit	P	ZCP	Range compare
✓	✓	✓		
9 steps	17 steps		Instruction format: ZCP S1 S2 S D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D		✓	✓	✓											

Need contact drive, there are 4 operating variables. When the control power flow is valid, perform algebraic comparison operations based on signed numbers, take S1 S2 as the interval, and store the value of S in the interval as the result, and store it in 3 consecutive bit variables with D as the starting address.

Where:

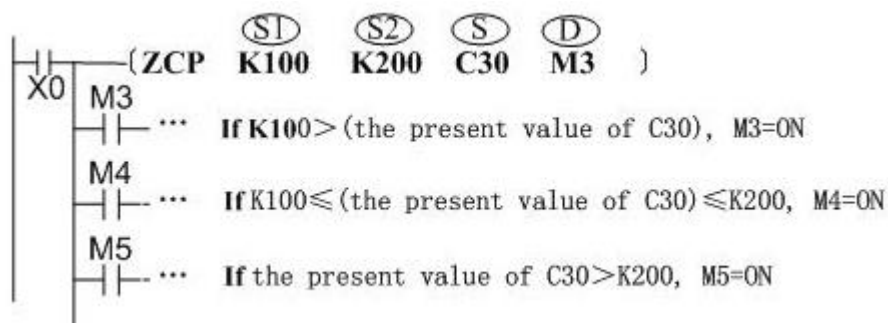
S1 is lower limit of comparison area

S2 is upper limit of comparison area

S is comparison variable

D is the storage unit for comparison result, occupying three continue bit variables.

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.

16-bit	32-bit	P	MOV	Value transfer
✓	✓	✓		
5 steps	9 steps		Instruction format: MOV S D	

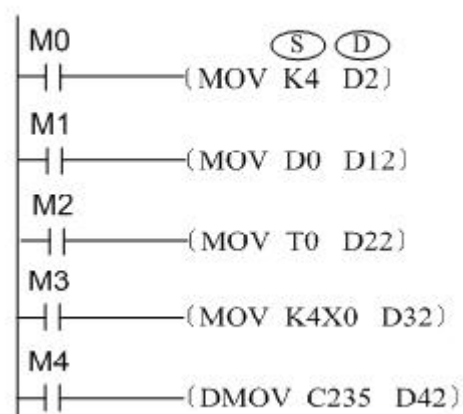
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

The instruction is driven by contact with two operation variables. S value is copied to D. When it is 32-bit instruction (DMOV), the corresponding S and D variable unit in high address will be involved in calculation.

Example code: the calculation result for (DMOV D1 D5) is: D1→D5; D2→D6

Example:

#### Ladder diagram



#### 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.

16-bit	32-bit	P	SMOV	Shift transfer
✓		✓		
11 steps		11 steps	Instruction format: SMOV S m1 m2 D n	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							✓	✓	✓	✓	✓	✓	✓	✓	✓
m1					✓	✓									
m2					✓	✓									
D								✓	✓	✓	✓	✓	✓	✓	✓
n					✓	✓									

The instruction is driven by contact with up to five operation variables, where:

S is the data source variable, which is to be copied;

m1 is the starting bit number for transmitting data source with range of 1~4;

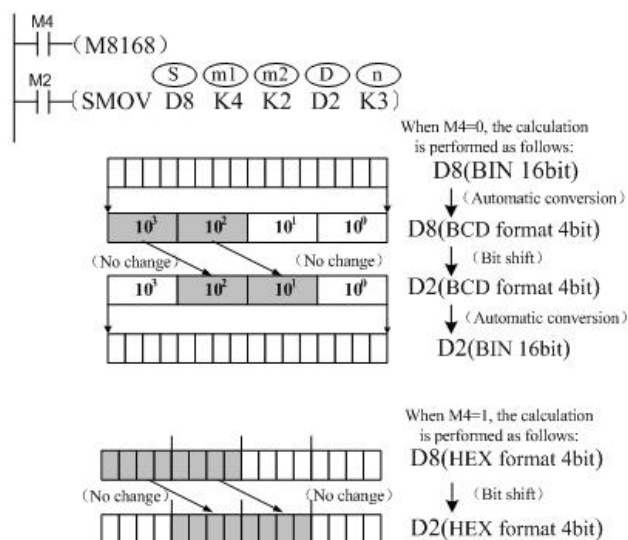
m2 is the bit number for transmitting data source with range of 1~m1;

D is the target variable for transmitting data source;

n is the starting bit of the target variable for transmitting data source with range of m2~4.

The data bit transmission processing is related with the state of special flag M8168. When M8168 is OFF, it is in BCD mode (decimal bit); when M8168 is ON, it is in BIN mode, in which 4-bit is taken as a unit for transmission (hexadecimal bit).

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.



16-bit	32-bit	P	CML	Inverted data transfer
✓	✓	✓		
5 steps	9 steps		Instruction format: SMOV S D	

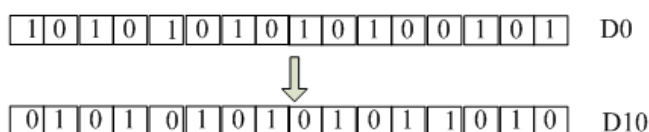
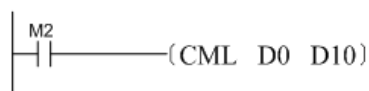
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

The instruction is driven by contact with two operation variables. S (BIN) value is converted in converse bit by bit and then copied to D.

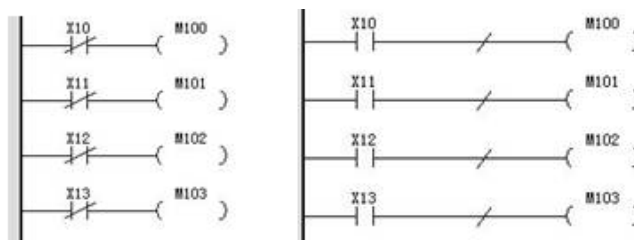
When D digit number is less than 16-bit, converting S in converse and transmitting to D variable with low bit alignment.

When it is 32-bit instruction (DCML), the corresponding S and D variable unit in high address will be involved in calculation. Example: the calculation result for [DCML D1 D5] is: /D1→D5; /D2→D6

Example 1:



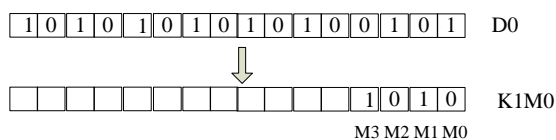
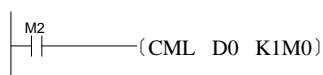
Example 2:



The above two program can be implemented with following CML instruction.



Example 3:



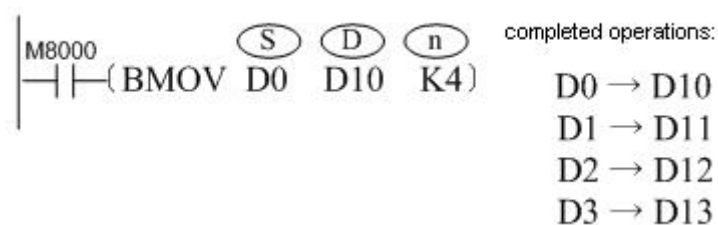
16-bit	32-bit	P	BMOV	Batch data transfer
✓		✓		
7 steps		7 steps	Instruction format: BMOV S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							✓	✓	✓	✓	✓	✓	✓		
D								✓	✓	✓	✓	✓	✓		
n	Constant, n=1~512														

The instruction is driven by contact with three operation variables. n variables with starting address specified by S are transmitted to n units with starting address specified by D.

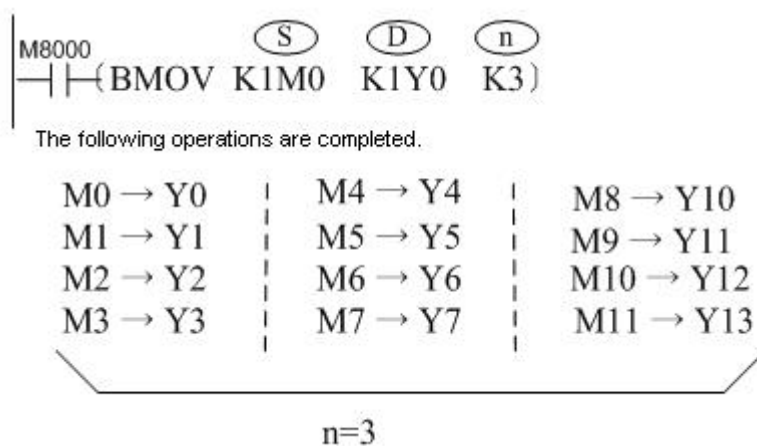
Where, n is within the range of 1~512.

When special variable M8024=1, the transmission direct is opposite, which means that n variables with starting address specified by D are transmitted to n units with starting address specified by S.



When operand is Bit element, the digit number of S and D should be same.

Example:



16-bit	32-bit	P	FMOV	One-to-multiple data transfer
✓	✓			
7 steps	13 steps		Instruction format: FMOV S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							✓	✓	✓	✓	✓	✓	✓		
D								✓	✓	✓	✓	✓	✓		
n	Constant, n=1~512														

The instruction is driven by contact with three operation variables. S data are transmitted to n units with starting address specified by D.

n is within the range of 1~512.

Example:



The following operation is completed  
when M8=ON.

k100 → D100

k100 → D101

k100 → D102

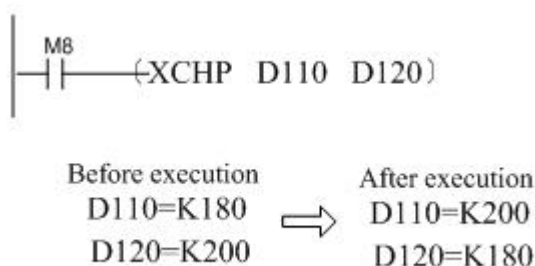
k100 → D103

16-bit	32-bit	P	XCH	Data exchange
✓	✓	✓		
5 steps	9 steps		Instruction format: XCH S D	

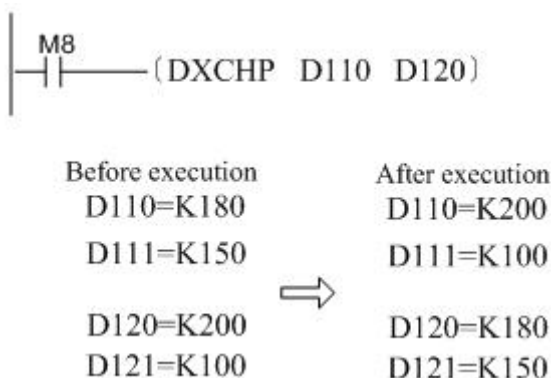
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								✓	✓	✓	✓	✓	✓		
D								✓	✓	✓	✓	✓	✓		

The XCH instruction requires contact driving and has two operands. It exchanges the values in S and D.

Example 1:



Example 2:



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.

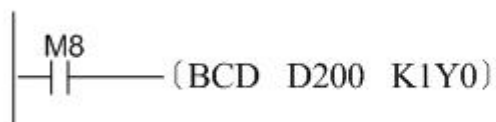
16-bit	32-bit	P	BCD	Conversion from binary to BCD format
✓	✓	✓		
5 steps	9 steps		Instruction format: BCD S D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

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).

16-bit	32-bit	P	BIN	Conversion from BCD to binary format
✓	✓	✓		
5 steps	9 steps		Instruction format: BIN S D	

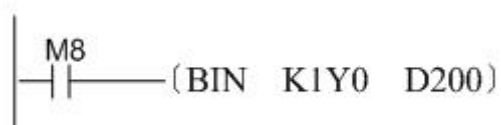
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

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 converted to a BIN equivalent. The result is stored in D200.

## 4.3.3 Four logical arithmetic operations (20~29)

16-bit	32-bit	P	ADD	BIN number addition
✓	✓	✓		
7 steps	13 steps		Instruction format: ADD S1 S2 D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

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 is 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.

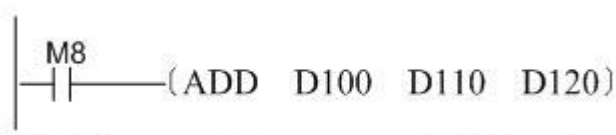
The zero flag M8020 is set if the result of the calculation is 0.

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).

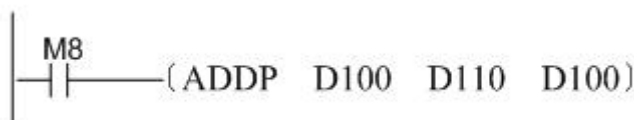
In 32-bit operation, the variable addresses in the ADD instruction contain the lower 16 bits, and the adjacent high-numbered address unit contains the higher 16 bits. Avoid repeated or overlapping addresses during programming.

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.

16-bit	32-bit	P	SUB	BIN number subtraction
✓	✓	✓		
7 steps	13 steps		Instruction format: SUB S1 S2 D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

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.

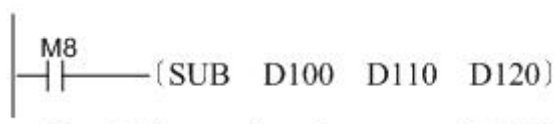
The zero flag M8020 is set if the result of the calculation is 0.

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.  
 $D120 = 10 - 8 = 2$  if  $D100 = 10$  and  $D110 = 8$ .

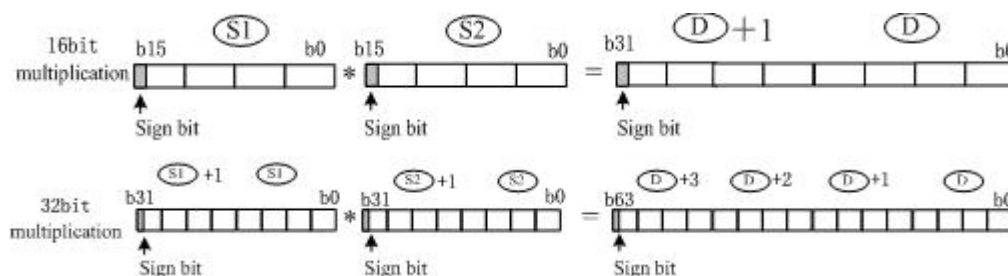


16-bit	32-bit	P	MUL	BIN number multiplication
✓	✓	✓		
7 steps	13 steps		Instruction format: MUL S1 S2 D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓		

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 is 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:



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.

16-bit	32-bit	P	DIV	BIN multiplication operation
✓	✓	✓		
7 steps	13 steps		Instruction format: DIV S1 S2 D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓		

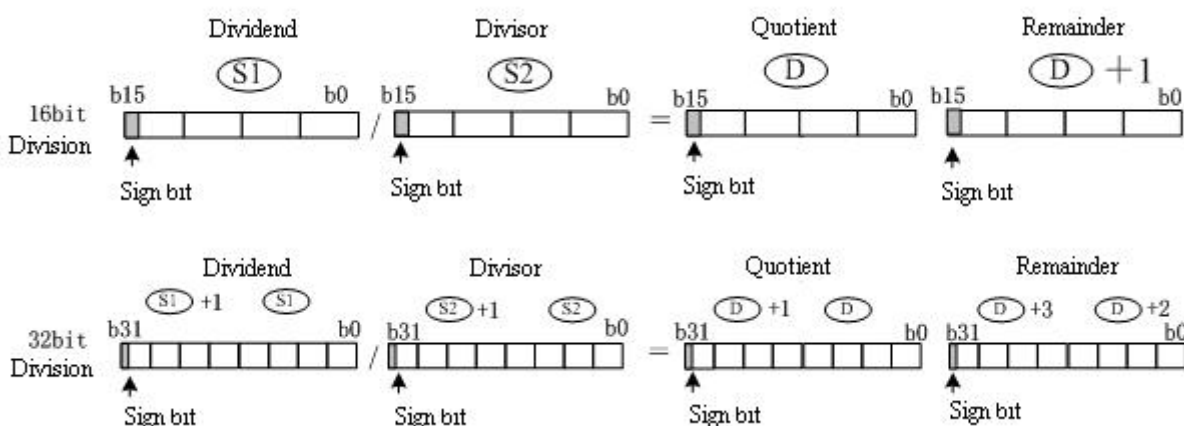
The DIV instruction requires contact driving and has three operands. Use this instruction to algebraically divide the value in S1 by that in S2 in BIN mode. The result is stored in D. The variables in the algebraic operation is processed as signed numbers. The highest bit is the sign bit. The value 0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

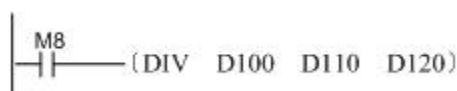
A calculation error will occur if the divider S2 is 0.

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:



When M8 is set, D100 (dividend) is divided by D110 (divisor).

The quotient is stored in D120.

If D100 = K5 and D110 = K2, the remainder is stored in D121 (= K1).

16-bit	32-bit	P	INC	BIN number incremented by 1
✓	✓	✓		
3 steps	5 steps		Instruction format: INC D	

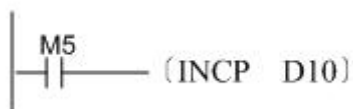
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								✓	✓	✓	✓	✓	✓	✓	✓

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.

Examples:



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

16-bit	32-bit	P	DEC	BIN number decremented by 1											
✓	✓	✓													
3 steps	5 steps		Instruction format: DEC D												
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								✓	✓	✓	✓	✓	✓	✓	✓

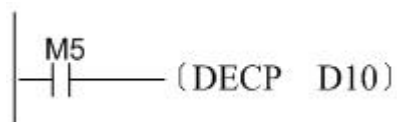
On every execution of the DEC instruction, D has its current value decremented by a value of 1.

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.

16-bit	32-bit	P	WAND	Binary number logical AND
✓	✓	✓		
7 steps	13 steps		Instruction format: WAND S1 S2 D (32-bit, DAND)	

The bit patterns of the BIN values in S1 and S2 are analysed. 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.

$$1 \cdot 1 = 1 \quad 1 \cdot 0 = 0 \quad 0 \cdot 1 = 0 \quad 0 \cdot 0 = 0$$

16-bit	32-bit	P	WOR	Binary number logical OR
✓	✓	✓		
7 steps	13 steps		Instruction format: WOR S1 S2 D (32-bit, DOR)	

The bit patterns of the BIN values in S1 and S2 are analysed. 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.

$$1 + 1 = 1 \quad 1 + 0 = 1 \quad 0 + 1 = 1 \quad 0 + 0 = 0$$

16-bit	32-bit	P	WXOR	Binary number logical XOR
✓	✓	✓		
7 steps	13 steps		Instruction format: WXOR S1 S2 D (32-bit, DXOR)	

The bit patterns of the BIN values in S1 and S2 are analysed. 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.

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

Applicable models of the above three instructions:

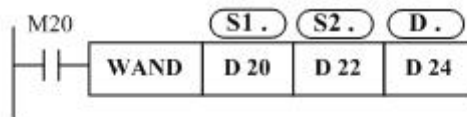
Applicable models		
Series	Universal	XP
H1U	✓	✓
H2U	✓	✓

The applicable variable types of above three instruction operands are shown in the following table. When it is a 32-bit instruction, the register variable occupies a total of 2 units of subsequent adjacent addresses:

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

Example:

Logic AND



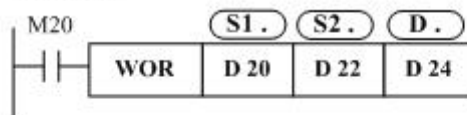
D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1

D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1



D24 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1

Logic OR



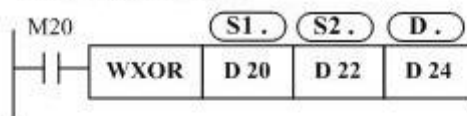
D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1

D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1



D24 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1

Logic Exclusive-OR



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

16-bit	32-bit	P	NEG	Binary number negation
✓	✓	✓		
3 steps	5 steps		Instruction format: NEG D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								✓	✓	✓	✓	✓	✓	✓	✓

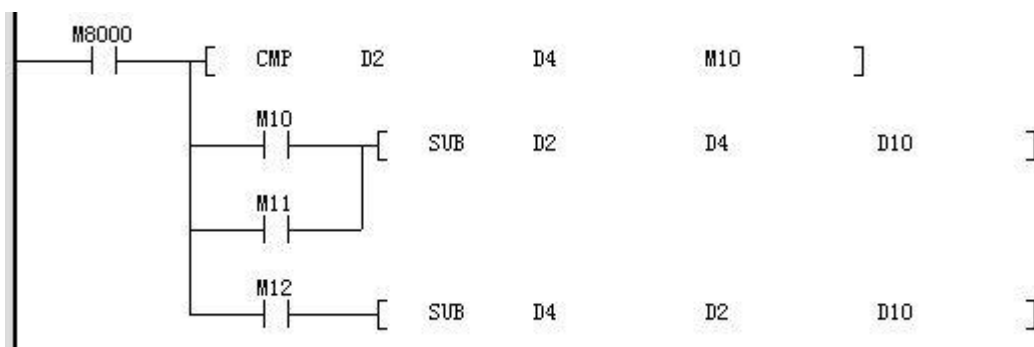
The NEG instruction requires contact driving and has one operand. It inverts the bit pattern of D, adds 1 to the bit pattern, and then writes the result to D.

The pulse execution type is generally used.

This gets the absolute value of a negative BIN number.

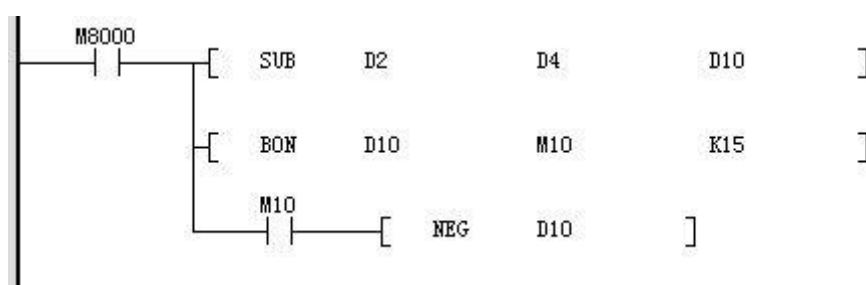
Example:

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.

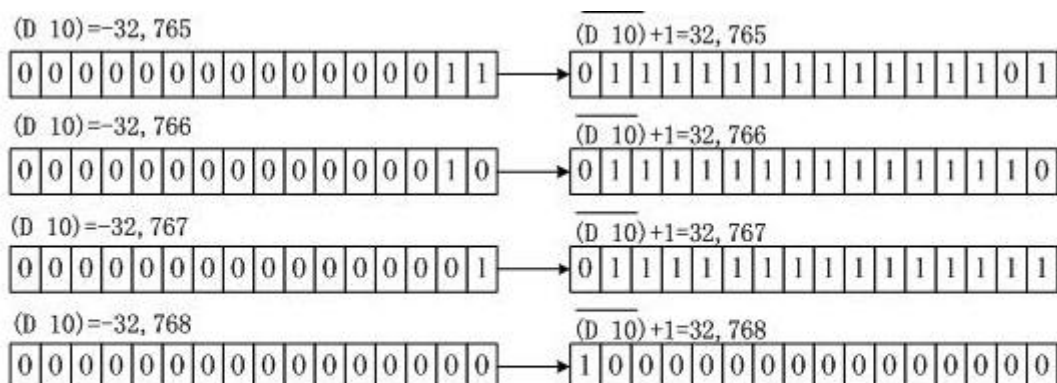
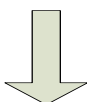
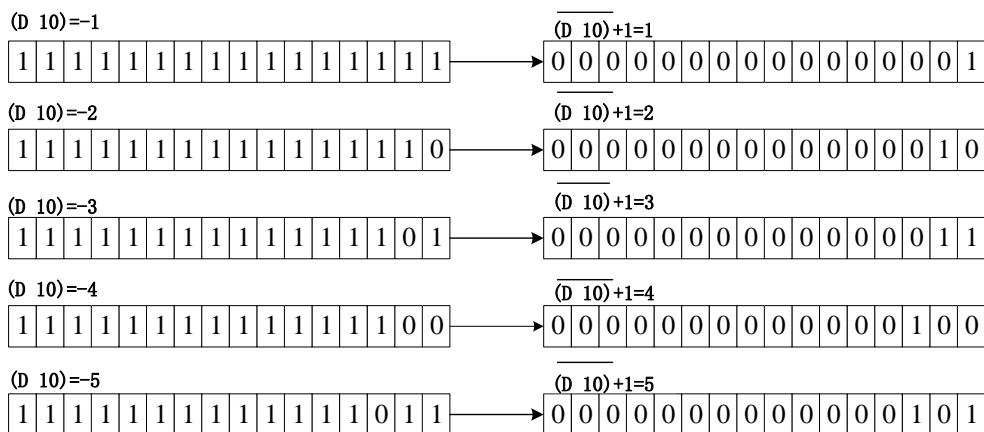
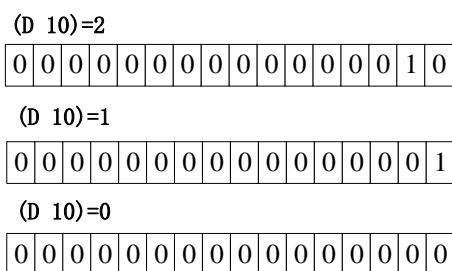
In the preceding examples, D10 = K4 if D2 = K4 and D4 = K8 or D2 = K8 and D4 = K4.

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.



Maximum absolute value can only be 32,767.



## 4.3.4 Cyclic shift (30~39)

16-bit	32-bit	P	ROR	Rotation right
✓	✓	✓		
5 steps	9 steps		Instruction format: ROR D n	

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.

16-bit	32-bit	P	ROL	Rotation left
✓	✓	✓		
5 steps	9 steps		Instruction format: ROL D n	

The bit pattern in D is rotated n bit places to the left. The instruction of the pulse execution type is generally

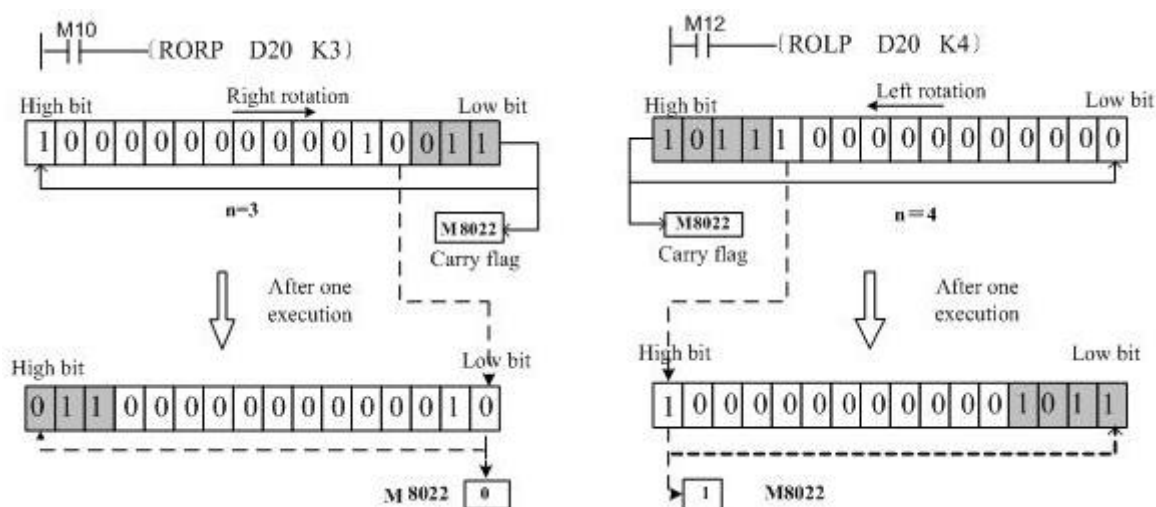
used. When the 32-bit instruction is executed, the register variable occupies two consecutive units.

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.

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(n)	Constant, n=1~16 (16-bit); n=1~32 (32-bit)														

Examples:



16-bit	32-bit	P	RCR	Rotation right with carry
✓	✓	✓		
5 steps	9 steps		Instruction format: RCR D n	

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.

16-bit	32-bit	P	RCL	Carry Cycle Left Shift
✓	✓	✓		
5 steps	9 steps		Instruction format: RCL D n	

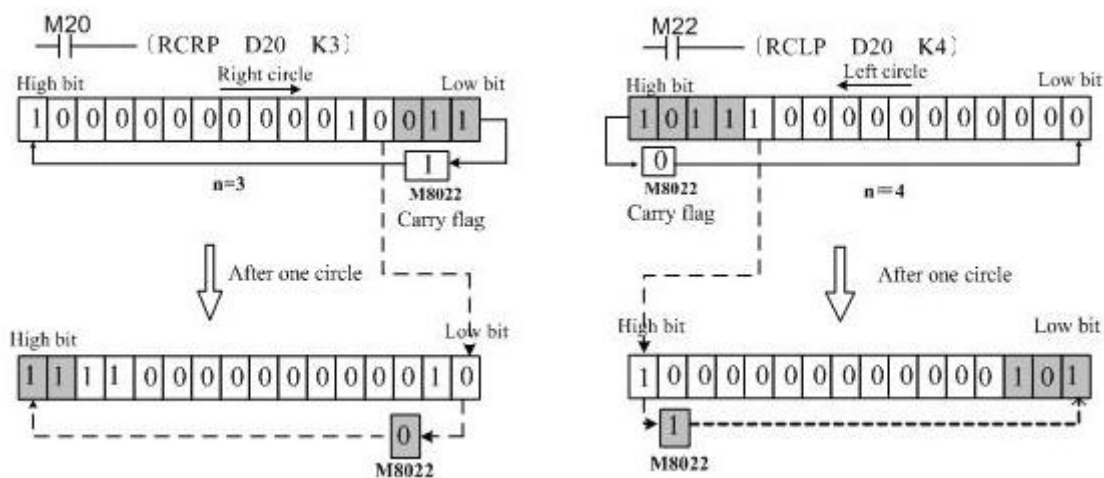
The bit pattern in D with the carry flag M8022 is rotated n bit places to the left. The instruction of the pulse execution type is generally used.

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.

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
n	Constant, n=1~16 (16-bit) ; n=1~32 (32 bit)														

Example:



16-bit	32-bit	P	SFTR	Bit shift right
✓		✓		
9 steps		9 steps	Instruction format: SFTR S D n1 n2	

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.

16-bit	32-bit	P	SFTL	Bit shift left
✓		✓		
9 steps		9 steps	Instruction format: SFTL S D n1 n2	

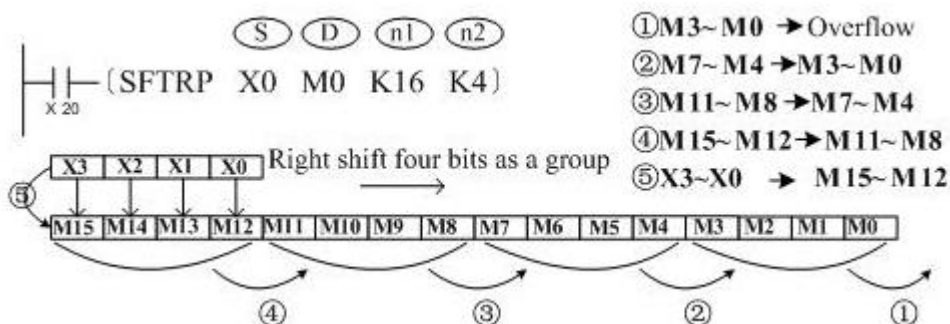
n1 bit variables from head address D are shifted n2 bit places to the left, to accommodate n2 bit variables from head address S that fill the lower bits.

The instruction of the pulse execution type is generally used.

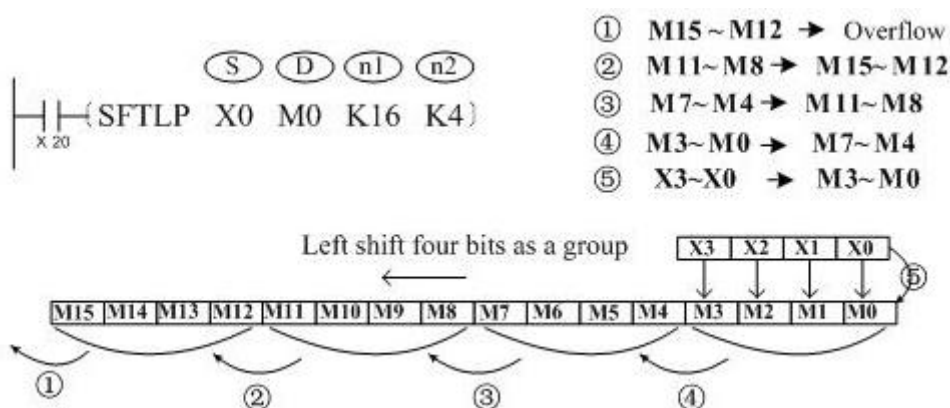
The variable types applicable to the operands of SFTR and SFTL are as follows:

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓	✓	✓	✓											
D		✓	✓	✓											
n1	Constant, n1 ≤ 1024														
n2	Constant, n2 ≤ n1														

SFTR command:



SFTL command:



16-bit	32-bit	P	WSFR	Word shift right
✓		✓		
9 steps		9 steps	Instruction format: WSFR S D n1 n2	

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.

16-bit	32-bit	P	WSFL	Word shift left
✓		✓		
9 steps		9 steps	Instruction format: WSFL S D n1 n2	

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.

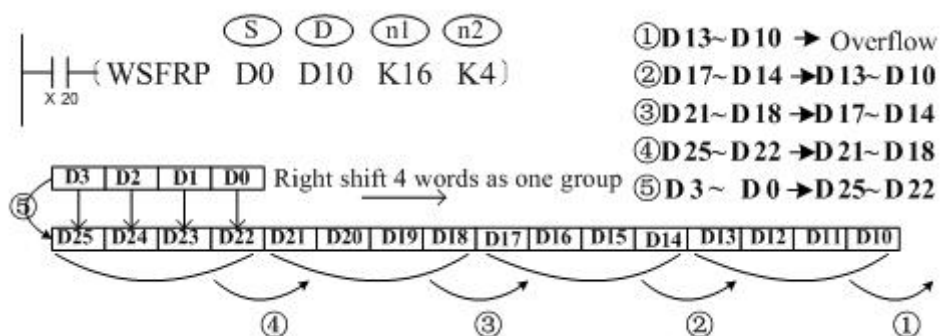
The instruction of the pulse execution type is generally used.

The applicable variable types of the operands of the above two instructions are as follows:

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✓	✓	✓	✓	✓	✓	✓		
(D)								✓	✓	✓	✓	✓	✓		
(n1)	Constant, H1U model: $n1 \leq 2048$ ; H2U model: $n1 \leq 2048$														
(n2)	Constant, $n2 \leq n1$														

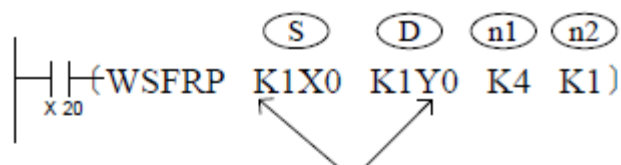
Example 1:

WSFR command:

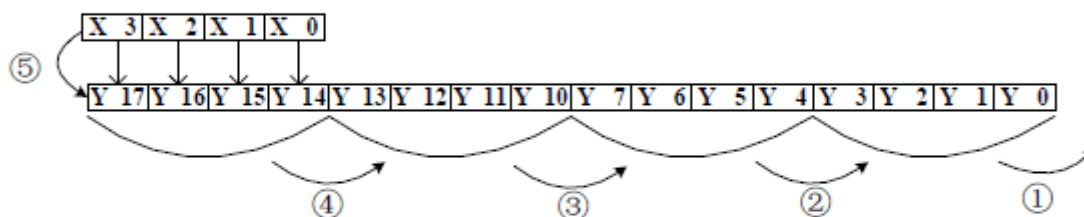


Example 2:

WSFR command:



Need to specify the same number of bits when using Kn type devices

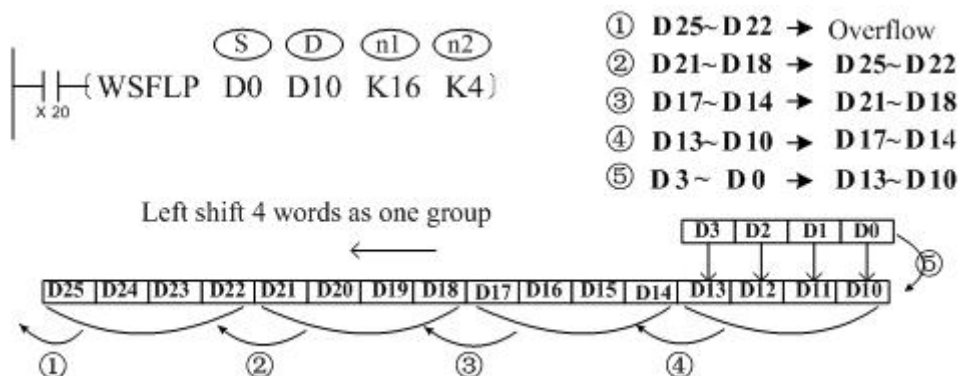


The number of digits scanned once is moved to the right to act according to the following numbers 1~5.

- 1: **Y3~Y0** → Carry forward
- 2: **Y17~Y14** → **Y13~Y10**
- 3: **Y13~Y10** → **Y7~Y4**
- 4: **Y7~Y4** → **Y3~Y0**
- 5: **X3~X0** → **Y17~Y14** Complete

Example 3:

WSFL command:



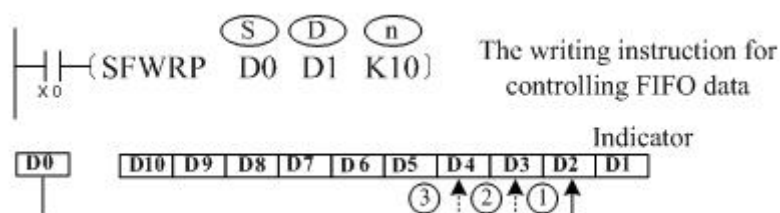
16-bit	32-bit	P	SFWR	"FIFO" data write
✓		✓		
7 steps		7 steps	Instruction format: SFWR S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓		
D								✓	✓	✓	✓	✓	✓		
n	Constant, $2 \leq n \leq 2048$ (should not exceed the D range)														

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.

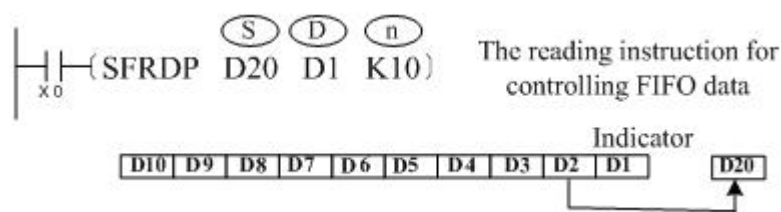
16-bit	32-bit	P	SFRD	"FIFO" data read
✓		✓		
7 steps		7 steps	Instruction format: SFRD S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓		
D								✓	✓	✓	✓	✓	✓		
n	Constant, $2 \leq n \leq 2048$ (should not exceed the D range)														

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.

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.

## 4.3.5 Data processing (40~49)

16-bit	32-bit	P	ZRST	Full data reset
✓		✓		
5 steps		5 steps	Instruction format: ZRST D1 D2	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D1		✓	✓	✓							✓	✓	✓		
D2		✓	✓	✓							✓	✓	✓		

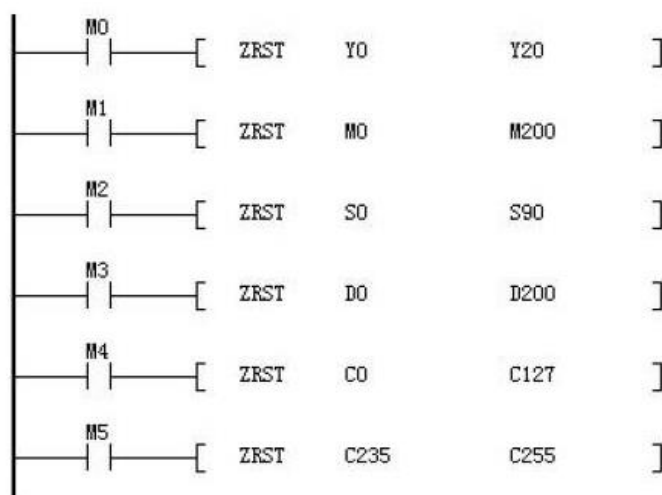
The values of all variables between D1 and D2 are cleared. D1 and D2 can be specified as word variables or Y, M, and S bit variables.

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.

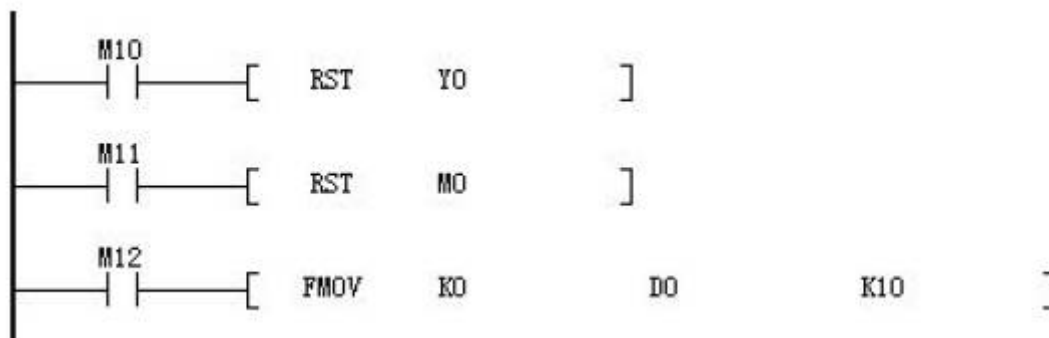
The ZRST instruction is a 16-bit instruction, but 32-bit counters can be designated for D1 and D2. Ensure that they use counters of the same bit type.

Examples:



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:





16-bit	32-bit	P	DECO	Data decoding
✓		✓		
7 steps		7 steps	Instruction format: DECO S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓	✓
D		✓	✓	✓							✓	✓	✓		
n	Constant, n=1~8. If n=0, the instruction will not be executed; other values will cause an error in execution.														

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.

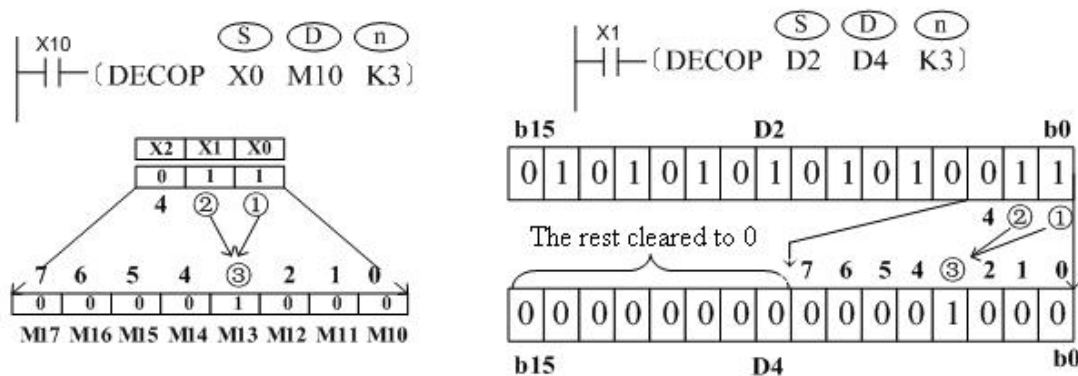
When n = 0, the instruction is not executed. When n is out of the range 0 to 8, an operation error will occur.

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:



16-bit	32-bit	P	ENCO	Data encoding
✓		✓		
7 steps		7 steps	Instruction format: ENCO S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓	✓
D		✓	✓	✓							✓	✓	✓		
n	Constant, n=1~8. If n=0, the instruction will not be executed; other values will cause an error in execution.														

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.

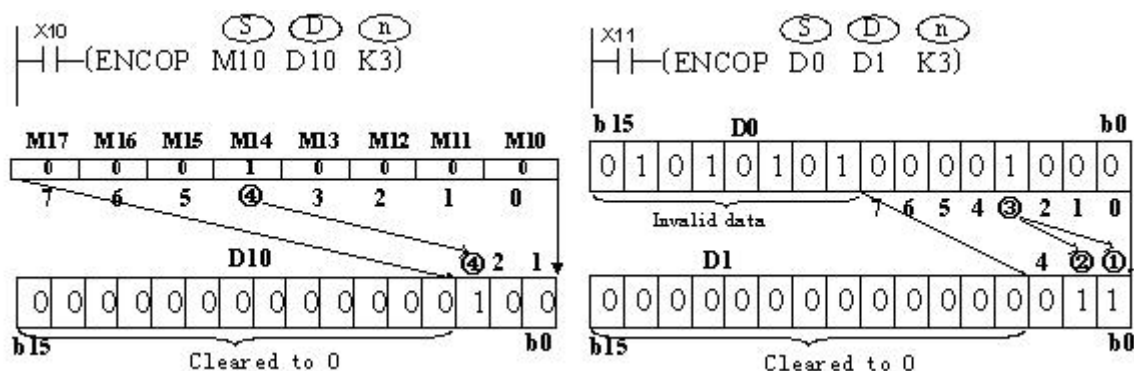
When all bits from head address S are 0, an operation error will occur.

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:



16-bit	32-bit	P	SUM	Total number of ON bits
✓	✓	✓		
5 steps	9 steps		Instruction format: SUM S D	

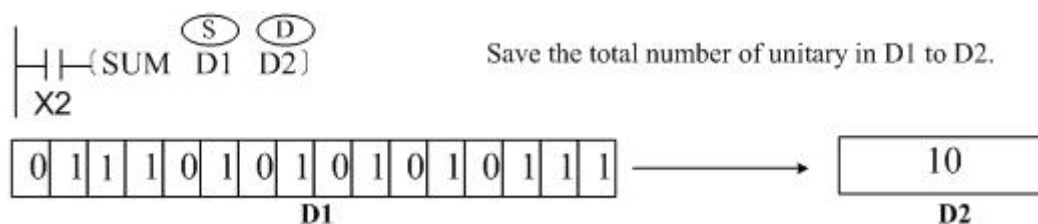
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

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:

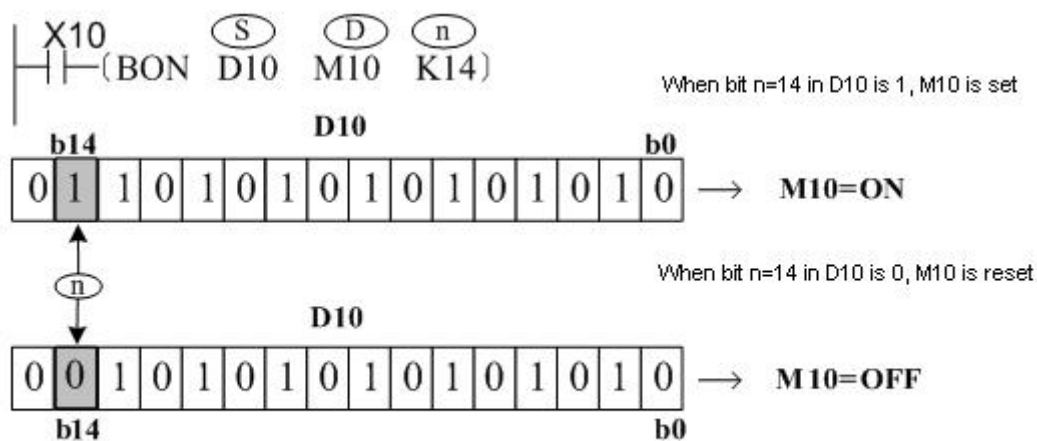


16-bit	32-bit	P	BON	ON bit check
✓	✓	✓		
5 steps	9 steps		Instruction format: BON S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D		✓	✓	✓											
n	n = 0~15 (16-bit); n = 0~31 (32-bit)														

The status of the nth bit in S is checked. The result is stored in D.

Example:



M10 remains the previous state when X10 is changed from ON to OFF.

16-bit	32-bit	P	MEAN	Mean value calculation
✓	✓	✓		
7 steps	13 steps		Instruction format: MEAN S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							✓	✓	✓	✓	✓	✓	✓		
D								✓	✓	✓	✓	✓	✓	✓	✓
n	Constant, n = 1-64, otherwise an error will occur.														

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.

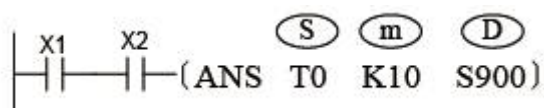
16-bit	32-bit	P	ANS	Annunciator setting
✓				
7 steps			Instruction format: ANS S m D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											✓				
m				✓											
D	Constant, m=1~32767, (unit: 100ms).														

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.

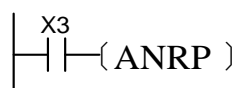
Note: The following table lists related elements.

Soft element	Name	Content
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.

16-bit	32-bit	P	ANR	Annunciator reset
✓		✓		
1 step		1 step	Instruction format: ANR (no operand)	

When driving conditions are met, the ANR instruction resets the annunciators in the range S900 to S999 in batches.

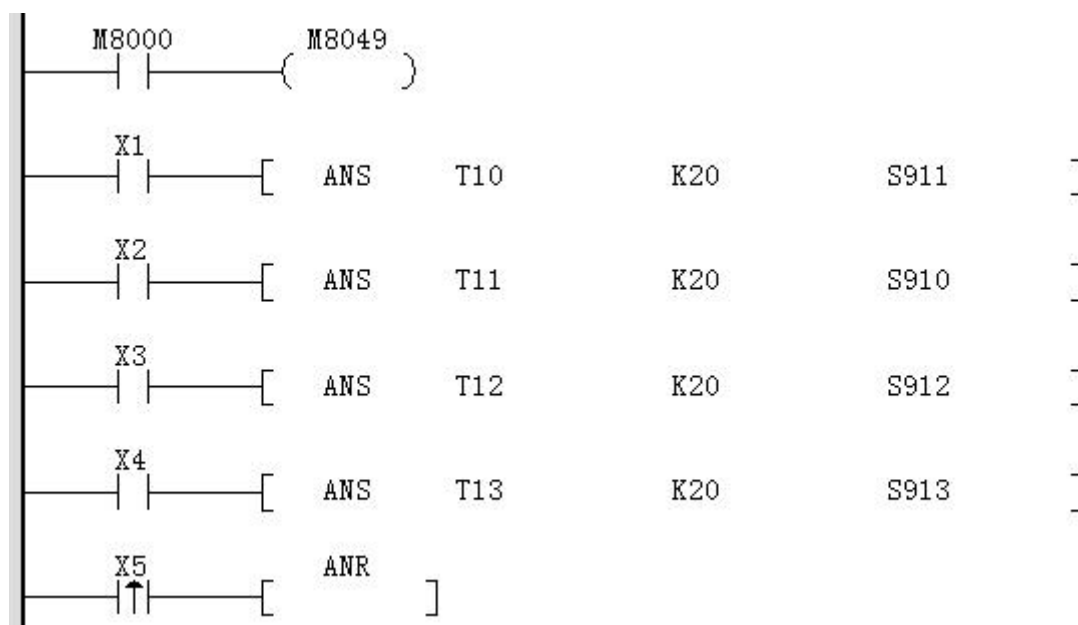
Example:



If X3 is turned on, the active alarm points of the signal alarms S900~S999 are reset. If multiple alarm points act at the same time, the alarm point with the smallest number of ON will be reset.

If X3 is turned on again, the state of the next number is reset. The ANRP instruction is often used in actual use.

Example:



M8049 is ON. When any one of S900~S999 is ON in the program, M8048 will be ON.

If S910, S911, S912, and S913 are all ON in the program, when X5 turns ON from OFF for the first time, S910 is reset, and when X5 turns ON for the second time, S911 is reset, and so on.

Note: Related software components

Soft element	Name	Content
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.

16-bit	32-bit	P	SQR	Binary number square root operation
✓	✓	✓		
5 steps	9 steps		Instruction format: SQR S D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓							✓		
D													✓		

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:

$$\begin{array}{c} \text{X2} \\ | \text{---} | \text{---} | \text{---} | \text{---} | \\ \sqrt{D0} \rightarrow D12 \end{array}$$

If D0 = K100, then D12 = K10 when X2 = ON.

If D0 = K110, then D12 = K10 when X2 = ON and the fractional part is discarded.



16-bit	32-bit	P	FLT	Conversion from binary integer format to binary floating-point number format
✓	✓	✓		
5 steps	9 steps		Instruction format: FLT S D	

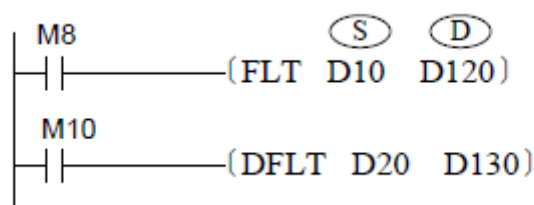
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S													✓		
D													✓		

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:



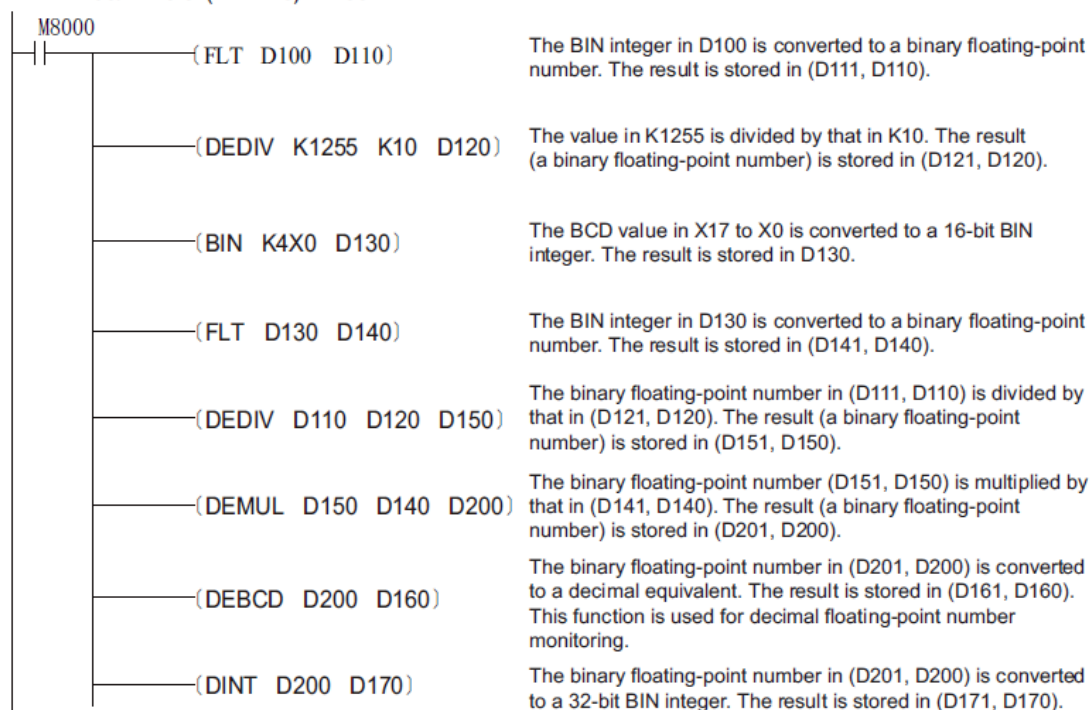
When M8 = ON, the 16-bit BIN integer in D10 is converted to a binary floating-point number. The result is stored in (D121, D120).

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.

$$D100/K125.5*(X17-X0)=D200$$



## 4.3.6 High-speed processing (50~59)

16-bit	32-bit	P	REF	Input/output refreshing
✓		✓		
5 steps		5 steps	Instruction format: REF D n	

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:

Operands	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
D	✓	✓														
n	Constant, n=8~256, it must be multiple of 8															

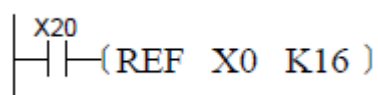
The address S must be an element number whose lowest bit is 0, such as X0, X10, .....Y0, and Y10. The value of n must be a multiple of 8 (in the range 8 to 256).

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 REFF instruction) within the range 0ms to 60ms. Other I/O ports use hardware filter with a filter time of about 10ms.
- 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 0.5ms.

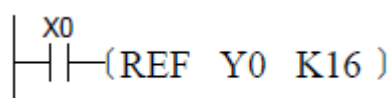
For specific parameters, please refer to the user manual of the programmable controller.

Example 1:



When the above program is implemented, if X20=ON, the input point state of X0~X17 will be read and the input signal will be updated without input delay.

Example 2:



When the above program is implemented, if X0=ON, immediately the state of Y0~Y17 will be refreshed and the output signal will be updated without waiting for END instruction.

16-bit	32-bit	P	REFF	Input filter time adjustment
✓		✓		
3 steps		3 steps	Instruction format: REFF n	

Operands	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
n	Constant, n = 0~60, Unit:ms															

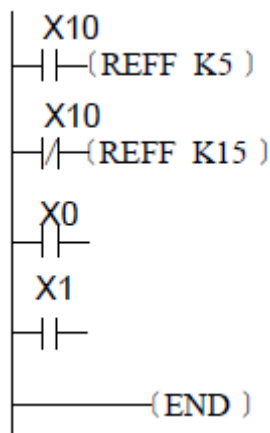
The filter time constants of X0 to X7 input ports are set to n ms.

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 0ms to 60ms. Other X ports support only hardware RC filter. The filter time constant is about 10ms, 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 is ON, set the input filter time of X0~X7 is set to 5ms, and when X10 is OFF, set the input filter time of X0~X7 to 15ms.

16-bit	32-bit	P	FNC	MTR	Input matrix
✓		✓	52		
9 steps		9 steps	Instruction format: MTR S D1 D2 n		

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓														
D1		✓													
D2		✓	✓	✓											
n	Constant, n = 2~8														

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, where:

S is the starting address of the hardware X port of the matrix scan input. It is required to be X0, X10... and other numbered components with the lowest bit of 0, occupying 8 consecutives;

D1 is the starting address of the hardware Y port of the matrix scan output. It is required to be Y0, Y10... and other numbered components with the lowest bit being 0, occupying n consecutive (n=2~8);

D2 is the starting address of the storage unit in the matrix scanning read state, and it is required to be Y0, M0, S0 and other numbered components with the lowest bit of 0;

n is the number of columns scanned by the matrix, that is the number of Y outputs for scanning

The conditional contact of this instruction generally uses the normally-on contact M8000.

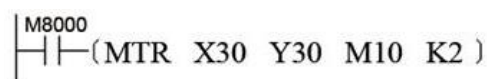
Example:

Considering the 10ms response delay of the X input filter, the output of Y30 and Y31 is interrupted every 20ms to enable instant input/output processing.

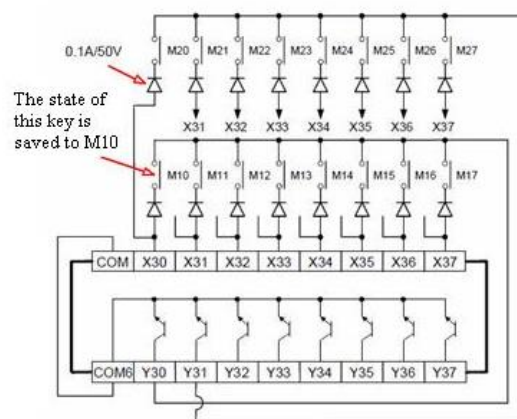
The M8029 flag is set to ON after every automatic read operation is completed.

A maximum of 64 scan inputs can be acquired using eight X inputs and eight transistor Y outputs. However, it takes 160ms (= 20ms x 8 columns) to read all inputs, which is impractical for high-speed input.

Therefore, ports after X20 are typically used for scan input.



The wiring diagram is as follows:



16-bit	32-bit	P	FNC	HSCS	High-speed counter comparison setting
	✓				
	13steps		Instruction format: HSCS S1 S2 D		

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2												✓			
D		✓	✓	✓											

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.

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 (10ms) or transistor (10us).

#### Instruction description:

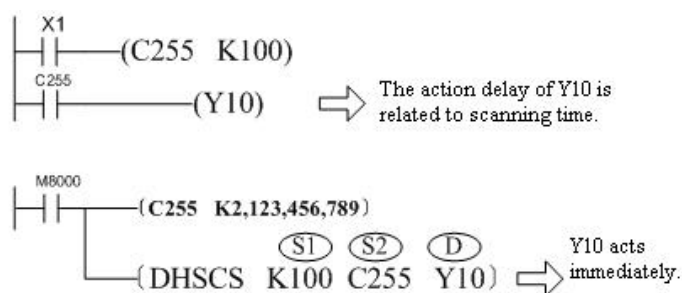
- Before the HSCS instruction is executed, the counter used must have been enabled (see Example 1); otherwise, the value of the counter will not change.
- The counter responds to input signals in interrupt mode and timely compare values. If the compared values are matched, the comparison output is immediately 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.

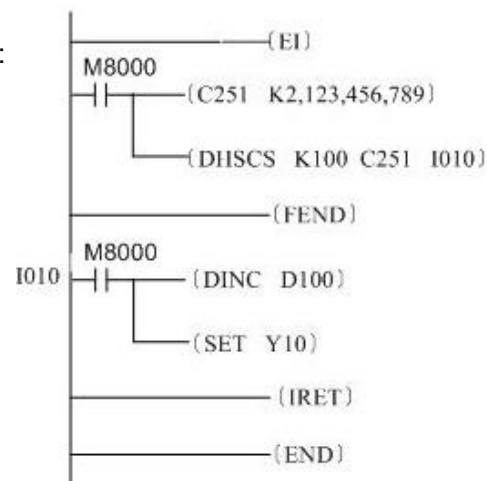
High-speed counting interrupt pointer and settings:

Input number	Interrupt disable instruction
I010	M8059
I020	
I030	
I040	
I050	
I060	

Example 1:



Example 2:



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.

16-bit	32-bit	P	HSCR	High-speed comparison counter reset
	✓			
	13steps		Instruction format: HSCR S1 S2 D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2												✓			
D		✓	✓	✓											

When the present value of S2 counter is equal to the value of S1, immediately reset [D].

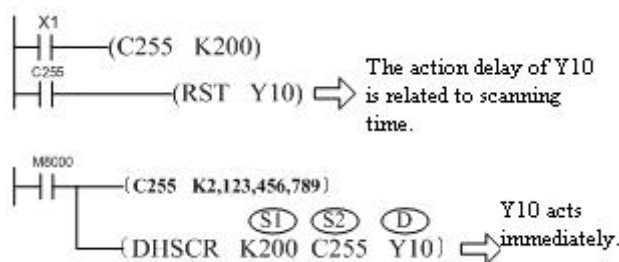
S2 variables must be high-speed counters C235–C255. As all counters involved are 32-bit counters, the 32-bit instruction DHSCR must be used.

[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.

Example:



### Instruction description

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.

16-bit	32-bit	P	HSZ	High-speed counter range comparison
	✓			
	17 steps		Instruction format: HSZ S1 S2 S D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S												✓			
D		✓	✓	✓											

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,  $[S1] \leq [S2]$ .

[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,  $[S1] \leq [S2]$ .

[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.

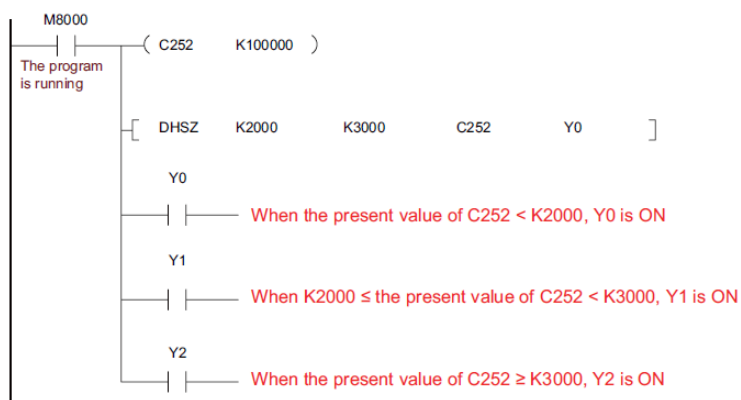
[D] is the head address of three consecutive storage units for the comparison result: when it is Y0 to Y17 port, the result is immediately output; when it is a port with the number greater than Y20, the result is output after the user program is scanned; for M, S, or SM variable, the result is immediately refreshed.

Note:

The action principle of this instruction is like that of the HSCS and HSCR instructions. The difference is that two comparison values are used, and three consecutive address units are used after comparison. See the description of the HSCR instruction. The HSZ instruction is executed in interrupt mode. Values are compared and the corresponding comparison result is refreshed only when counting pulses are input to the counter.

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:

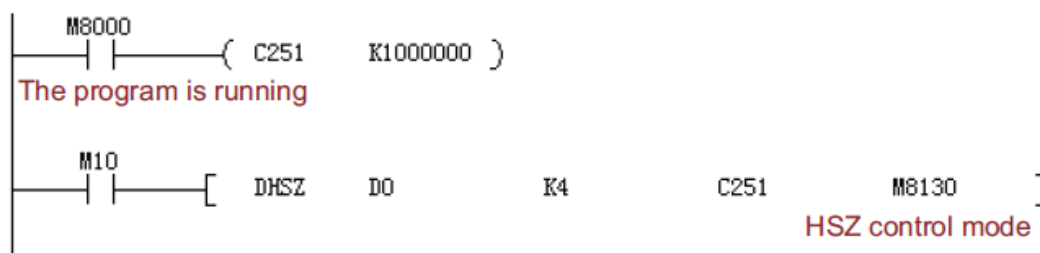
[S1] only corresponds to variables of register D and indicates the head address of the comparison table. Z can be used. After the instruction is enabled, [S1] will no longer be affected by Z.

[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.

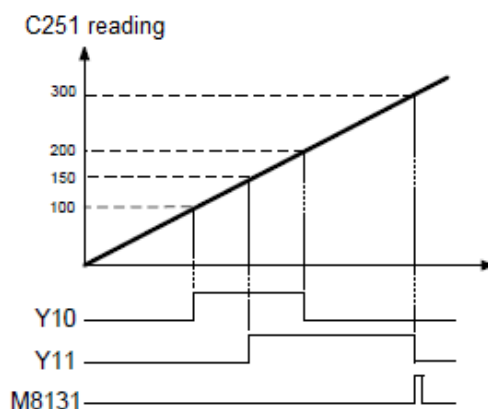
Example:



The equivalent comparison table is:

The initial variable of [S1] table is D0	Comparison value (32-bit)	Y output number	ON/OFF status	Table counter D8130
	(High-order, low-order) byte			
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.	

The explanation of the implementation:



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.

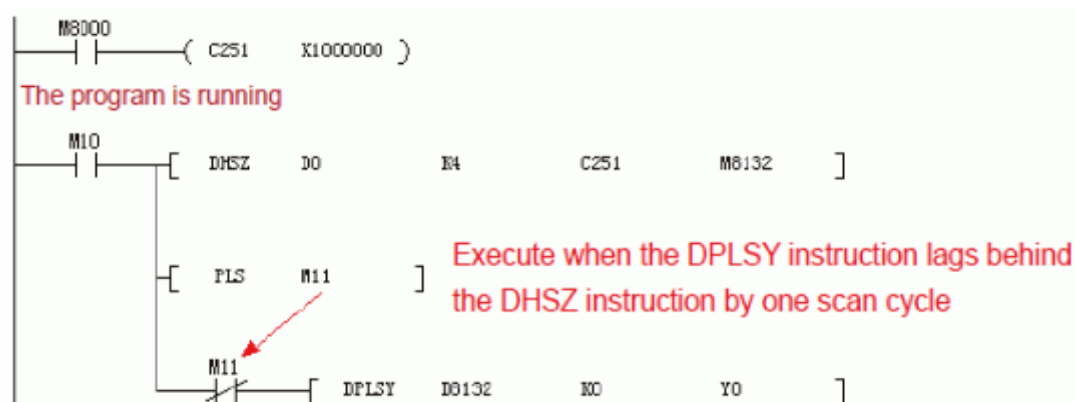
[S2] can use the constant K or H only to indicate the number of rows of the table.  $1 \leq (K \text{ or } H) \leq 128$ . V can be used. After the instruction is enabled, [S1] will no longer be affected by V or Z.

[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.

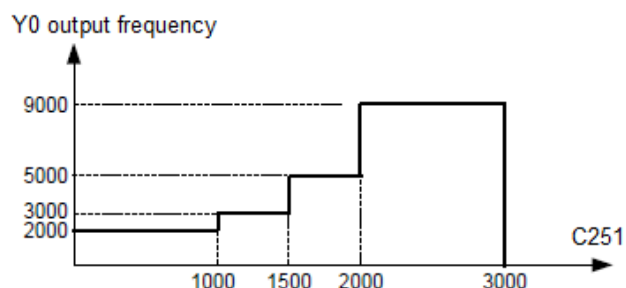
Example:



The 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-bit)	Y0 output frequency	Table counter D8131
	(High-order, low-order) byte		
The number of rows of [S2] table is 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 port is set with the preset value in corresponding table column.	

The explanation of the implementation:



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  $0 \leq C251 < (D1, D0)$ , the Y0 output frequency is equal to the value of (D3, D2);

When  $(D1, D0) \leq C251 < (D5, D4)$ , the Y0 output frequency is equal to the value of (D7, D6);

When  $(D5, D4) \leq C251 < (D9, D8)$ , the Y0 output frequency is equal to the value of (D11, D10);

When  $(D9, D8) \leq C251 < (D13, D12)$ , the Y0 output frequency is equal to the value of (D15, D14);

When  $(D13, D12) \leq C251 < (D17, D16)$ , 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 M10] 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.

16-bit	32-bit	P	SPD	Pulse density detection
✓				
7 steps			Instruction format: SPD S1 S2 D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1	✓														
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D											✓	✓	✓	✓	✓

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 X05.

[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]

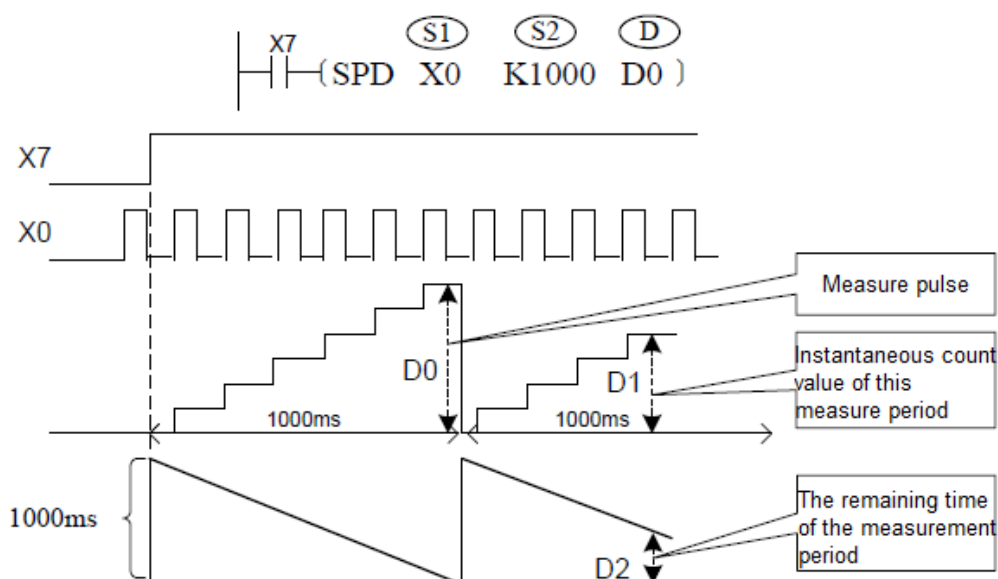
indicates the number of pulses within the set duration [S2]. [D + 1] indicates the real-time number of

pulses.

[D + 2] indicates remaining time of the sampling period.

X00 to X05 ports used for the SPD instruction can be used for high-speed counters or interrupt inputs.

Example 1:



In the figure above, when X7 turns ON, D1 counts the OFF→ON actions of X0, and stores the result in D0 after 1000ms, and then resets D1 and counts the actions of X0 again. D2 is used to determine the remaining time.

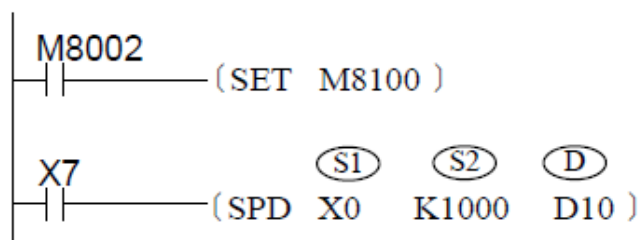
Therefore, the pulse frequency can be obtained according to the setting values of D0 and S2; if the pulse signal is taken from a rotary encoder, the speed can be obtained;

The maximum input of SPD speed measurement is 10K.

S1 pulse signal input port, only X0~X5;

- S2 is the set pulse detection time length (ms) 1~32767;
- D +0: is the number of pulses in time, which is 16-bit data;
- D +1 is the real-time pulse count value.
- D +2 is the remaining time to complete the sampling period.

Example 2:



D10 is the number of pulses received within the set time length of 1000ms; D11 is the real-time pulse count value; D12 is the remaining time to complete the sampling period.

Note: Enhanced function of SPD instruction (not available in XP type)

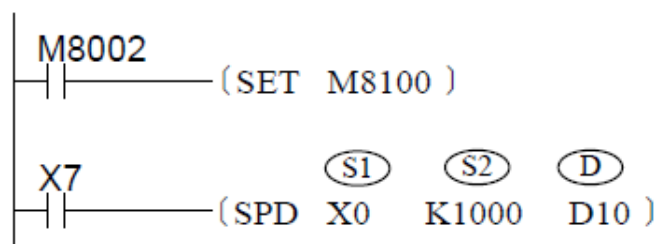
In the new version of the H1U/H2U series PLC, the SPD command function is enhanced. By setting the [M8100~M8105] flag, the counting function can be changed. The M8100~M8105 flags correspond to the X000~X005 high-speed terminals respectively. (The M8100~M8105 flags can be set separately, that is, the SPD command function of the X0~X5 ports can be processed separately.)

If the function enable flag [M8100~M8105] is OFF, the SPD instruction is as described above;

If the function enables flags M8100~M8105 are ON, the new function of the instruction is defined as follows:

- S1: Pulse signal input port, only X0~X5;
- S2: the set pulse detection time length (ms) 1~32767;
- D +0: is the number of pulses within S2 time, which is 16-bit data;
- D +1, D +2: the number of pulses per minute, which is 32-bit data;

Example 3:



D10 is the pulse number received in preset time (1000ms); D11, D12 is the pulse number \* 10 (the unit is 0.10 in the operation frequency = 1 min).

16-bit	32-bit	P	PLSY	Pulse output
✓	✓			
7steps	13steps		Instruction format: PLSY S1 S2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D		✓														

Because the relay is not suitable for high frequency action, only transistor output type PLC is suitable using this instruction. The instruction function is made up of:

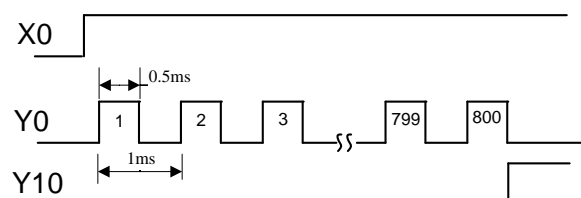
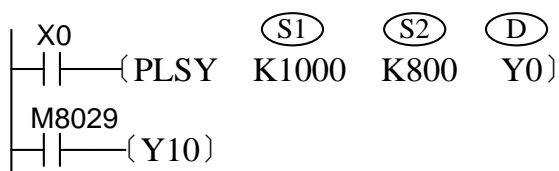
The port designated by D outputs S2 pulses at the frequency of S1. After the pulse is sent, the M8029 flag is set. where:

D is the pulse output port, only Y0/Y1/Y2 can be specified.

S1 is the set output pulse frequency. For 16-bit instructions (PLSY), the setting range is 0~32,767 (note that there is no pulse output when set to 0); for 32-bit instructions (DPLSY), the setting range is 1~100,000 (i.e. 1Hz~100kHz); The value of S1 can be changed during the execution of the instruction;

S2 is the set number of pulse outputs. For 16-bit instructions (PLSY), the setting range is 1~32,767; for 32-bit instructions (DPLSY), the setting range is 1~2,147,483,647; when S2 is equal to zero, to send uninterrupted infinite pulses.

Example:



When using PLSY (16-bit instruction), both S1 and S2 can only be 16-bit width;

When using DPLSY (32-bit instruction), if S1 and S2 are D, C, T variables, they will be processed as 32-bit width;

When the PLSY instruction is executed, Y starts to output pulses; if the parameter value of the S1 element (D, C, T variables) is changed during operation, it has no effect on the current output pulse number, and it will take effect the next time the instruction is started ; (In the new version of H2U series PLC, the number of output pulses can be changed during operation, please refer to Appendix 8.6 for details) During the PLSY output pulse process, if the command power flow X0 turns OFF, the output pulse is stopped; if X0 Turn ON, PLSY instruction will restart pulse output with current parameters.

**Instructions description:**

- The output Y port used by PLSY should not be duplicated with the Y port used by PWM or PLSR instructions;
- When using the PLSY instruction, the following special registers are used:

Register		Definition	Remarks
No.	Byte		
D8140	Lower	The total number of pulses, which is set in PLSY or PLSR instruction, is outputted to Y0 port	Available instruction: DMOV K0 D81xx for clear operation
D8141	Upper		
D8142	Lower	The total number of pulses, which is set in PLSY or PLSR instruction, is outputted to Y1 port	
D8143	Upper		
D8150	Lower	The total number of pulses, which is set in PLSY or PLSR instruction, is outputted to Y2 port (3624MT/2416MT doesn't feature this port)	
D8151	Upper		
D8152	Lower	The total number of pulses, which is set in PLSY or PLSR instruction, is outputted to Y3 port (available for MTQ)	
D8153	Upper		
D8154	Lower	The total number of pulses, which is set in PLSY or PLSR instruction, is outputted to Y4 port (available for MTQ)	
D8155	Upper		
D8136	Lower	Accumulative value of the number of the pulses already output to Y0 and Y1	
D8137	Upper		

**Notice:**

In the new version of the H1U/H2U series PLC, the function of the PLSY instruction has been enhanced. The number of pulses can be modified during the operation of the PLSY instruction, or the next pulse output instruction can be started immediately, or the pulse output completion interruption can be achieved.

- 1) By using special bits M8135~M8137 (corresponding to Y0~Y2 respectively) to be ON, the number of S2 output pulses (large or small) can be changed during operation;
- 2) By using the special bits M8085~M8087 (respectively corresponding to Y0~Y2) to be ON, the next pulse output command can be started immediately, without the need to deal with the invalidation of the previous power flow;
- 3) By using special bits M8090~M8092 (corresponding to Y0~Y2 respectively) to be ON, the pulse output completion interrupt can be realized; the specific correspondence is as follows:

Port number	Used special bit	Corresponding user interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

For more detail information, refer to appendix 8.6

16-bit	32-bit	P	PWM	Pulse width modulation
✓				
7steps			Instruction format: PWM S1 S2 D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D		✓													

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.

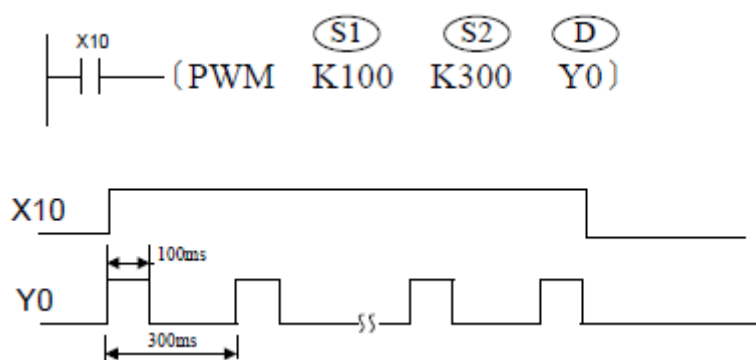
S2 indicates the pulse output period. S1 must be less than or equal to S2. The range is 1 to 32,767 ms.

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.

Example:



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 special registers corresponding to each output port are as follows:

Register	Pulse output unit
M8095	Y0 When using the PWM function, the output pulse unit is changed from ms to $\mu$ s.
M8096	Y1 When using the PWM function, the output pulse unit is changed from ms to $\mu$ s.
M8097	Y2 When using the PWM function, the output pulse unit is changed from ms to $\mu$ s.



16-bit	32-bit	P	PLSR	Pulse output with acceleration/deceleration
✓	✓			
9 steps	17 steps		Instruction format: PLSR S1 S2 S3 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S3					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D		✓														

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. Where:

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 100,000 Hz.

S2 indicates the number of pulses to be output. For a 16-bit instruction, the number ranges from 10 to 32,767. For a 32-bit instruction, the number ranges from 10 to 2,147,483,647; when the set pulse number is less than 110, the pulse output is not normal.

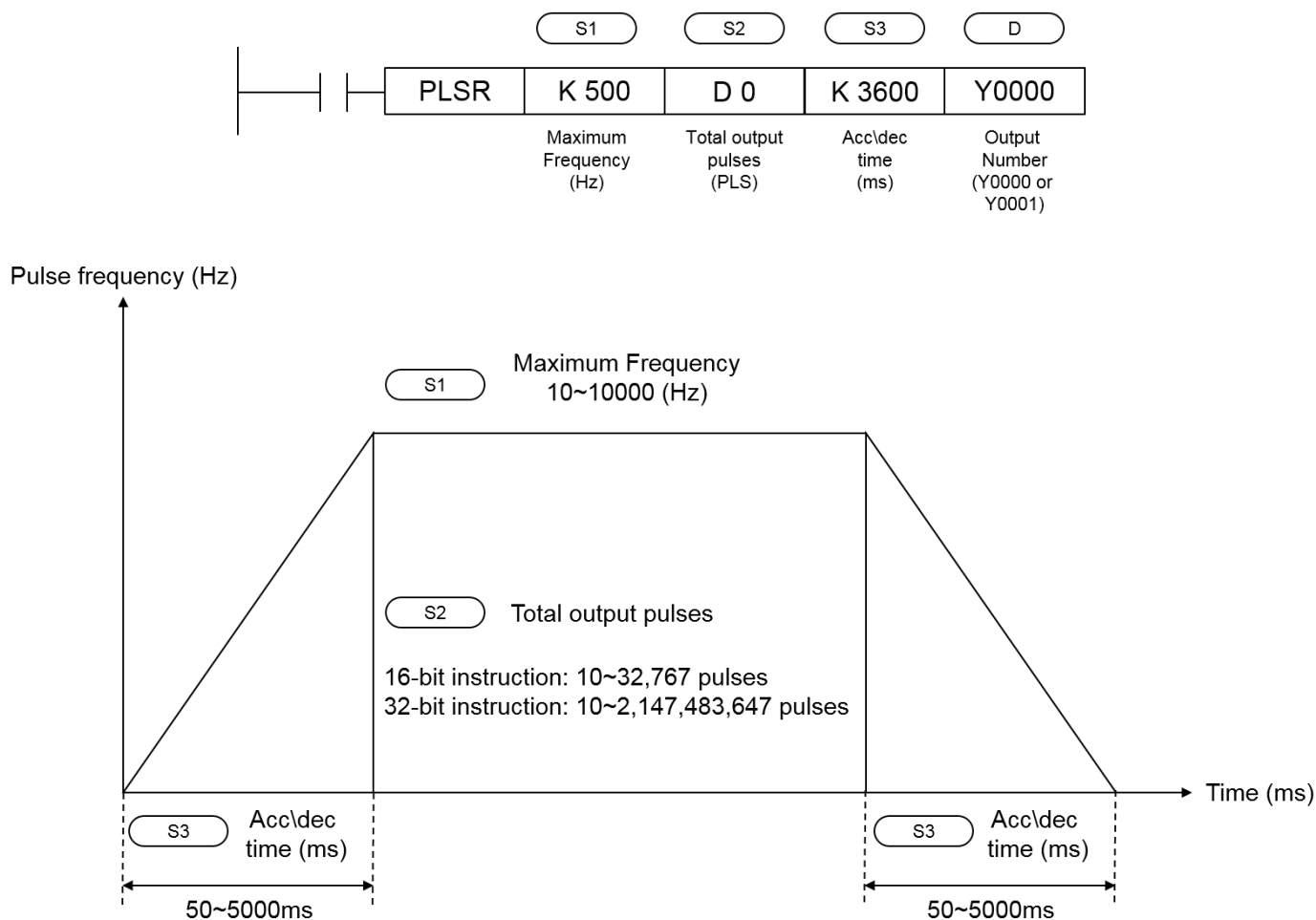
S3 indicates the acceleration/deceleration time ranging from 50 to 5000ms. Note that the deceleration time is the same as the acceleration time by default. (In the H1U\H2U series PLCs, the deceleration time can be set separately. See the following description.)

D indicates the pulse output port which can be Y0, Y1, Y2. Do not duplicate the output port of the PLSY instruction.

#### Instructions description:

- This instruction is executed in interrupt mode and is not affected by the scan cycle;
- When the command power flow is OFF, it will decelerate and stop; when the power flow turns from OFF to ON, the pulse output processing will restart;
  - During the pulse output, changing the operand has no effect on this output. The modified content will take effect the next time the instruction is executed. After the instruction is executed, the M8029 flag is set to ON;
  - The output port number of the PWM command cannot be repeated;
  - When the PLSR instruction is restarted, the last pulse output operation must be completed (M8147=0 at the end of Y0; M8148=0 at the end of Y1; M8149=0 at the end of Y2), and then a scan cycle is delayed before restarting this instruction (in the new version of H1U/H2U series PLC, this restriction can be exempted by setting, please refer to the description in Appendix 8.6 for details);

Example:



The special registers corresponding to each output port are listed as follow:

Register		Definition	Remarks
D8140	Lower byte	Number of total pulses output to Y0 port set in the PLSY or PLSR instruction	Applicable instructions: use DMOV K0 D81xx to perform clear operation
D8141	Upper byte		
D8142	Lower byte	Number of total pulses output to Y1 port set in the PLSY or PLSR instruction	
D8143	Upper byte		
D8150	Lower byte	Number of total pulses output to Y2 port set in the PLSY or PLSR instruction (3624MT/2416MT doesn't feature this port)	
D8151	Upper byte		
D8136	Lower byte	Accumulative value of the number of the pulses already output to Y0 and Y1	
D8137	Upper byte		

The output frequency range of this instruction is 10~100,000 Hz. When the maximum speed or high-speed conversion of acceleration/deceleration exceeds this range, it will be automatically converted (increased or decreased) to a value within the range and then executed. However, the lowest value of output frequency that can be output depends on the following formula:

$$\sqrt{\text{Max. Frequency}} = \frac{\text{S1. Hz}}{(2 \times (\text{Acc/dec time S3. Ms} \div 1000))} = \text{Minimum number of output frequency}$$

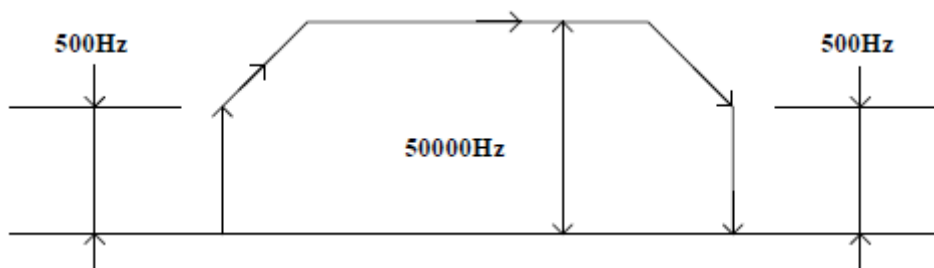
The frequency of the initial stage of acceleration and the final stage of deceleration shall not be lower than the calculation result of the above formula.

[Example]

Maximum speed: 50000Hz, acceleration and deceleration time: 100ms

$$\sqrt{50000 \div (2 \times (100 \div 1000))} = 500\text{Hz}$$

When the max frequency S1 is 50000Hz. The actual output frequency at the beginning of acceleration and the end of deceleration is 500 Hz.



Notice:

In the new version of H1U/H2U series PLC, the functions of PLSR, DRVI, DRVA and other instructions are enhanced;

- 1) By using special bits M8135~M8137 (respectively corresponding to Y0~Y2) to be ON, the number of output pulses can be changed during operation (large or small); at the same time, the acceleration time is determined by S3 the deceleration time is defined by the following registers:

Port number	Used special bit	The register used for changing deceleration time
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167

- 2) The special bits M8085 to M8087 (corresponding to Y0 to Y2) can be turned to ON to implement the following function: When the drive special bit is ON, the next pulse output instruction can be executed immediately, and the invalid flow of previous instruction does not need to be processed.
- 3) The special bits M8090 to M8092 (corresponding to Y0 to Y2) can be turned to ON to implement the following function: An interrupt is performed after the pulse output is completed.

The specific interrupts are listed in the table below.

Port number	Used special bit	Corresponding user interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

For details, please refer to the description in Appendix 8.6.

## 4.3.7 Convenient instructions (60~69)

16-bit	32-bit	P	SER	Data search
✓	✓	✓		
9steps	17 steps		Instruction format: SER S1 S2 D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1							✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓
n					✓	✓							✓		
Range: 16-bit instruction: n = 1~256; 32-bit instruction: n = 1~128															

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, where;

S1 is the head address of the searched data stack.

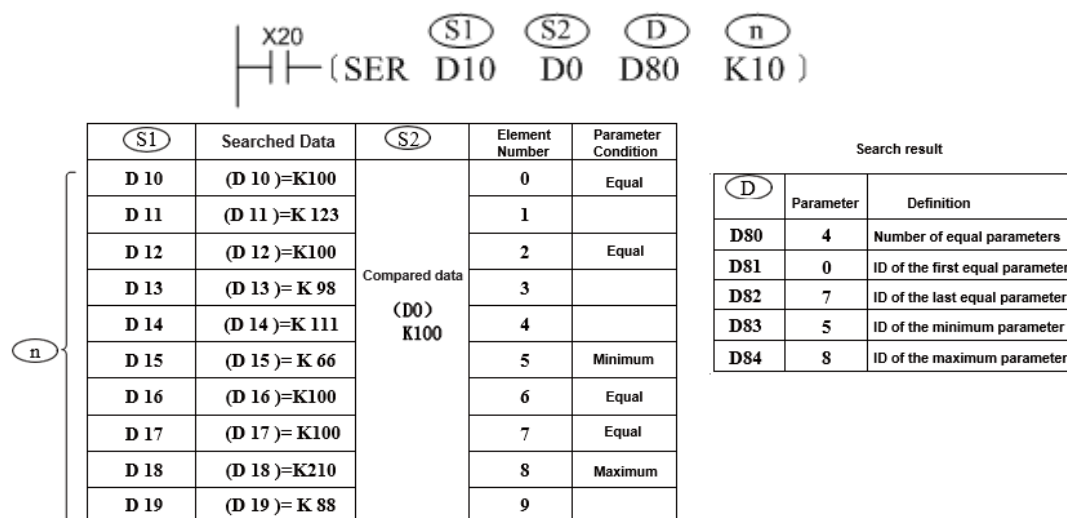
S2 is the data to be searched for.

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.

Example:



### Instruction description

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.

16-bit	32-bit	P	ABSD	Absolute cam control mode
✓	✓			
9 steps	17 steps		Instruction format: ABSD S1 S2 D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1							✓	✓	✓	✓	✓	✓	✓		
S2												✓			
D		✓	✓	✓											
n	Constant, n=1~64;														

When the operand of S1 is KnX, KnY, KnM, KnS, if it is a 16-bit instruction then K4 must be specified; if it is a 32-bit instruction then K8 must be specified and the component numbers of X, Y, M, and S must be multiples of 8. S2 operand can only specify C0~C199 for 16-bit instruction and for 32bit instruction only C200~C254 can be specified.

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, where:

S1 is the starting component address of the comparison table;

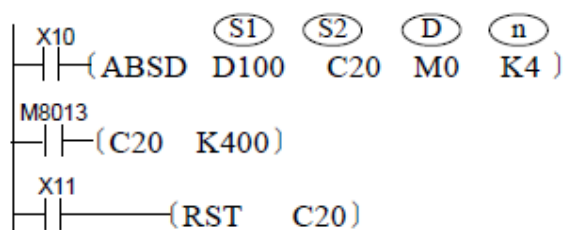
S2 is the serial number of the counter component. When using 32-bit instructions, it can be a 32-bit counter;

D is the starting address of the comparison result storage area, occupying n consecutive address bit variable units;

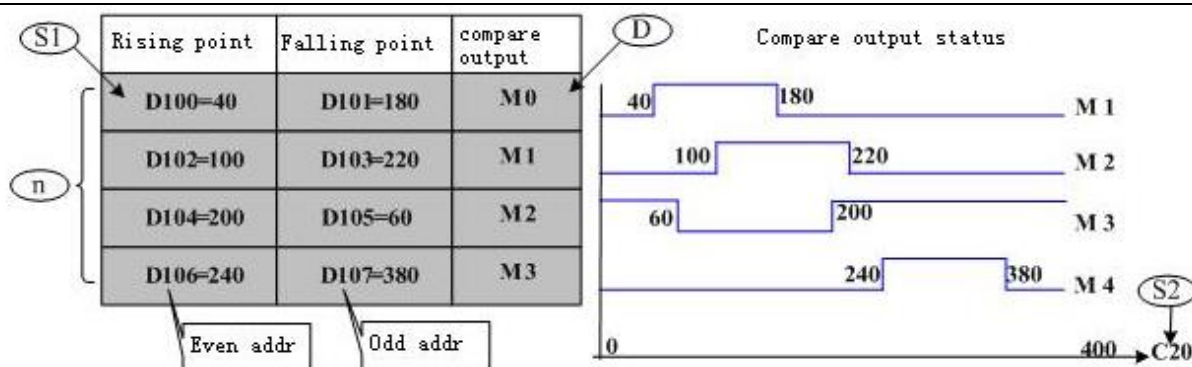
n is the number of groups of multiple comparison data.

In 32-bit operation, S1, S2, and D point to 32-bit variables and n is calculated based on 32-bit variable width.

Example 1:



Assume that related variables are assigned values as follows. When X10 = ON, the execution result is shown in the following figure.



### Instruction description:

- 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.

16-bit	32-bit	P	INCD	Incremental cam control mode
✓				
9 steps			Instruction format: INCD S1 S2 D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1							✓	✓	✓	✓	✓	✓	✓		
S2												✓			
D		✓	✓	✓											
n	Constant, n=1~64;														
When the S1 operand is KnX, KnY, KnM, KnS, if it is a 16-bit instruction then K4 must be specified; S2 operand specifies C0~C233 in 16-bit instruction.															

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, where:

S1 is the starting component address of the comparison table;

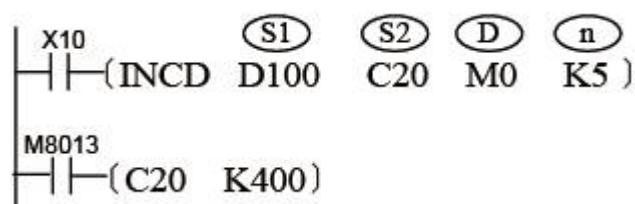
S2 is the counter component number, and its adjacent S2 + 1 unit is used to count the number of counter resets after a comparison match.

D is the start address of the comparison result storage area, occupying n consecutive address bit variable units;

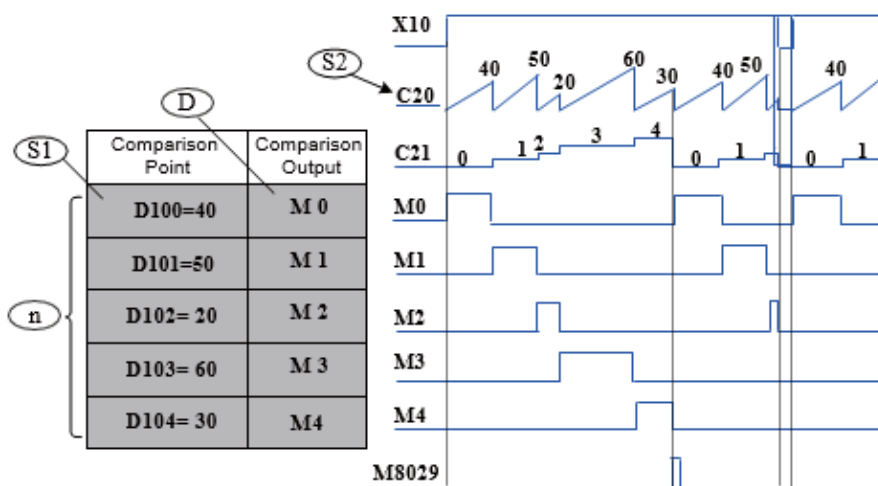
n is the group number of multiple comparison data.

The complete flag M8029 is set to ON after comparison of n data entries is completed.

Example 1:



Assume that related variables are assigned values as follows. When X10 = ON, the execution result is shown in the following figure.



#### Instruction description:

- 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.



16-bit	32-bit	P	TTMR	Teaching timer
✓				
5 steps			Instruction format: TTMR D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D													✓		
n					✓	✓									

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 100ms.

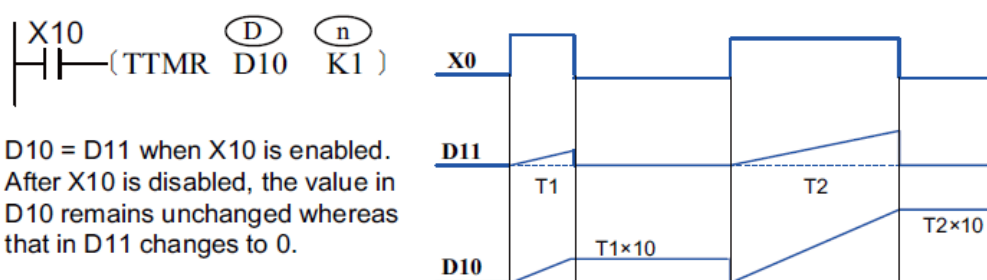
n is the multiple. The actual multiple is calculated by  $10n$  (n in the range 0 to 2).

When n = K0, the actual multiple is x1.

When n = K1, the actual multiple is x10.

When n = K2, the actual multiple is x100.

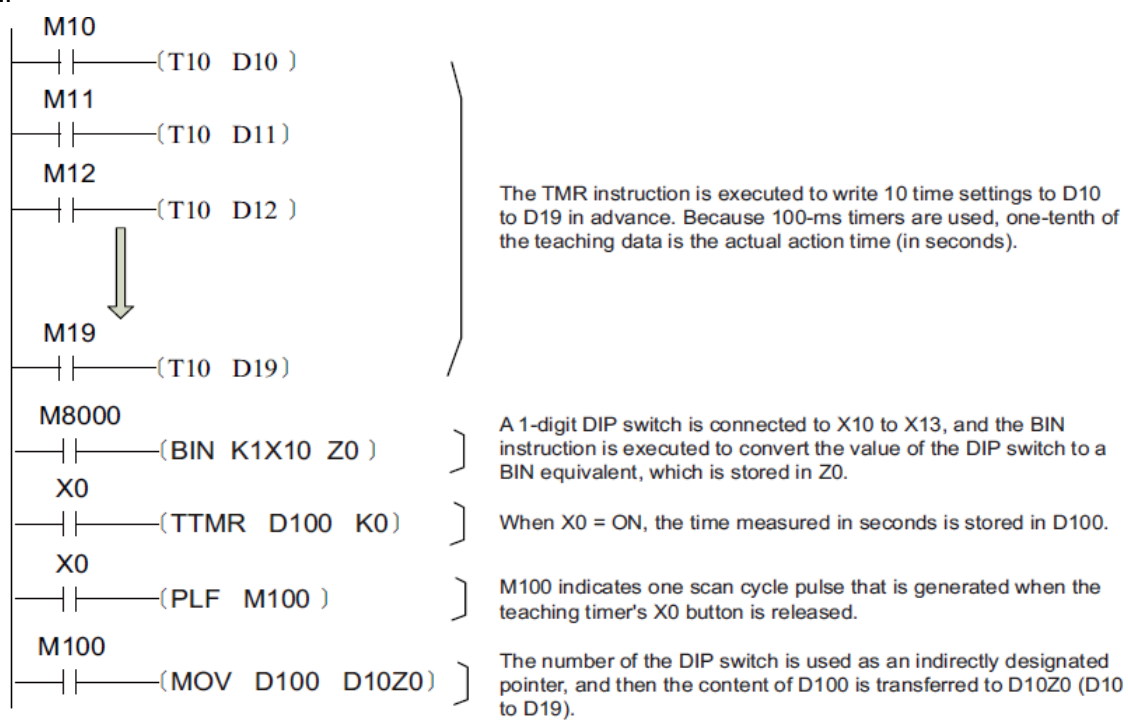
Example 1:



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: second)	$1 \times T$	$D11 = D10 \times 10$
K1 (Unit: 100 ms)	$10 \times T$	$D11 = D10$
K2 (Unit: 10 ms)	$100 \times T$	$D11 = D10 / 10$

## Example 2:



16-bit	32-bit	P	STMR	Special timer
✓				
7 steps			Instruction format: STMR S m D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											✓				
m	Constant, m = 1~32,767														
D		✓	✓	✓											

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.

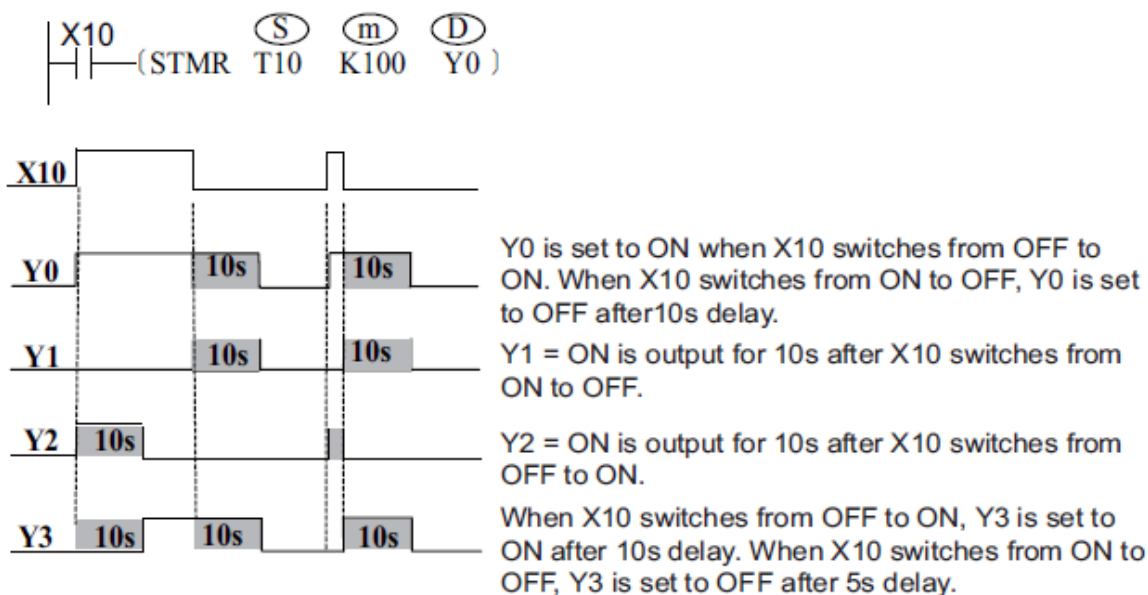
m is the preset delay value, in the unit of 100 ms. The value range is K1 to K32,767.

D is the start number of four consecutive elements that output the delay actions.

Note:

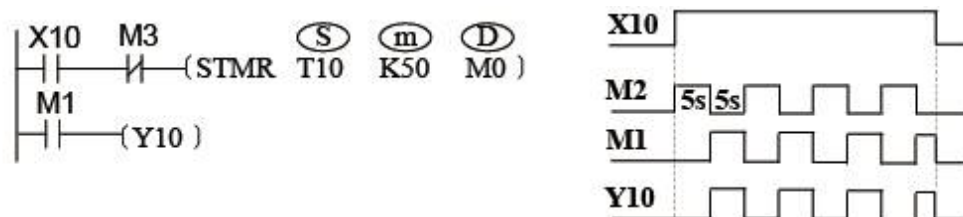
The timer used by this instruction and the corresponding outputs cannot be reused by other instructions.

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.)

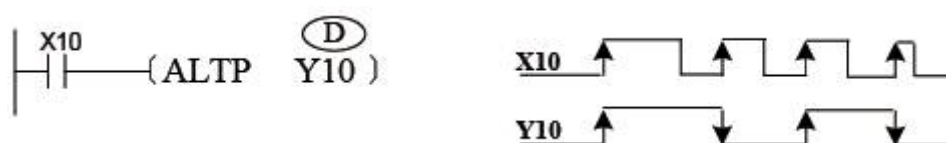


16-bit	32-bit	P	ALT	Alternate output
✓		✓		
3 steps		3 steps	Instruction format: ALT D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D		✓	✓	✓											

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:

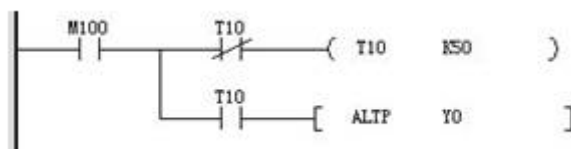


The action triggered by the following instruction is the same as that triggered by the ALTP instruction.



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.



16-bit	32-bit	P	RAMP	Ramp instruction
✓				
9 steps			Instruction format: RAMP S1 S2 D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													✓		
S2													✓		
D													✓		
n	Constant, 1~32,767														

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.

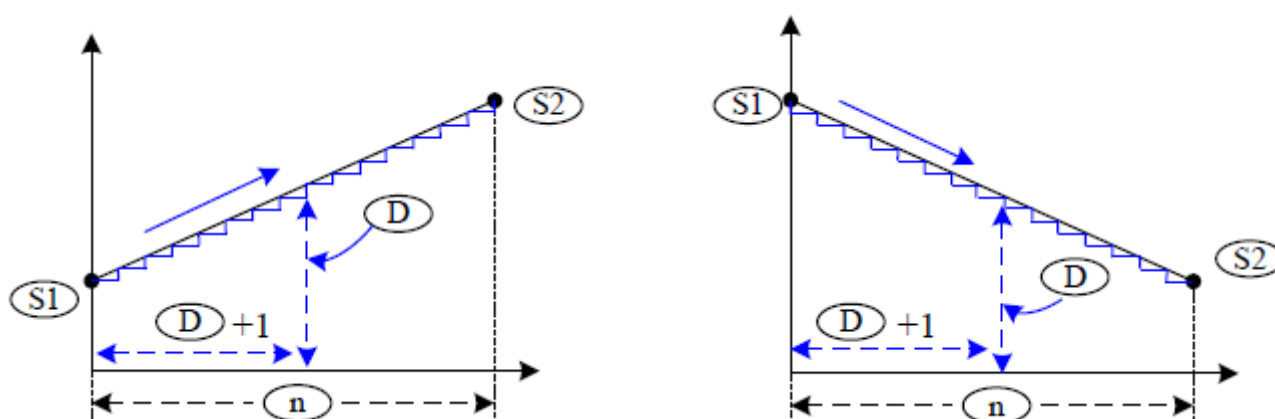
S2 is the end value unit of a ramp signal.

D stores the process values of a linear interpolation signal. The interpolation times counter is stored in D+1.

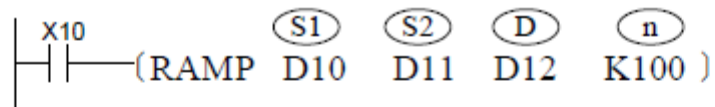
n is the number of scan execution times required to complete interpolation. The value range is 1 to 32,767.

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.)

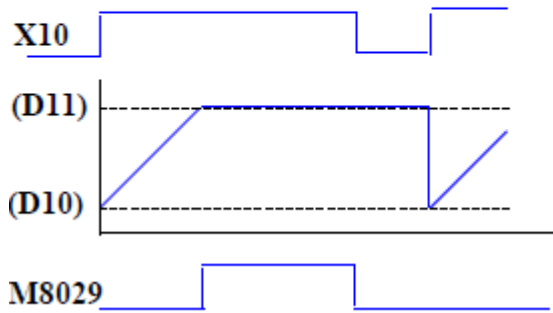
Interpolation adopts integer calculation, and the fractional part is discarded. The function of the instruction is shown as follows:



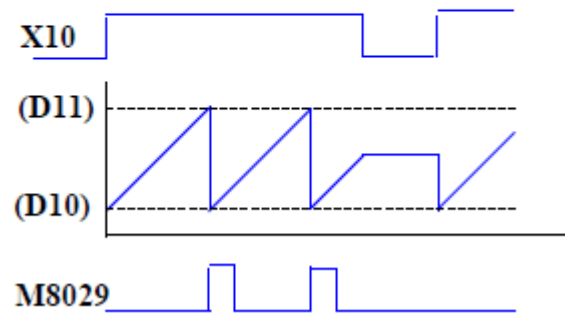
The RAMP instruction has two modes, which mode to use is selected by the M8026 flag. M8029 is set to ON when interpolation is completed. The execution is shown as follows:



M8026=ON



M8026=OFF



16-bit	32-bit	P	ROTC	Rotary table control
✓				
9 steps			Instruction format: ROTC S m1 m2 D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S													✓		
D		✓	✓	✓											
m1	2~32,767, m1 ≥ m2														
m2	0~32,767, m1 ≥ m2														

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, where:

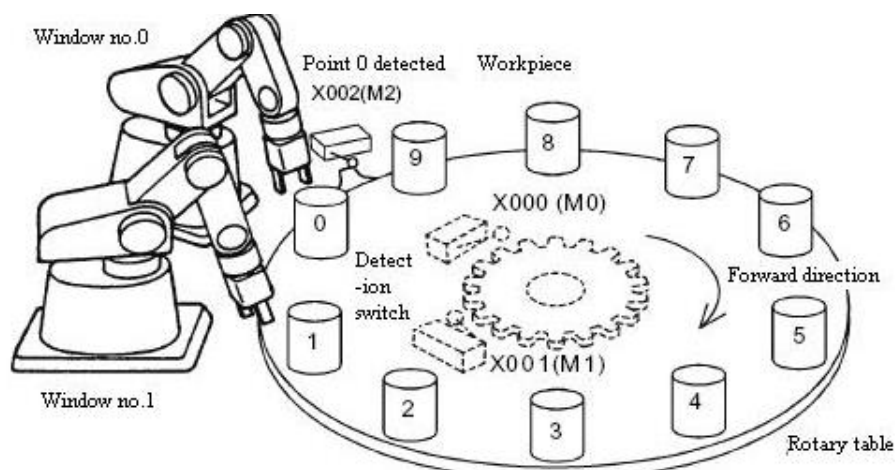
S starting unit of the counting variable;

m1 number of stations on the rotating worktable must be  $m1 \geq m2$ ;

m2 number of low-speed stations on the rotating worktable must be  $m1 \geq m2$ ;

D is the starting unit for storing the position detection signal of the rotary table, which occupies the following 8 bit variable units.

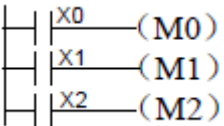
The following figure shows the signal configuration method. X0 and X1 are respectively connected to the A-phase and B-phase output signals of the A/B quadrature encoder. The quadrature phase signal can be acquired using a mechanical switch. X2 is connected to the detection input of workstation 0 (the ON state is enabled when the rotary table rotates to this workstation). The three signals are used to detect the current rotational speed, direction, and workstation of the rotary table...



Application example:



The table below describes the variables used in this instruction.

Variable	Function	Operation descriptions
D200	Used as a counting register	The three units need to be set in a user program in advance.
D201	Window no. calling setting	
D202	Workpiece no. calling setting	
M0	A phase signal	Executed before a user program scans this statement each time 
M1	B phase signal	
M2	Zero-point detection signal	
M3	High-speed forward rotation	When X10 is ON, the results of M3 to M7 can be automatically obtained. When X10 is OFF, the M3 to M7 are OFF.
M4	Low-speed forward rotation	
M5	Stop	
M6	Low-speed reverse rotation	
M7	High-speed reverse rotation	

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.



16-bit	32-bit	P	SORT	Data sorting
✓				
11 steps			Instruction format: SORT S m1 m2 D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S													✓		
m1	Constant, 1~32														
m2	Constant, 1~6														
D													✓		
n					✓	✓							✓		

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).

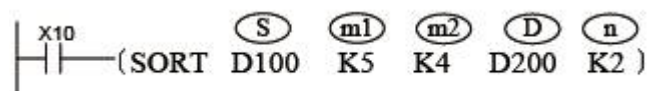
m2 is the number of array columns, or the number of columns in every record.

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.

Example:



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:

Row Number	1	2	3	4
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:

Row Number	1	2	3	4
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:

Row Number	1	2	3	4
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

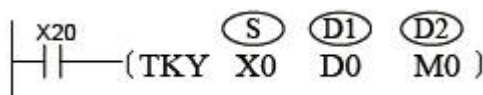
### 4.3.8 Peripheral I/O (70~79)

16-bit	32-bit	P	TKY	Ten key input
✓	✓			
7 steps	13 steps		Instruction format: TKY S D1 D2	

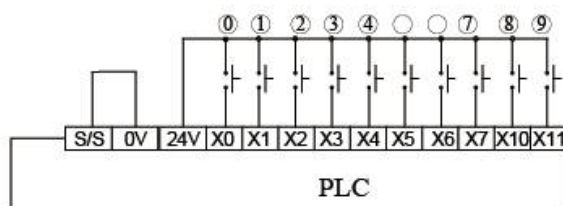
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓	✓	✓	✓											
D1								✓	✓	✓	✓	✓	✓	✓	✓
D2		✓	✓	✓											

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.

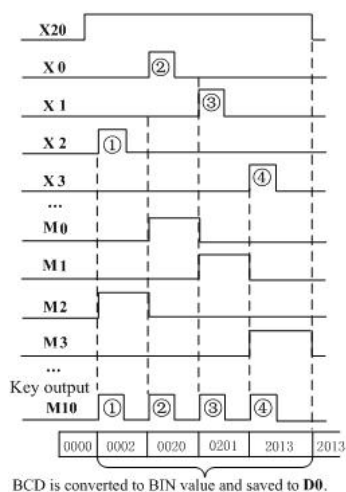
Example:



The following figure shows the hardware wiring diagram:



To input a number "2013", press keys ②①③ (X2, X0, X1, X3) in sequence. The following figure shows the actions of the PLC's internal variables.



Set by parameters in an instruction, X0~X11 respectively correspond to numeric keys 0~9; M0~M9 correspond to the status of keys; Key output unit will be reset whenever a key is pressed.

Key values (e.g. 2013) are converted to BIN and saved to the designated (D1) unit D; (D0=0x7DD), D0 will never change even when the power flow turns OFF.

When several keys are pressed simultaneously, the key which is firstly detected is valid; if the number entered is more than 4 digits, the first entered number will overflow and only a 4-digit number is left.

If the 32-bit instruction (DTKY) is used, D1 stores the 32-bit variable that is, D1 and D0 in preceding example, which correspond to the higher and lower byte respectively.

16-bit	32-bit	P	HKY	Hexadecimal key input
✓	✓			
9 steps	17 steps		Instruction format: HKY S D1 D2 D3	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓														
D1		✓													
D2										✓	✓	✓	✓	✓	✓
D3		✓	✓	✓											

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 PLC's of the transistor output type, where:

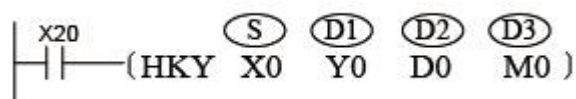
S is the number of the starting port of the scanning input port X of the keys, 4 X ports starting with which will be used;

D1 is the starting port button of scanning output Y port, and it uses the four succeeding Y ports.

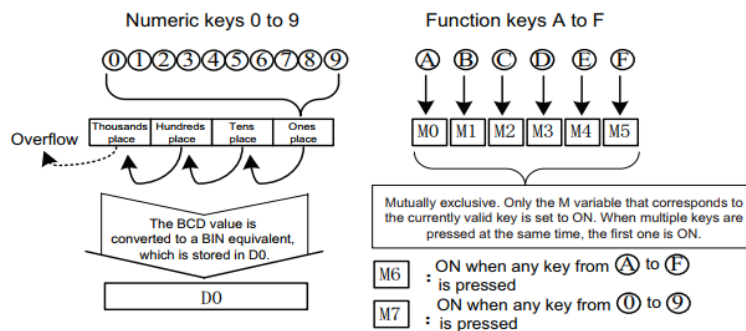
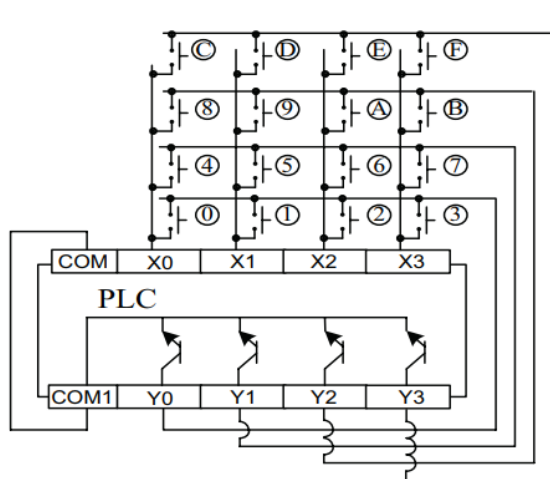
D2 is the storage cell of input number which ranges from 0~9999; 8 digits decimal number ranges from 0~99, 999, 999 can be entered when 32-bit instruction is used;

D3 is the address of the starting unit of the entering status of the keys, which occupies a variable unit of 8 continuous bits.

Example:



The wiring diagram and parameter response description as below:



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 keys ① to ⑨ in hexadecimal format in D2.

16-bit	32-bit	P	DSW	Digital switch
✓				
9 steps			Instruction format: DSW S D1 D2 n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓														
D1		✓													
D2											✓	✓	✓	✓	✓
n	Constant, n = 1~2														

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, where:

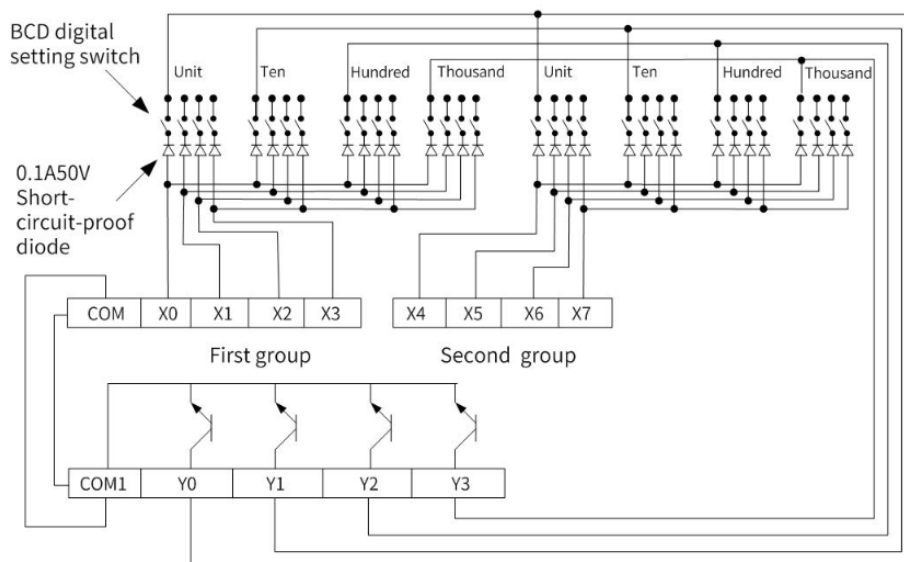
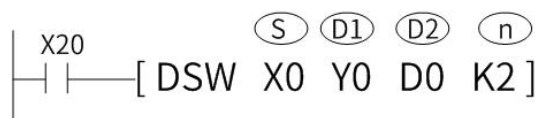
S is the start terminal number of key scanning X terminal. If n = 1 then the following 4X terminals will be used; If n = 2 then the following 8X terminals will be used;

D1 is the starting port button of scanning output Y port, and it uses the four succeeding Y ports;

D2 is the storage cell of key input value, 0~9999;

n is the number of switch set. Only 1~2 can be selected.

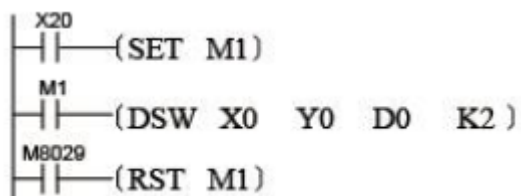
Example:



When X20=ON, digital switch setting is scanned and read.  
 1. Values of the first group of digital switches are converted to binary values, which are stored in D0.  
 2. Values of the second group of digital switches are converted to binary values, which are stored in D1.  
 3. After reading is complete once, M8029 is set to ON.

### Instruction description

The digital switch can be properly detected only for PLCs using output ports of the transistor type. It takes several scanning periods to complete the read operation of one digital switch. If the read operation is triggered by keys, it is recommended to use the following program statement to ensure the integrity of the scanning periods.



16-bit	32-bit	P	SEGD	7-segment decoding
✓		✓		
5 steps		5 steps	Instruction format: SEGD S D	

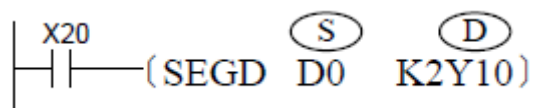
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓	✓	✓

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.

S is the data source to be decoded (take the lowest four bits b0~b3 of the BIN content);

D is the variable that stores the 7-segment code after decoding.

Example 1:



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		7-segment LED Display Combination	Internal Decoding Table Value								Decoded Character		
HEX	BIN		B7	B6	B5	B4	B3	B2	B1	B0			
0	0000	<p>Each bit corresponding to one segment 1 = Segment ON 0 = Segment OFF</p>		0	1	1	1	1	1	1	0	0	
1	0001		0	0	0	0	0	1	1	0			
2	0010		0	1	0	1	1	0	1	1			
3	0011		0	1	0	0	1	1	1	1			
4	0100		0	1	1	0	0	1	1	0			
5	0101		0	1	1	0	1	1	0	1			
6	0110		0	1	1	1	1	1	0	1			
7	0111		0	0	1	0	0	1	1	1			
8	1000		0	1	1	1	1	1	1	1			
9	1001		0	1	1	0	1	1	1	1			
A	1010		0	1	1	1	0	1	1	1			
B	1011		0	1	1	1	1	1	0	0			
C	1100		0	0	1	1	1	0	0	1			
D	1101		0	1	0	1	1	1	1	0			
E	1110		0	1	1	1	1	0	0	1			
F	1111		0	1	1	1	0	0	0	1			



16-bit	32-bit	P	SEGL	7-segment with latch
✓				
7 steps			Instruction format: SEGL S D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D		✓													
n	Constant, n = 0~7														

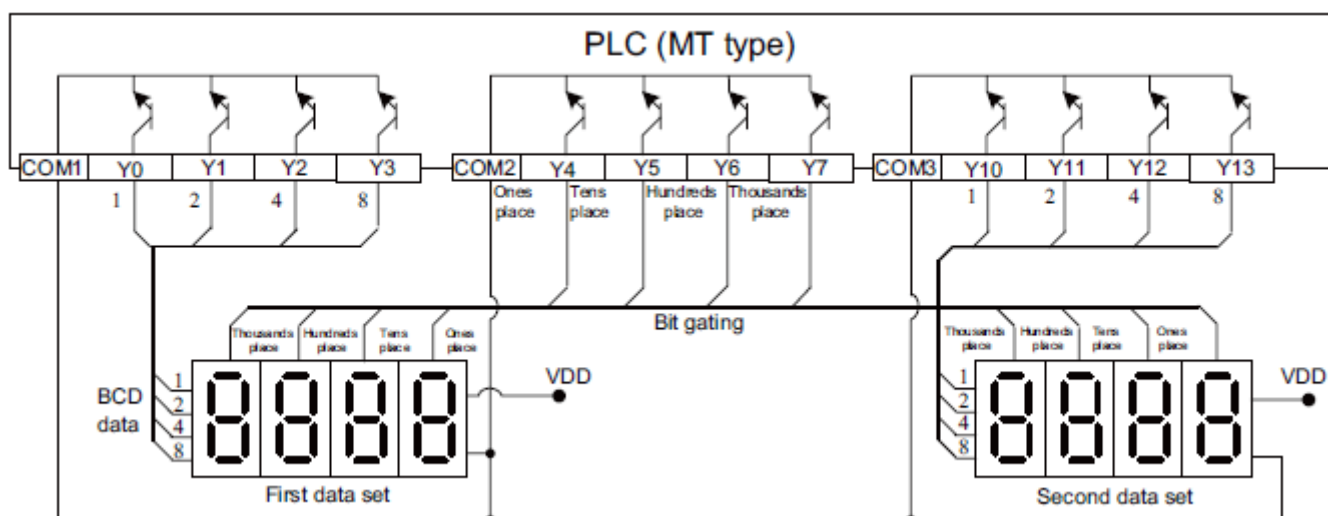
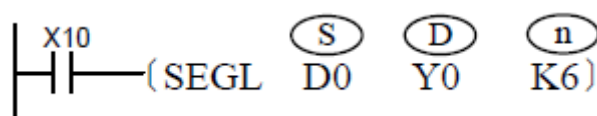
Eight or 12 Y ports are used to drive 4- or 8-digit latched 7-segment LED displays. The display mode is scan-driven, where:

S is the data to be displayed, and its value is sent to the LED for display after BCD conversion;

D is the starting address number of the Y port used by the display driver;

n is the starting address number of the Y port used by the display driver;

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.

Group number	1 <sup>st</sup> group				2 <sup>nd</sup> group			
Output polarity of Y data	PNP		NPN		PNP		NPN	
Strobe and data polarity	Same	Oposite	same	Oposite	same	Oposite	same	Oposite
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.

#### Instruction description:

Because relays are not suitable for high-frequency scan output, this instruction is only applicable to PLCs of the transistor output type.

16-bit	32-bit	P	ARWS	Arrow switch
✓				
9 steps			Instruction format: ARWS S D1 D2 n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓	✓	✓	✓											
D1											✓	✓	✓	✓	✓
D2		✓													
n	Constant, n = 1~3														

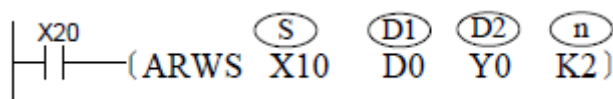
Parameters stored in registers can be edited easily by using X ports to provide edit keys and Y ports to drive 4-digit 7-segment LED displays, where:

S is the start address of key-press input which occupies the following 4bit cells

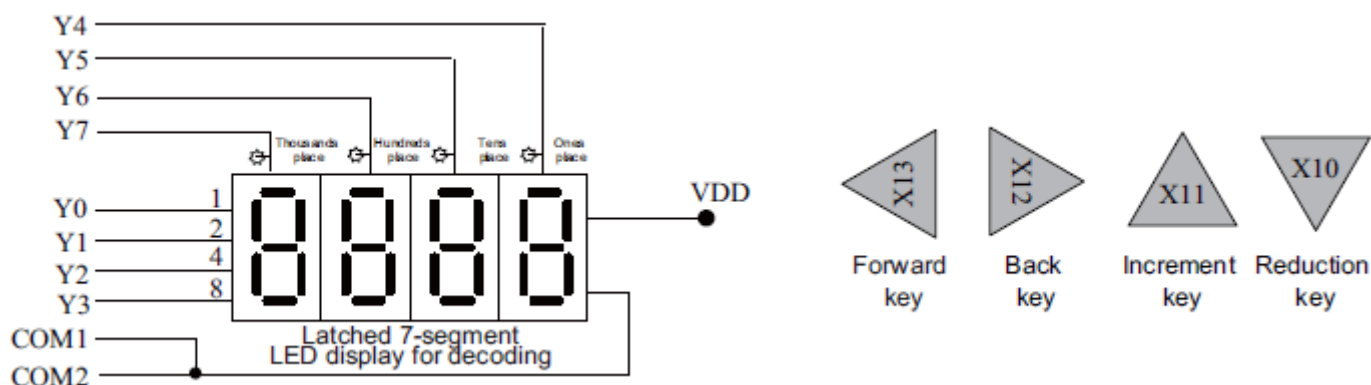
D1 is used for displayed and modified variables, and it only can display a variable of 16-bit width

D2 is used as the start address of Y terminal for driving the LED display, and it occupies the following 8 Y terminals n is the setting value of signal logic, please refer to above detailed description of n in SEGL instruction.

Example:



The following figure shows the hardware wiring diagram. The PLC is of the transistor output type.



Operation method:

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.

**Instruction description:**

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.

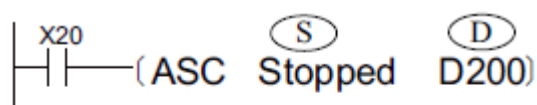
16-bit	32-bit	P	ASC	ASCII code conversion
✓				
11 steps			Instruction format: ASC S D	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	When inputting instruction, it is inputted by a constant 8 characters in length.														
D												✓	✓	✓	

S is the English character which will be converted to ASCII code, it is entered from computer, and the permissible maximum length is 8 characters

D stores the starting cell number of ASCII code and it occupies the following 4 (M8161=0) or 8 (M8161=1) variable cells.

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 digit	ASCII code (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

16-bit	32-bit	P	PR	ASCII code printing
✓				
5 steps			Instruction format: PR S D	

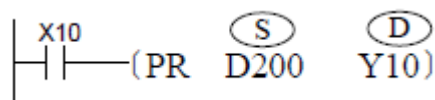
Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											✓	✓	✓		
D		✓													

The values stored in designated variable units are output synchronously byte by byte through Y output ports, where:

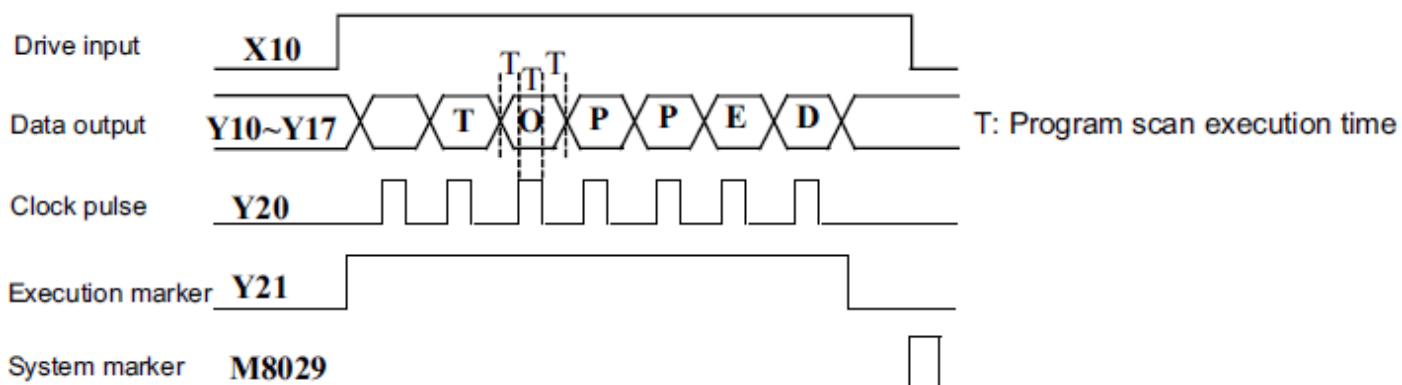
S is the starting address of the variable units to be output;

D is the starting number of the Y port for output print.

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.

### Instruction description

- 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 act 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.



16-bit	32-bit	P	FROM	Read BFM area
✓	✓	✓		
9 steps	17 steps		Instruction format: FROM m1 m2 D n	

Operands	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)								✓	✓	✓	✓	✓	✓	✓	✓
m1 = 0~7 (local), 100+station number (remote); m2 = 0~32767; n = 1~32767; If D is specified then K1~K4 can be used for 16-bit instruction and K1~K8 can be used for 32-bit instruction; m1, m2, n don't support character device D register															

This instruction is used to read the data read operation of the BFM area register of the special expansion module. where:

m1 is the address number of the expansion module:

In the local expansion module, the value range is 0~7, the nearest to the main module is 0, numbered sequentially, and a maximum of 8 local expansion modules are allowed. It is suitable for local analog modules, temperature modules, etc., but cannot be used for local digital expansion modules.

In the remote expansion module, the value range is module communication station number + 100, and a maximum of 62 remote expansion modules are allowed. Applicable to all remote expansion modules, including remote digital expansion modules, pay attention to the difference between the use of local digital modules and remote digital modules;

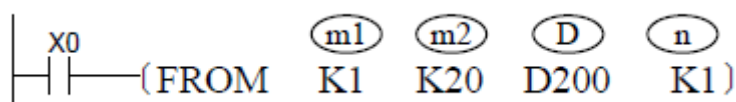
m2 is the register address number of the BFM in the special module, with a value range of 0~32767

D is the storage address of the read parameter in the main module. When the number of read registers is more than one, the subsequent unit is occupied

n is the number of parameters read this time (calculated in word), the value range is 1 to 32767, which are read sequentially according to the register address.

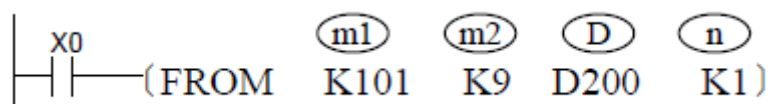
Please note that H1U only can use remote extension module.

Example 1:



Access to the local extension module means that when X0 is ON, the content of the 20<sup>th</sup> address (16-bit width) in the #1 special module is read out to the D200 register of the PLC, one reading at a time (n=1). When X0 is OFF, no operation is performed.

Example 2:



To access the remote expansion module, assuming that the H2U-4ADR is used.

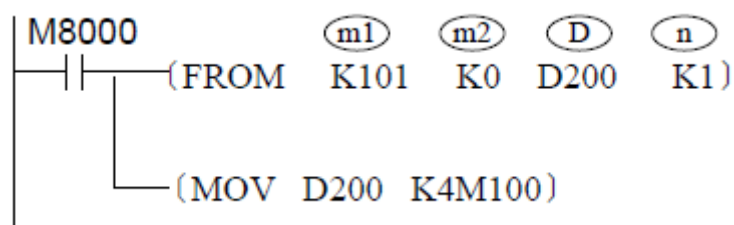
K101 stands for (100 + station number 1); when the remote module uses FROM/TO instructions, add the station number offset 100.

K9 represents the BFM9 of the 4AD module.

D200 represents the storage address of the read BFM9.

K1 means to read a 16-bit BFM.

Example 3:



This example introduces the access method of the remote digital expansion module H2U-1600ENDR, because the 16-point input module uses a BFM area to represent all input points, that is, each bit of K0 (a register) in the BFM area is each input point, so you can read the status of the module and use MOV D200 K4M100 to take the position respectively. At this time, M100~M107 are X0~X7 of the remote module, and M108~M115 are X10~X17 of the remote module. We can read M100~M115 to read the status of X0~X17.

Note: In order to ensure that the latest status of the input points of the digital expansion module can be read, please use M8000 to drive the FROM instruction, which can be updated every scan cycle.

When using 32-bit instructions, the address designated by D is the low 16-bit address, and D+1 is the high 16-bit address.

16-bit	32-bit	P	TO	Write to BFM area
✓	✓	✓		
9 steps	17 steps		Instruction format: TO m1 m2 D n	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
D								✓	✓	✓	✓	✓	✓	✓	✓	

m1 = 0~7 (local), 100+station number (remote);  
m2 = 0~32767;  
n=1~32767;  
If D is specified then K1~K4 can be used for 16-bit instruction and K1~K8 can be used for 32-bit instruction;  
m1, m2, n does not support character device D register

This instruction is used to write data to the BFM area register of the special expansion module. where:  
m1 is the address number of the special expansion module:

In the local expansion module, the value range is 0~7, the nearest to the main module is 0, numbered sequentially, and a maximum of 8 local expansion modules are allowed. It is suitable for local analog modules, temperature modules, etc., but cannot be used for local digital expansion modules.

In the remote expansion module, the value range is module communication station number + 100, and a maximum of 62 remote expansion modules are allowed. Applicable to all remote expansion modules, including remote digital expansion modules, pay attention to the difference between the use of local digital modules and remote digital modules;

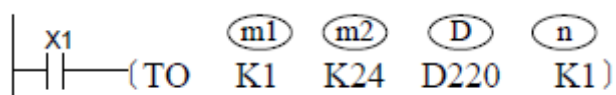
m2 is the register address number of the BFM in the special module, with a value range of 0~32767;

D is the address of the parameter register in the main module, and its parameter is used as the source of write operation data. When the number of registers for the write operation is more than one, the subsequent unit is occupied;

n is the number of write parameters this time, the value range is 1~32767, which are written sequentially according to the register address.

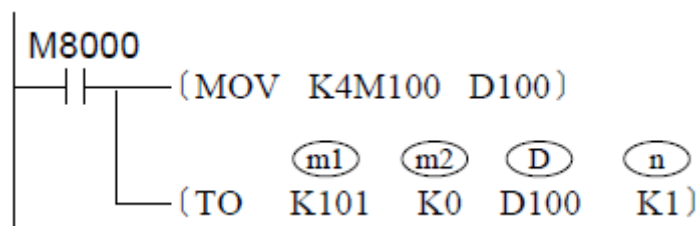
Note that H1U can only use remote expansion modules.

Example 1:



It means that when X1 is ON, write the data in the PLC's D220 register to the 24<sup>th</sup> address in the #1 special module, and only write one entry at a time. When X1 is OFF, no operation is performed.

## Example 2:

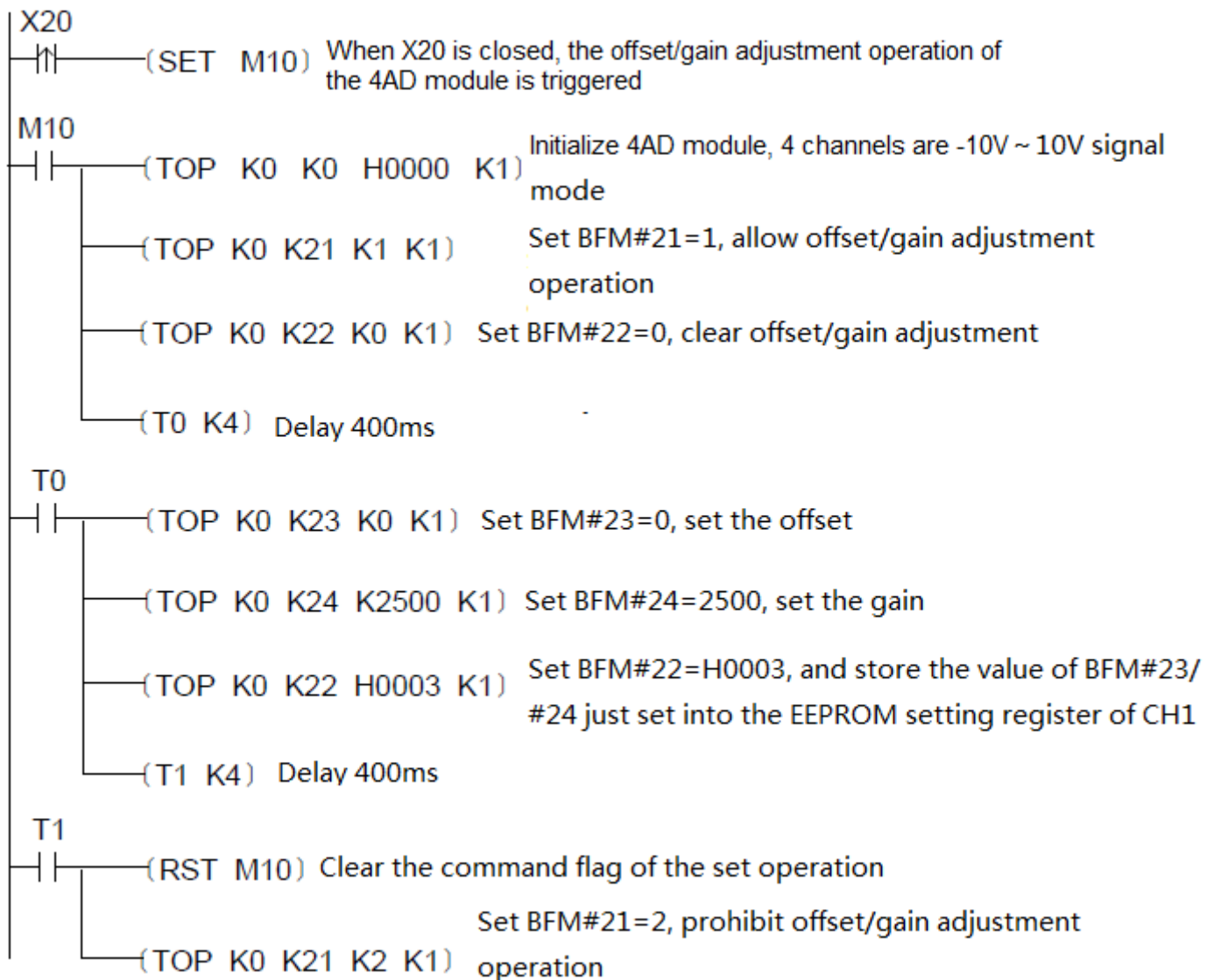


This example introduces the access method of the remote digital expansion module H2U-0016ERDR, because the 16-point input module uses a BFM area to represent all output points, that is, each bit of K0 (a register) in the BFM area is each output. Therefore, you can put the status of all output points in one word and then transfer it to the module.

In the example, M100~M107 are Y0~Y7 of the remote module, and M108~M115 are Y10~Y17 of the remote module. Changing M100~M115 can change the status of Y0~Y17 in the remote module. In the remote digital output module, the TO command can be driven after each output point is updated, or it can be driven by M8000.

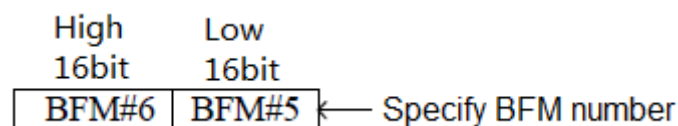
## Example 3:

To set the offset/gain of the CH1 channel of a H2U-4AD module with address #0, the programming is as follows:



When using 32-bit instructions, the address designated by D is the lower 16-bit address, and D+1 is the higher 16-bit address.

For example:



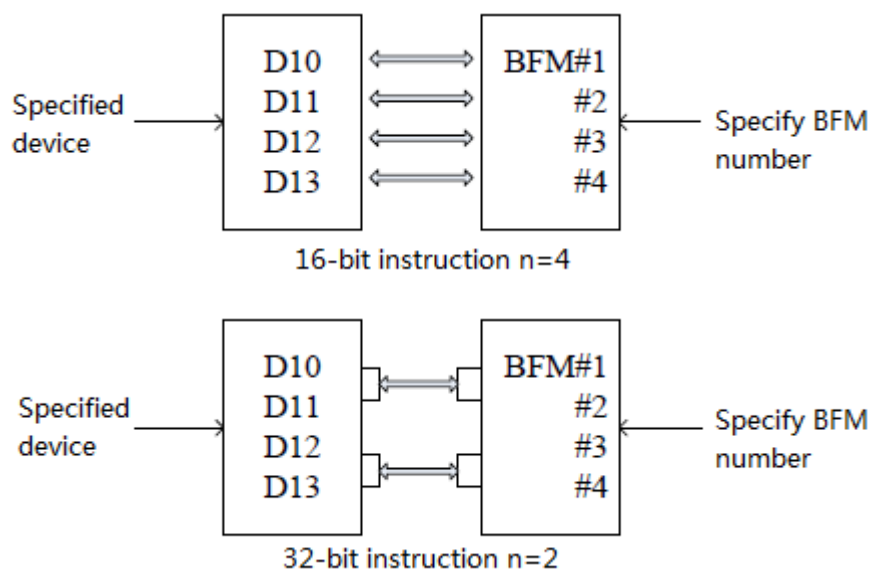
n is the number of operating data

n = 2 means 2 word (16-bit instruction)

n=1 means 2 Word (32-bit instruction)

So please pay attention to that n = 2 (16-bit instruction) and n = 1(32-bit instruction) have the same meanings.

For example:



Instructions to use FROM/TO instructions:

Description of special registers related to FROM/TO instruction:

1) M8164 (FROM/TO command variable transmission point mode)

If M8164=ON, when the FROM/TO instruction is executed, the content of the special data register D8164 (transfer point number designation registers of the FROM/TO instruction) is used as the number of transfer points n to process

- 2) It is a time-consuming operation to use FROM/TO instructions to access the expansion module. When multiple FROM/TO instructions are executed or multiple buffer memory data are transmitted, the PLC scan cycle will be prolonged. In order to prevent running timeout, you can add WDT instruction to extend the monitoring timer time before FROM/TO or differ the execution time of FROM/TO instruction or use pulse execution instruction.
- 3) Please refer to the dedicated manual attached to each special module for the connection method and input/output number of the special module.
- 4) For the configuration of the remote module, please refer to "5.9 CAN Communication Command Description".

### 4.3.9 Peripheral SER equipment (80~93)

16-bit	32-bit	P	RS	Serial data transfer
✓				
9 steps			Instruction format: RS S m D n	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
m					✓	✓							✓			
D													✓			
n					✓	✓							✓			

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 allows 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, where:

S is the initial address of the register area where the data to be sent will be stored;

m is the length of the data to be sent (bytes), ranges (0~256);

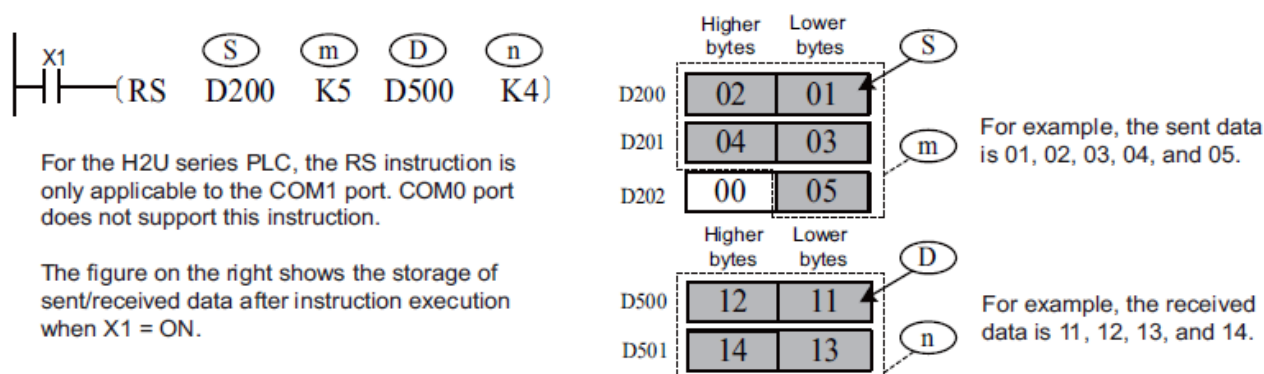
D is the initial address of the storage register which receives communication data;

n is the length of the communication data received (bytes), ranges (0~256).

Whether the RS instruction works in half-duplex or full-duplex mode is determined by bit 10 of D8120.

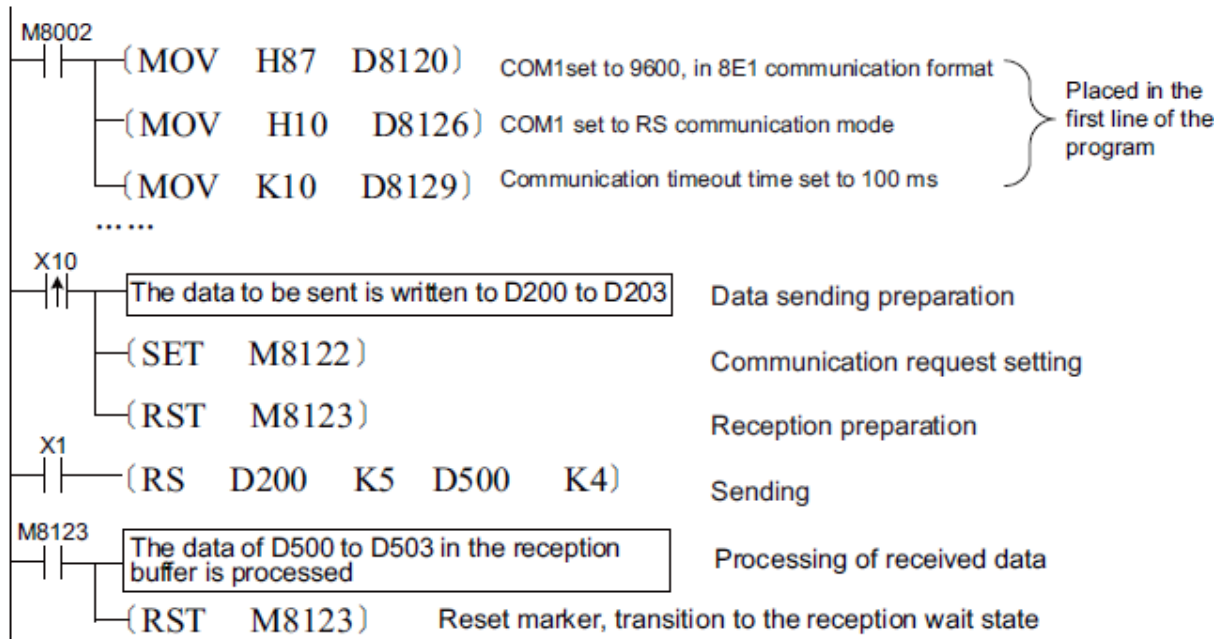
Multiple RS instructions can be compiled by the user program, but only one instruction can be driven at a time. M8122 must be set before the RS instruction is driven.

Example 1:



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:





The hardware format and protocol selection of the communication port involve many special registers and flags, which are explained as follows:

### Serial communication format setting:

Item	Setting bit	Definition	COM0 port description	COM1 port description
			D8110	D8120
Data bit	Bit0	0b-7Bits 1b-8Bits	MODBUS-RTU slave station protocol and instruction only support 8bit data bit, or error will occur to the communication	
Parity bit	Bit2~Bit1	00b-none (N)		
		01b-odd (O)		
		11b- even(E)		
Stop bit	Bit3	0-1Bits		
		1-2Bits		
Baudrate	Bit7~Bit4	0011b-300bps		
		0100b-600bps		
		0101b-1200bps		
		0110b-2400bps		
		0111b-4800bps		
		1000b-9600bps		
		1001b-19200bps		
		1010b-38400bps		
		1011b-57600bps		
		1100b-115200bps		
Start character	Bit8	0b- none		
		1b-(D8124)		
Termination character	Bit9	0b- none		
		1b-(D8125)		
Half-duplex/full-duplex	Bit10	0b- full-duplex		Applicable to RS instruction

Note For details, please refer to "Chapter 5 Communication Functions".

16-bit	32-bit	P	PRUN	Octal bit transfer
✓	✓	✓		
5 steps	9 steps		Instruction format: PRUN S D	

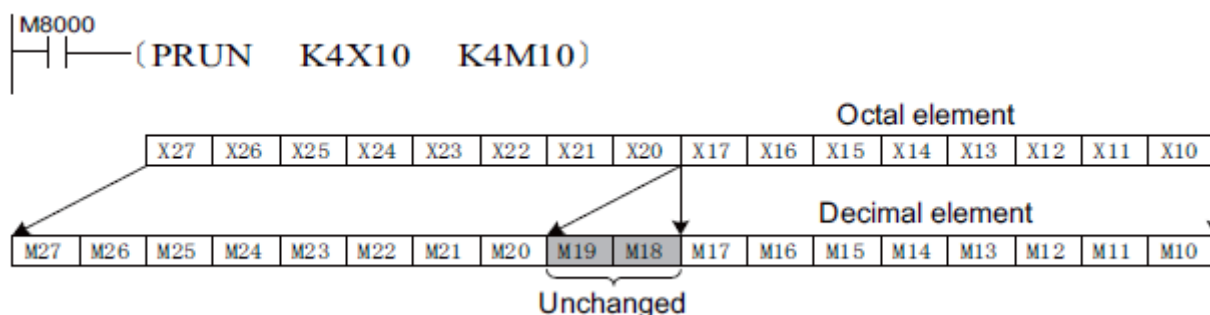
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S							✓		✓							
D								✓	✓							

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, where:

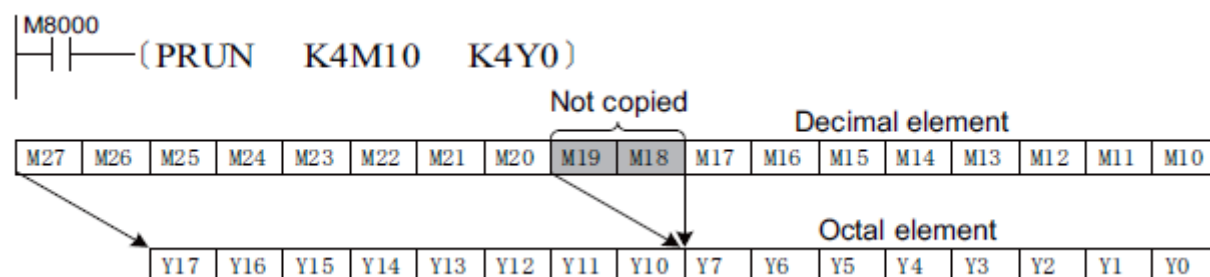
S is the starting address of the bit variables to be copied, where the unit digit of the addresses must be 0, such as X10, M20, etc.

D is the starting address of the target bit variables. Also, the unit digit of the addresses must be 0, such as M30, Y10, etc.

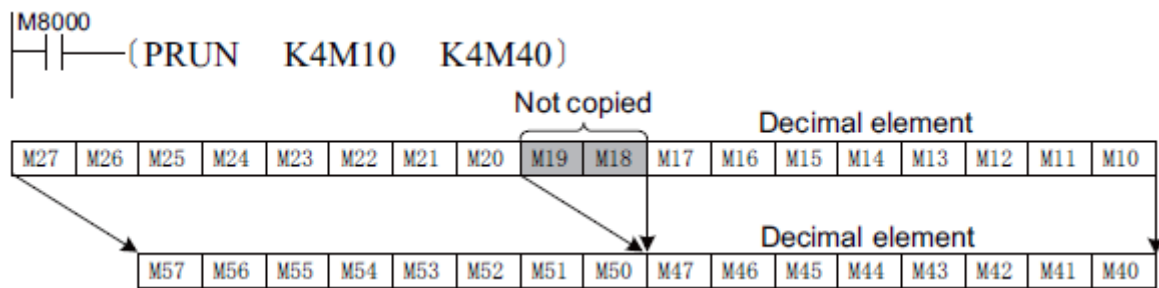
Example 1:



Example 2:



Example 3:



16-bit	32-bit	P	ASCII	ASCII code conversion
✓		✓		
7 steps		7 steps	Instruction format: ASCII S D n	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D								✓	✓	✓	✓	✓	✓		
n	Constant, n = 1~256														

This instruction is used to convert the value of S into an ASCII code and stores the conversion result in variables starting with the D address. In the instruction:

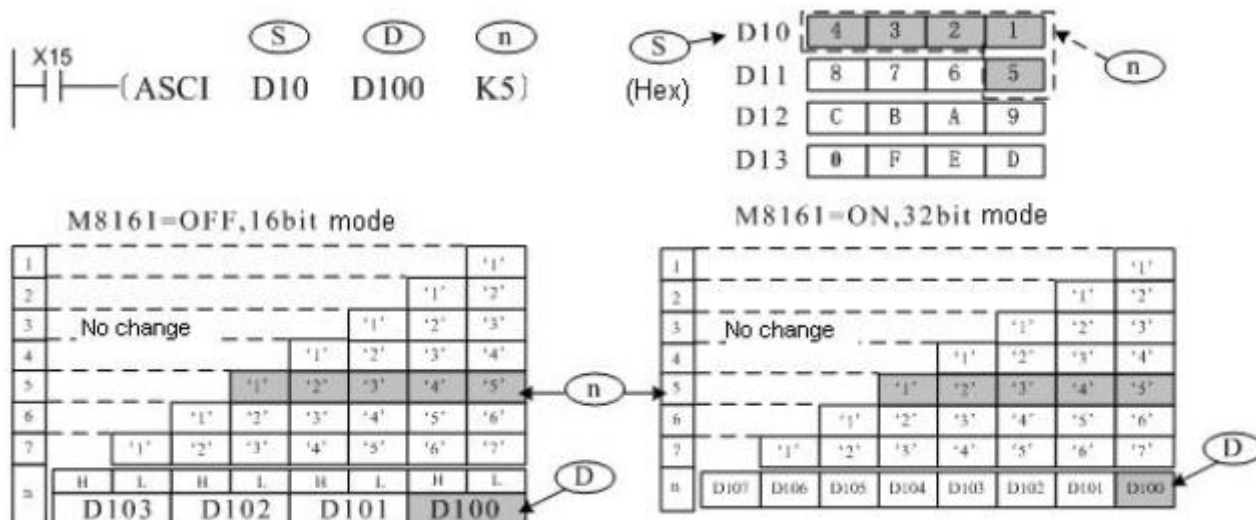
S indicates the variable addresses or constant values to be converted.

D is the start No. of addresses for storing ASCII codes after conversion.

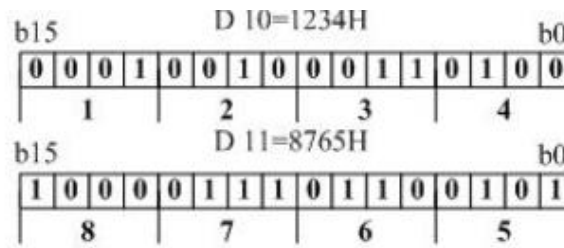
n indicates the number of character bits to be converted. The value ranges from 1 to 256.

The conversion executed by the ASCII instruction complies with the mapping table of ASCII codes and hexadecimal data. For example, ASCII "0" maps to HEX "H30" and ASCII "F" maps to HEX "H46". For the mapping between hexadecimal data and ASCII codes, see the appendix.

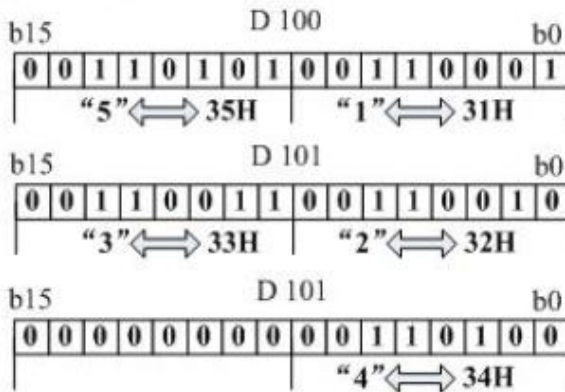
Example:



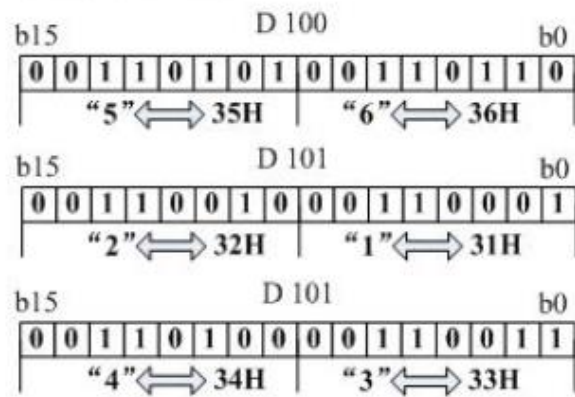
M8161 determines the width mode of the destination variables for storing conversion results. When M8161 is OFF, the 16-bit mode is adopted, that is, the higher bytes and lower bytes of variables are stored. When M8161 is ON, the 8-bit mode is adopted, that is, only the low bytes of variables are stored. Therefore, the length of the variable area that is used increases.



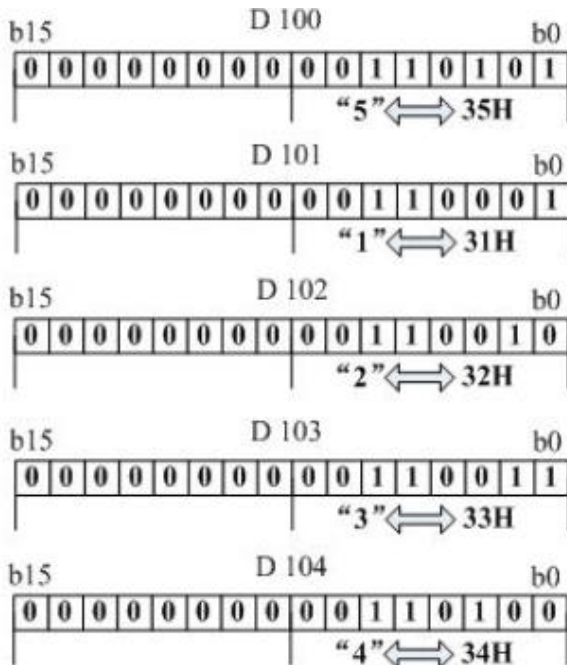
Bit component when M8161=OFF, n=5  
(D10-D11) conversion



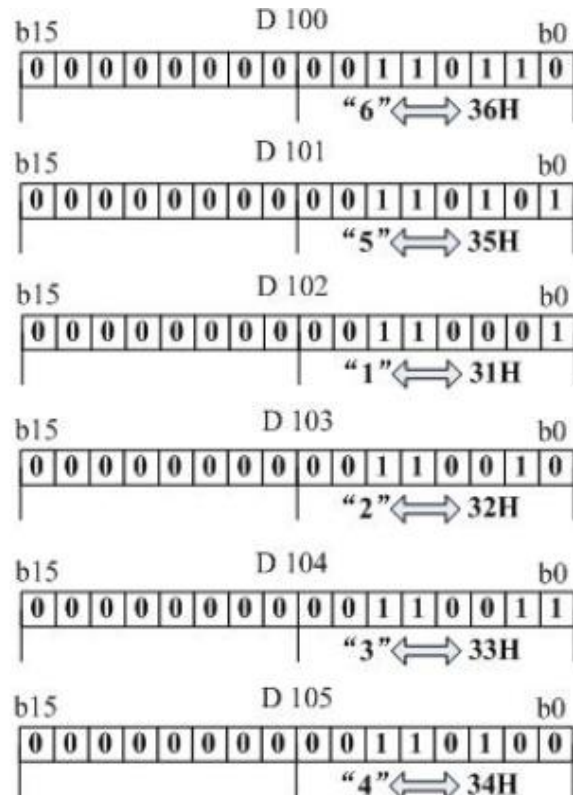
Bit component when M8161=OFF, n=6  
(D10-D11) conversion



Bit component when M8161=ON, n=5  
(D10-D11) conversion



Bit component when M8161=ON, n=6  
(D10-D11) conversion



Note: The RS, HEX, ASCII, and CCD instructions share the M8161 mode flag. Therefore, exercise caution during programming.

16-bit	32-bit	P	HEX	ASCII-HEX conversion
✓		✓		
7 steps		7 steps	Instruction format: HEX S D n	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D								✓	✓	✓	✓	✓	✓	✓	✓	
n	Constant, n = 1~256															

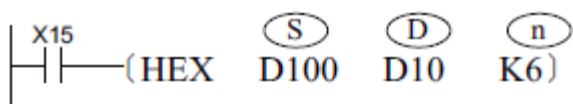
The values of variables from head address S are converted to hexadecimal equivalents. The result is stored in variables from head address D. The number of converted characters and storage mode are configurable.

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:



Data from head address D100:

M8161 = OFF, 16-bit mode

	H	L
D100	42H(B)	41H(A)
D101	44H(D)	43H(C)
D102	31H(1)	30H(0)
D103	33H(3)	32H(2)
D104	35H(5)	34H(4)
D105	37H(7)	36H(6)
D106	39H(9)	38H(8)
D107	42H(0)	41H(A)
D108	42H(C)	42H(B)

Diagram shows a dashed box around D100-D102 with an arrow from 'S' pointing to D100 and an arrow from 'n' pointing to the box.

M8161 = ON, 8-bit mode

	H	L
D100	42H(B)	41H(A)
D101	44H(D)	43H(C)
D102	31H(1)	30H(0)
D103	33H(3)	32H(2)
D104	35H(5)	34H(4)
D105	37H(7)	36H(6)
D106	39H(9)	38H(8)
D107	42H(0)	41H(A)
D108	42H(C)	42H(B)

Diagram shows a dashed box around D100-D102 with an arrow from 'n' pointing to the box.

1		...AH
2		..ABH
3	Unchanged	.ABCH
4		ABCDH
5		..AH BCD0H
6		..ABH CD01H
7		.ABCH D012H
8		ABCDH 0123H
9	...AH	BCD0H 1234H
n	D12	D11 D10

Diagram shows an arrow from 'n' pointing to the row containing D10 and an arrow from 'D' pointing to the row containing D10.

1		...AH
2		..ACH
3	Unchanged	.AC0H
4		AC02H
5		..AH C024H
6		..ACH 0246H
7		.AC0H 2468H
8		AC02H 468AH
9	...AH	C024H 68ABH
n	D12	D11 D10

Diagram shows an arrow from 'n' pointing to the row containing D10 and an arrow from 'D' pointing to the row containing D10.

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	0	1
C				D				0				1				

D 11

0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	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			

## Notes:

- The RS, HEX, ASCII, 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.

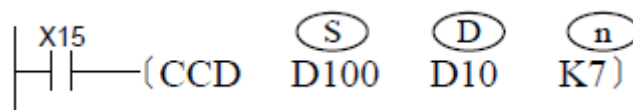


16-bit	32-bit	P	CCD	Check code
✓		✓		
7 steps		7 steps	Instruction format: CCD S D n	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S							✓	✓	✓	✓	✓	✓	✓			
D								✓	✓	✓	✓	✓	✓			
n	Constant, n = 1~256															

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. This instruction is used to perform the sum check to ensure the correctness of data transmission in the 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).

For example, the data  
from head address  
D100 is as follows:

M8161 = OFF, 16-bit mode

	H	L
D100	12H=00010010	01H=00000001
D101	44H=01000100	A3H=10100011
D102	21H=00100001	3FH=00111111
D103	33H=00110011	D2H=11010010
D104	65H=01100101	A1H=10100001
D105	37H=00110111	C6H=11000101
D106	A9H=10101001	02H=00000010

(S) → D100

(n)

M8161 = ON, 8-bit mode

	H	L
D100	12H=00010010	01H=00000001
D101	44H=01000100	A3H=10100011
D102	21H=00100001	3FH=00111111
D103	33H=00110011	D2H=11010010
D104	65H=01100101	A1H=10100001
D105	37H=00110111	C6H=11000101
D106	A9H=10101001	02H=00000010

(n)

Cumulative sum: D10 

22CH
------

  
XOR (polarity): D11 

01111000
----------

Cumulative sum: D10 

31EH
------

  
XOR (polarity): D11 

00101010
----------

The RS, HEX, ASCII, and CCD instructions share the M8161 flag. Pay attention to the flag processing during programming.

16-bit	32-bit	P	PID	PID calculation
✓				
9 steps			Instruction format: PID S1 S2 S3 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1													✓			
S2													✓			
S3													✓			
D													✓			

This instruction is to complete the PID operation and is used to control the parameters of the closed-loop control system. PID control has a wide range of applications in mechanical equipment, pneumatic equipment, constant pressure water supply and electronic equipment, etc. where:

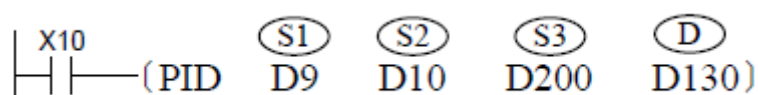
S1 is the target value of PID control;

S2 is the measured feedback value;

S3 is the starting address of the buffer area for setting parameters required for PID operation and saving intermediate results. It occupies a total of 25 variable units in the subsequent addresses. The value range is D0~D7975. It is best to specify the power failure retention area. Keep the set value, otherwise the buffer area needs to be assigned value before starting the operation for the first time. The function and parameter of each unit are described in this section;

D is the storage unit of the PID calculation results. Please specify D as the non-battery holding area, otherwise it needs to be initialized and cleared before starting the calculation for the first time.

Example:



The parameters are as follows:

What is stored in D9 is the target value of PID adjustment, and D10 is the closed-loop feedback value. Note that D9 and D10 must be the same dimension, such as both 0.01MPa units, or 1°C units, etc.; A total of 25 units of D200~D224 are used to store the set value and process value of PID operation. These values must be set item by item before the first PID calculation;

The D130 unit is used to store the calculated control output value, which is used to control the execution of the action.

The function and setting method of each unit parameter value started by S3 are described in the following table:

Unit	Function	Setting instructions
S3	Sampling time (TS)	The value ranges from 1ms to 32767ms but should be greater than the program scanning period of the PLC
S3+1	Action direction (ACT)	Bit 0: 0 = forward action; 1 = reverse action Bit 1: 0 = The input change alarm is invalid; 1 = The input change alarm is valid. Bit 2: 0 = The output change alarm is invalid; 1 = The output change alarm is valid. Bit 3: unavailable Bit 4: 0 = Auto-tuning does not act; 1 = Performing auto-tuning (this function is not supported in current versions) Bit 5: 0 = The upper and lower limits for output values are invalid; 1 = The upper and lower limits for output values are valid. Bit 6 to Bit 15: unavailable In addition, Bit 5 and Bit 2 cannot be set to ON simultaneously.
S3+2	Input filtering constant (a)	0 to 99[%], 0 = no input filtering
S3+3	Proportional gain (Kp)	1~32767[%]
S3+4	Integration time (T1)	0 to 32767 (x 100ms), 0 = processed as 8 (no integral)
S3+5	Differentiation gain (KD)	0 to 100[%], 0 = no differential gain
S3+6	Differentiation time (TD)	0 to 32767 (x 10ms), 0 = no differential processing
S3+(7~19)	Occupied by internal processing of PID operation. It should be cleared prior to first running	
When Bit 1 = 1, Bit 2 = 1, or Bit 5 = 1 in <ACT>, S3+(20 to 24) are occupied and they are defined as follows:		
S3+20	Input change (increase side) alarm setting value	0 to 32767, (valid when Bit 1 = 1 in <ACT>)
S3+21	Input change (decrease side) alarm setting value	0 to 32767, (valid when Bit 1 = 1 in <ACT>)
S3+22	Output change (increase side) alarm setting value	0 to 32767, (valid when Bit 2 = 1 and Bit 5 = 0 in <ACT>)
		Upper output limit: -32768 to 32767, (valid when Bit 1 = 0 and Bit 5 = 1 in <ACT>)
S3+23	Output change (decrease side) alarm setting value	0 to 32767, (valid when Bit 2 = 1 and Bit 5 = 0 in <ACT> in S3+1)
		Lower output limit: -32768 to 32767, (valid when Bit 1 = 0 and Bit 5 = 1 in <ACT>)
S3+24	Alarm output	Bit 0 input change (increase side) overflow Bit 1 input change (decrease side) overflow Bit 2 output change (increase side) overflow Bit 3 output change (decrease side) overflow (valid when Bit 1 = 1 or Bit 2 = 1 in <ACT>)

The theoretical calculation formula of PID is as follows:

EV<sub>n</sub>: Offset during current sample

D<sub>n</sub>: Current differential coefficient

EV<sub>n-1</sub>: Offset 1 cycle before

D<sub>n-1</sub>: Differential coefficient, 1 cycle before

SV: Target value

K<sub>p</sub>: Proportional gain

T<sub>s</sub>: Sample period

T<sub>i</sub>: Integral constant

Reverse logic

$$\Delta MV = K_p \left| (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n \right|$$

$$EV_n = SV - PV_{nf}$$

$$D_n = \frac{T_D}{T_s + \alpha_D * T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{\alpha_D * T_D}{T_s + \alpha_D * T_D} * D_{n-1}$$

$$MV_n = \sum \Delta MV_{n\Delta}$$

PV<sub>nf</sub>: Measured value during current sample (filtered)

PV<sub>nf-2</sub>: Measured value, 2 cycle before (filtered)

ΔT<sub>D</sub> : Differential constant

ΔMV: Output variance

Δα<sub>D</sub> : Differential gain

MV<sub>n</sub> : current controlled variable

L : Filter coefficient

Reverse logic

$$\Delta MV = K_p \left| (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n \right|$$

$$EV_n = SV - PV_{nf}$$

$$D_n = \frac{T_D}{T_s + \alpha_D * T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{\alpha_D * T_D}{T_s + \alpha_D * T_D} * D_{n-1}$$

$$MV_n = \sum \Delta MV_{n\Delta}$$

PV<sub>nf-1</sub>: measured value, 1 cycle before (filtered)

PV<sub>nf</sub> is calculated from the measured value using following formula:

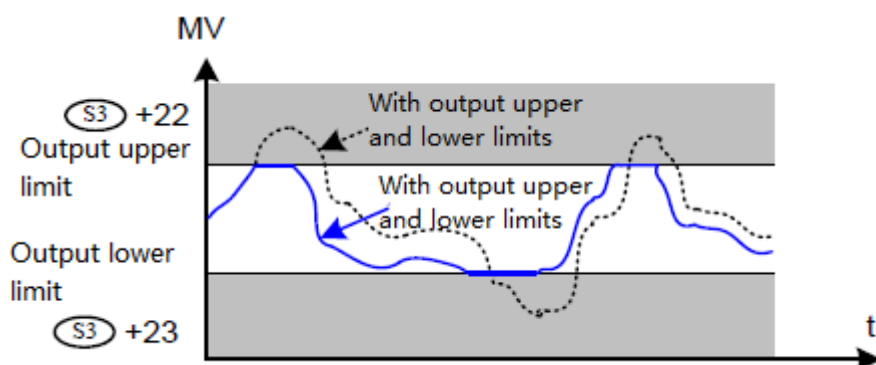
$$[\text{measured value after filter } PV_{nf}] = PV_n + L (PV_{nf-1} - PV_n)$$

PV<sub>n</sub> : measured value during current sample

Positive logic is also called the positive direction, such as the heating power adjustment of the thermostatic control system, which belongs to the positive logic PID control; the negative logic is also called the reverse direction, such as the cooling fan speed control of the thermostatic control system, which belongs to the negative logic PID control.

Output upper and lower limit setting:

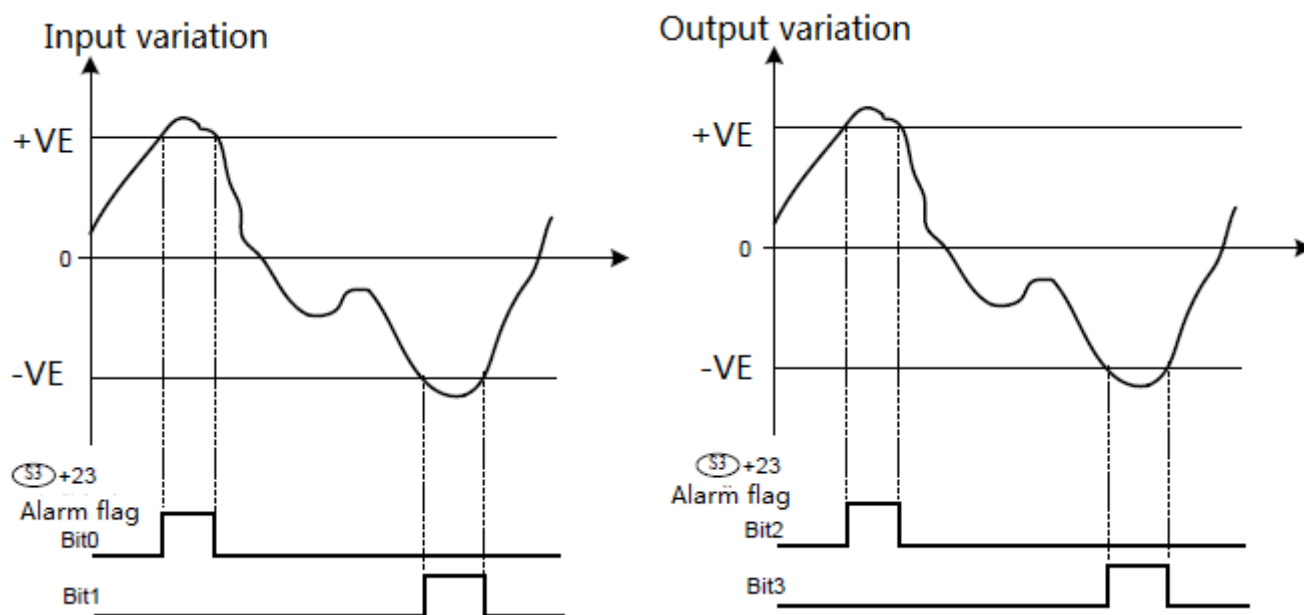
For most application systems, PID adjustment output range is limited, such as level signal amplitude range, frequency converter adjustment frequency range and so on. Set the output upper limit and output lower limit in PID operation according to the actual situation, so that the actuator of the closed-loop system can work under normal operating parameters.



### Over-limit alarm setting:

If you need to check whether the change of the controller's input (feedback) exceeds the limit (upper limit or lower limit), and whether the change of PID calculation output is too large (over the upper limit or lower limit), you can use the ACT (S3 + 1) unit, Set bit 1 = ON, bit 2 = ON of ACT to enable the alarm function; set the variation alarm limit in S3 + (20~23) unit, and read the parameter over-limit status in S3+24 unit during operation Up. This simplifies the calculation in some occasions where the adjustment status needs to be judged. When using the alarm function of output variation, bit 5 of S3 +1 (ACT) must be set to OFF.

The "change amount" mentioned here = (last value) - (this value), the corresponding action diagram is as follows:



It can be seen that the lower alarm limit setting value of "-VE" is negative. In order to avoid excessive fluctuations in the input quantity, the input signal is generally processed by hardware filtering, and software filtering can also be performed in the user program.

### PID constant setting:

The selection of PID constants such as  $K_p$ ,  $K_i$ ,  $K_d$ ,  $T_s$  is very important to PID control characteristics. The qualitative effects of these three parameters on stability are:

- 1) If the proportional gain  $K_p$  is too large, there will be overshoot, or even unable to stabilize;  $K_p$  is too small, the adjustment process is slow; when the setting is appropriate, the system PID adjustment response speed effect is good;

- 2) If the integral time  $T_i$  is too large, the adjustment process will be slow; if  $T_i$  is too small, the output mutation will be large and difficult to stabilize; when the setting is appropriate, the stability will be fast and the deviation will be small;
- 3) If the differential gain  $K_d$  is too large, the system is prone to self-excited oscillation and cannot be stabilized; a small  $K_d$  has a small effect on stability; if it is set properly, it will help to quickly approach the set value. Because this parameter has a great influence on stability, differential adjustment is often not used in most applications, that is, its gain is set to 0;
- 4) The period  $T_s$  used corresponds to the time interval for PID calculation, and its set value should be greater than the scan execution time of the PLC program. In order to ensure the PID adjustment effect, the PLC program is best run in a fixed scan cycle mode, or run PID in timed interruption.

## Error code

When an error occurs in the setting value of the control parameter or the data in the PID operation, the operation error M8067 will be turned on, and the following data will be stored in D8067 according to the error content.

Code	Error description	Solution conditions	Solution methods
K6705	The operand of the application instruction is outside the scope of the target device	PID command operation stop	Please confirm the content of the control data
K6706	The operand of the application instruction is outside the scope of the target device		
K6730	The sampling time (TS) is outside the range of the target device ( $TS < 0$ )		
K6732	The input filter constant is outside the range of the object ( $\alpha$ ) ( $\alpha < 0$ or $100 \leq \alpha$ )		
K6733	The proportional gain (KP) is outside the target range ( $KP < 0$ )		
K6734	The integration time (TI) is outside the target range ( $TI < 0$ )		
K6735	Differential gain (KD) is outside the target range ( $KD < 0$ or $201 \leq KD$ )		
K6736	Derivative time (TD) outside the scope of the target ( $KD < 0$ )		
K6740	Sampling time ( $T_s$ ) $\leq$ operation cycle	PID command operation	

Code	Error description	Solution conditions	Solution methods
K6742	The measured value change exceeds ((PV<-32768 or 32767<(PV)	continues	
K6743	Deviation exceeds (EV<-32768 or 32767<EV)		
K6744	Integral calculating value exceeds the range of -32768~32767		
K6745	Due to differentiation gains out of range, differentiation value is out of the range		
K6746	The differential calculation value exceeds (except -32768 to 32767)		
K6747	PID calculation result exceeds a certain value (except -32768~32767)		
K6750	Poor auto-tuning results		
K6751	Inconsistent auto-tuning action direction	Auto-tuning continues	The operating direction predicted from the measured value at the start of auto-tuning does not match the actual operating direction at the time of auto-tuning output. After correcting the relationship between the target value, output value for auto-tuning, and measured value, perform auto-tuning again.
K7652	Poor auto-tuning results	End of auto-tuning	The measured value during auto-tuning cannot operate correctly due to the up and down changes. Please make the sampling time much longer than the output change period and increase the input filter constant. Please perform auto tuning again after setting changes.

#### Key point

The correct measured value must be read into the PID measured value (PV) before the PID calculation is executed. Especially when performing PID calculation on the input value of the analog input module, pay attention to its conversion time.

#### Programming instructions:

5) The PID instruction can be used multiple times in the program and can be executed simultaneously, but the S3 variable area used in each PID instruction should not overlap; it can also be used in step instructions, jump instructions, timing interrupts, and subroutines. In this case, when the PID instruction is executed, the S3 +7



---

cache unit must be cleared in advance.

6) The maximum error of the sampling time TS is  $-(1 \text{ operation cycle} + 1 \text{ ms}) \sim + (1 \text{ operation cycle})$ . If the sampling time  $TS \leq 1 \text{ operation cycle}$  of the programmable controller, the following PID operation error (K6740) will occur, and the PID operation will be executed with  $TS = \text{operation cycle}$ . In this case, it is recommended to use constant scan mode or use PID instruction in timer interrupt (I6xx~I8xx).

16-bit	32-bit	P	MODBUS	MODBUS communication
✓				
9 steps			Instruction format: MODBUS S1 S2 n D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
n					✓	✓							✓			
D													✓			

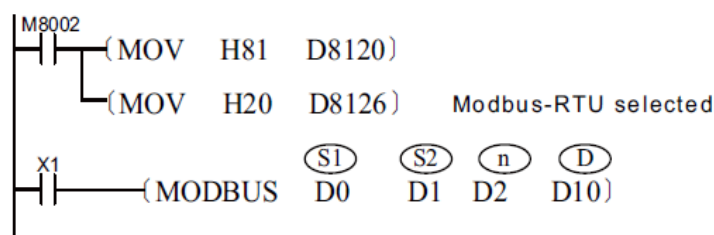
For more intuitive and convenient operation, the RS instruction can be replaced with the MODBUS instruction during communication through a standard Modbus master station protocol, where;

S1 is the slave address (high byte), communication command (low byte, defined by MODBUS protocol)

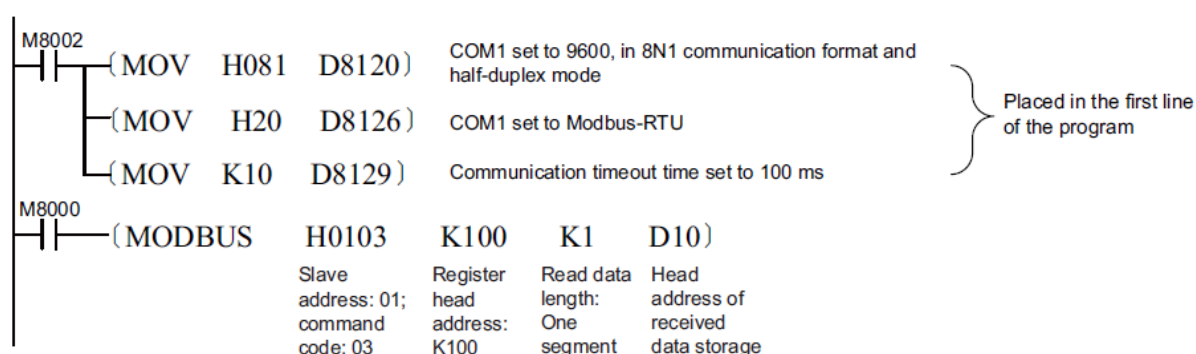
S2 is the starting address number of the register to access the slave station

n Data length read or written in advance

D is the starting address of the storage unit for reading or writing data, occupying the subsequent address unit, and the length is determined by n.



Example 1:



This instruction enables Modbus-compliant slave devices (such as MD310/290/5000 series frequency converters) to make communication, which is very convenient. In Example 1, the PLC frequently reads the register with the address of 100 in the #1 slave device and stores the data in the D10 unit.

Note: The MODBUS instruction can be used multiple times in a program. Different from the RS standard, the Modbus protocol allows the simultaneous execution of multiple MODBUS instructions. Compared with the RS instruction, the MODBUS instruction does not need to set M8122 to ON or reset M8123.

16-bit	32-bit	P	CANTX	CAN data transfer
✓				
13 steps			Instruction format: CANTX S1 S2 D n	

16-bit	32-bit	P	CANRX	CAN data transfer
✓				
13 steps			Instruction format : CANRX S1 S2 D n	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
D													✓			
n					✓	✓							✓			

H1U series main module PLC and H2U MRN (MTN) series main module PLC can form CAN network through CAN expansion card.

In the CAN network, through the CANTX/CANRX instructions, the user can write any CAN communication user protocol. The parameters are defined as follows:

S1 and S2 two parameters together form the CAN address: bit15 and bit14 of S1 are reserved, bit13 of S1 is the CAN address bit setting, "0" means standard CAN address (11 bits), "1" means extended CAN address (29 bit); if it is an 11-bit address, the address is represented by bit0~bit10 of S2. At this time, S1 is 0, and S2 cannot be greater than H7FF. If it is a 29-bit address, it consists of S2 (lower 16-bit address) and bit0~bit12 (higher 13-bit address) of S1. So bit13 of S1 is 1.

D is the sending buffer in the CANTX command, and the receiving buffer in the CANRX command; the maximum 4 D components starting from the D component are used as the sending or receiving buffer.

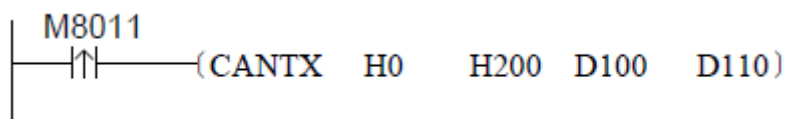
n In the CANTX instruction, it is the number of sent data, and in the CANRX instruction, it is the number of received data; in bytes, the maximum is 8.

A CANTX/CANRX instruction can only send/receive 8 bytes at most, but multiple CANTX/CANRX instructions can be written in the program to exchange more data without time-sharing driving.

Note: M8000 can be used to drive CANRX to receive commands at the same time, but CANRX can only be received one by one. The first receiving is completed, or the receiving timeout is over, then the next command can be received and executed in sequence. This time sharing is done by the PLC software, rather than the user program, there is no need for users to do programs for time sharing. If the ID of the sending command and the receiving command cannot match, the data will be lost.

## Example 1:

10ms to send a group of data, the buffer is D100~103, D110 stores the number of bytes sent



If D110 = k8, D100 = h1234, D101 = h5678, D102 = h9ABC, D103 = hDEF0 then the transferred data is shown in following table:

1 <sup>st</sup> byte	2 <sup>nd</sup> byte	3 <sup>rd</sup> byte	4 <sup>th</sup> byte	5 <sup>th</sup> byte	6 <sup>th</sup> byte	7 <sup>th</sup> byte	8 <sup>th</sup> byte
h12	h34	h56	h78	h9A	hBC	hDE	hF0

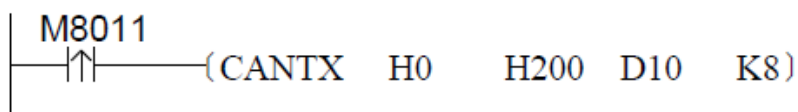
If the data is received from high byte to low byte, then the received data register of CAN communication will be MDL = h12345678, MDH = h9ABCDEF0

If D110 = k1, then only one byte is transferred: h12

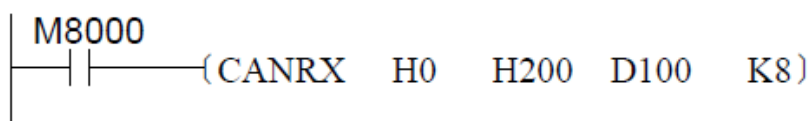
If D110 = k3, then the first three bytes are transferred: h12, h34, h56 and so on.

## Example 2: (bit13 of S1 is 0)

Suppose there are 8 PLC's connected to the CAN network, and one of the PLC programs has the following sending instructions written in it.



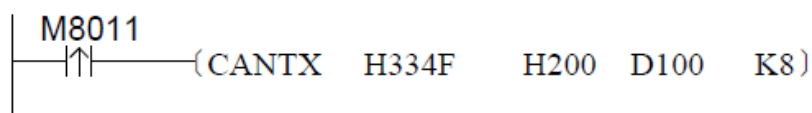
This PLC sends the data in the D10~D13 registers to address H200. Because the CAN protocol does not distinguish between master and slave stations, the data sent by this PLC to address H200 is open, and any other PLC in the network wants to receive the data of D10~D13 in this PLC can be written into the program of any PLC as follows to receive commands:



As long as the above statement is executed, the data with the receiving address H200 can be stored in D100~D103. You can write programs to receive this data in multiple PLC's.

Similarly, if any PLC wants to send data, CANTX instruction can also be written in the program. The address of S1 S2 can be freely defined by the user (note that the definition rules of 11-bit identifier and 29-bit identifier must be followed), as long as the receiver writes the same address in the CANRX instruction, the data of this user-defined address is received.

Example 3: (bit13 of S1 is 1)



Here, S1 is H334F (binary 11, 0011, 0100, 1111), and S2 is H200 (0000, 0010, 0000, 0000). The bit13 of S1 is 1, which represents a 29bit address. At this time, the address is composed of S2 (lower 16-bit address) and bit0~bit12 (higher 13-bit address) of S1, that is, 11 of S1.

0011, 0100, 1111 and S2 together form the address 11,0011, 0100, 1111, 0000, 0010, 0000, 0000 (H334F0200 in hexadecimal). That is, the PLC sends the data in the D100~D103 registers to the address H334F0200. In the equipment that meets the CANlink protocol or the remote expansion module access protocol, through the expansion module FROM/TO instruction, the remote expansion module and remote PLC connected via CAN can be read and written.

In the free CAN protocol, there is no need to allocate the station number of the PLC. In the equipment of the CANlink protocol or remote expansion module access protocol, the station number of each PLC or remote expansion module needs to be allocated. For details, please refer to "5.9 CAN Communication Command Description".

16-bit	32-bit	P	RS2	Serial data transmission 2
✓				
11 steps			Instruction format : RS2 S m D n n1	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
m					✓	✓							✓			
D													✓			
n					✓	✓							✓			
n1					✓	✓							✓			

This instruction is a communication sending and receiving instruction, which automatically sends the data in the designated register area to the serial port in turn, and stores the data received by the serial port in the designated area, which is equivalent to the user program directly accessing the communication buffer, realize the communication of the custom protocol. The RS command is fixed to COM1, while the RS2 command can use any COM port with the same method, where:

S is the starting address of the register area where the data to be sent is stored

m is the length of the data to be sent (number of bytes), the value range is 0~256

D is the starting address of the storage register area of the communication received data

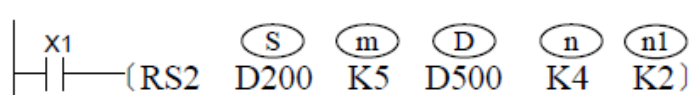
n is data length (number of bytes) received in this communication, the value range is 0~256

n1 is the serial port number used: 0: COM0; 1: COM1; 2: COM2; 3: COM3;

The RS2 instruction half-duplex/full-duplex mode is set by Bit10 of D8120. The user program can write multiple RS2 instructions, but only one RS2 instruction can be driven at the same time.

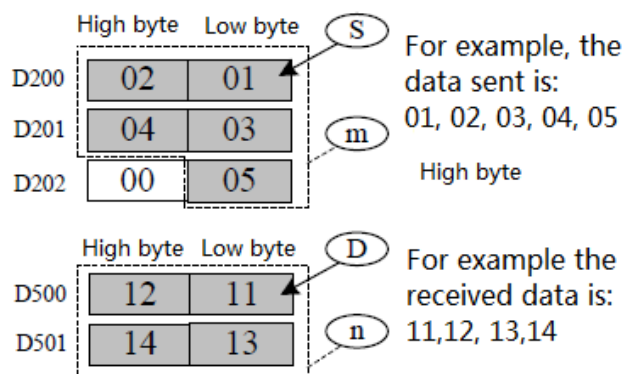
Before each RS2 command is driven, M8122 must be set.

Example:



(n1) Set to K2, RS2 command is used for COM2 communication port

When X1 is ON, the sending and receiving data of communication after the command is executed are stored as shown in the right figure.



16-bit	32-bit	P	MODBUS2	Data transfer 2
✓				
11 steps			Instruction format: MODBUS2 S1 S2 n D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
n					✓	✓							✓			
D																
n1					✓	✓							✓			

For intuitive convenience, we can use MODBUS2 to replace the RS2 instruction in the standard MODBUS master station protocol communication. MODBUS instruction is fixed to COM1,

The MODBUS 2 command can use any COM port, and the method is the same. The definitions are as follows:

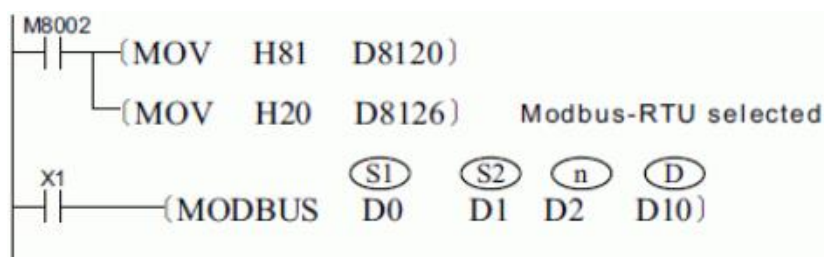
S1 is the slave address (high byte), communication command (low byte, defined by MODBUS protocol);

S2 is the starting address number of the register to access the slave station;

n Data length read or written in advance;

D is the starting address of the storage unit for reading or writing data, occupying the subsequent address unit, and the length is determined by n.

n1 is the serial port number used: 0: COM0; 1: COM1; 2: COM2; 3: COM3;



Here, n1 set to K2, MODBUS2 command is used for COM2 communication port.

## 4.3.10 Floating number (110~147)

16-bit	32-bit	P	ECMP	Binary floating-point comparison
	✓	✓		
	13 steps	13 steps	Instruction format: ECMP S1 S2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
D		✓	✓	✓												

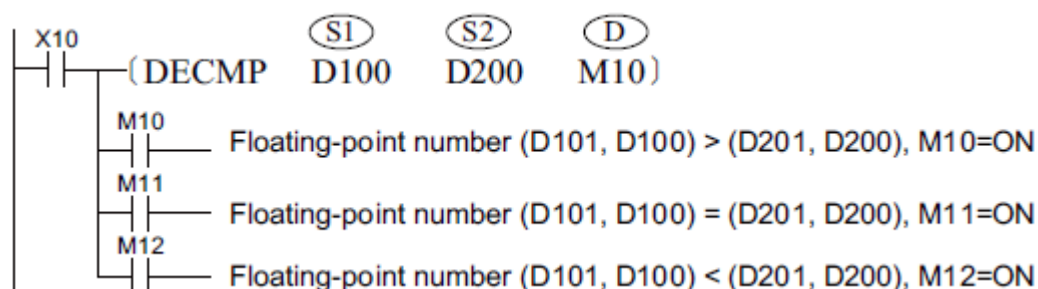
The values of two floating-point variables are compared. The comparison result is output to three variables from head address D, where:

S1 is the binary floating-point number 1 to be compared;

S2 is the binary floating-point number 2 to be compared;

D is the storage unit of the comparison result, occupying 3 (bit) variable units in total.

Example:



When X10 = ON, M10, M11, or M12 switches to ON.

The DECMP 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  $\geq$ ,  $\leq$ , and  $\neq$  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.



16-bit	32-bit	P	EZCP	Binary floating-point range comparison
	✓	✓		
	17 steps	17 steps	Instruction format: EZCP S1 S2 S D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓							✓		
S2					✓	✓							✓		
S					✓	✓							✓		
D		✓	✓	✓											

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.

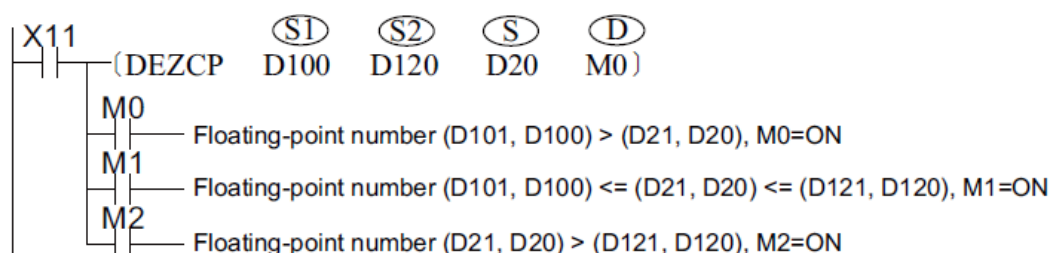
S1 is the lower limit of a binary floating-point variable range.

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.

16-bit	32-bit	P	EBCD	Conversion from binary to decimal floating-point number format
	✓	✓		
	9 steps	9 steps	Instruction format: EBCD S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
D													✓			

This instruction is to transfer binary floating-point data. Contact drive is required. When the instruction is executed, the binary floating-point data value of S is copied to D, where:

S is the transfer source of binary floating-point data

D is the storage unit for storing binary floating-point data

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  $D2 \times 10^{D3}$  in scientific notation.

The PLC uses only binary floating-point numbers for calculation. Convert binary floating-point numbers to decimal equivalents for easy monitoring.

16-bit	32-bit	P	EBIN	Conversion from decimal to binary floating-point number format
	✓	✓		
	9 steps	9 steps	Instruction format: EBIN S D	

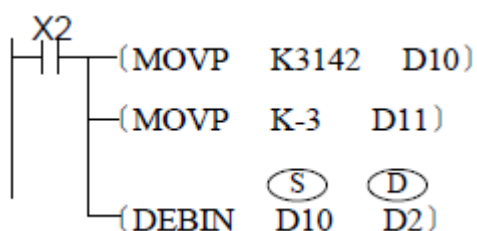
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
D													✓			

This instruction is to convert a binary floating-point number to a decimal floating-point. where:

S is a binary floating-point variable;

D is the storage unit of the result converted to decimal floating-point number.

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).

16-bit	32-bit	P	EADD	Binary floating-point addition
	✓	✓		
	13 steps	13 steps	Instruction format: EADD S1 S2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
D													✓			

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.

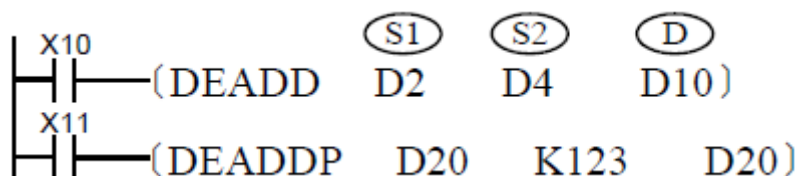
If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the addition operation.

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.

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.

16-bit	32-bit	P	ESUB	Binary floating-point subtraction
	✓	✓		
	13 steps	13 steps	Instruction format: ESUB S1 S2 D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✓	✓							✓		
S2					✓	✓							✓		
D													✓		

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.

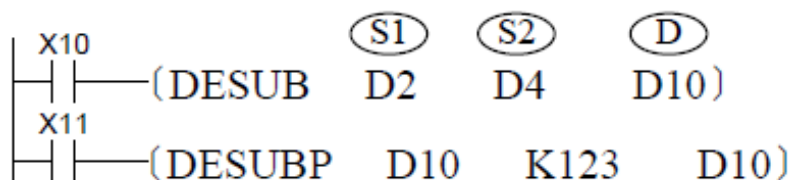
If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the subtraction operation.

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:



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.

16-bit	32-bit	P	EMUL	Binary floating-point multiplication
	✓	✓		
	13 steps	13 steps	Instruction format: EMUL S1 S2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
D													✓			

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.

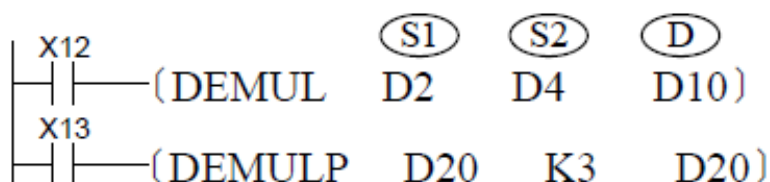
If the constant K or H is used as the operand S1 or S2, the value is converted to a binary floating-point number before the multiplication operation.

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.

16-bit	32-bit	P	EDIV	Binary floating-point division
	✓	✓		
	13 steps	13 steps	Instruction format: EDIV S1 S2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓							✓			
S2					✓	✓							✓			
D													✓			

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.

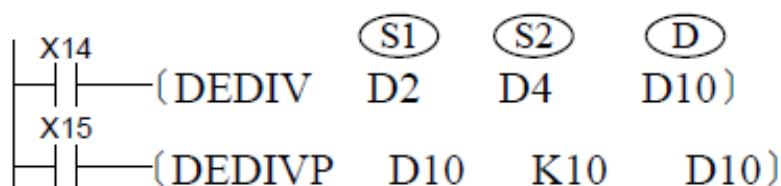
The zero flag M8020 is set if the result of the calculation is 0.

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.

16-bit	32-bit	P	EXP	Binary floating-point exponent operation
	✓	✓		
	9 steps	9 steps	Instruction format: EXP S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S														✓	✓	✓
D														✓	✓	✓

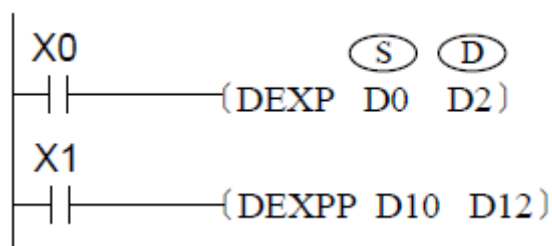
Exponentiation is performed on the mathematical constant  $e$  (approximately equal to 2.71828). The exponent is a binary floating-point number.

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).  $e(D1, D0) \rightarrow (D3, D2)$ .

Because  $\log_e 2^{128} = 88.7$ , when the value in (D1, D0) is greater than 88.7, then D8067 = K6706 and M8067 = ON.



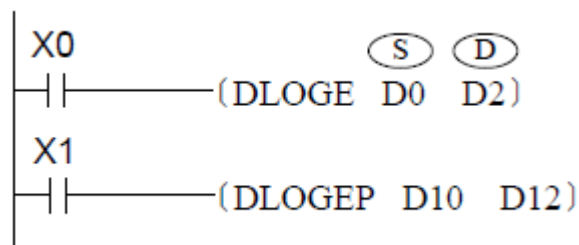
16-bit	32-bit	P	LOGE	Binary floating-point natural logarithm operation
	✓	✓		
	9 steps	9 steps	Instruction format: LOGE S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S														✓	✓	✓
D														✓	✓	✓

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. where:

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):

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

16-bit	32-bit	P	DLOG	Binary floating-point logarithm operation with base 10
	✓	✓		
	9 steps	9 steps	Instruction format: DLOG S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

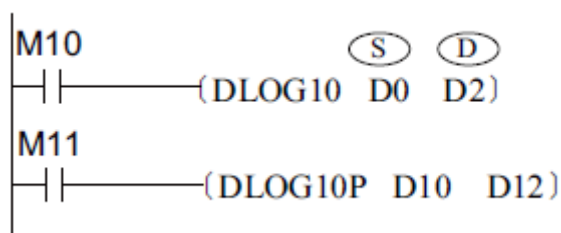
The common logarithm of a binary floating-point number is calculated. The base is 10, where:

S is the binary floating-point variable of the common logarithm of the binary floating-point number to be determined;

D is the storage unit of the calculation result after common logarithmic operation.

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.

$$\log_{10}(D1, D0) \Rightarrow (D3, D2)$$

16-bit	32-bit	P	ESQR	Binary floating-point square root operation
	✓	✓		
	9 steps	9 steps	Instruction format: ESQR S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S					✓	✓							✓			
D													✓			

The square root of a binary floating-point number is calculated, where:

S is the binary floating-point variable to be square rooted;

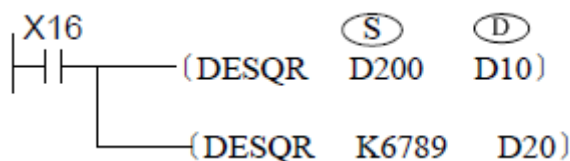
D is the storage unit of binary floating-point square root.

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.

16-bit	32-bit	P	INT	Conversion from binary floating-point number format to BIN integer format
✓	✓	✓		
5 steps	9 steps		Instruction format: INT S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
D													✓			

Binary floating-point numbers are rounded to remove the fractional part. The result is stored in D, where:

S is the binary floating-point number variable to be rounded and transformed

D is the storage unit of the converted BIN integer result

M8020 is set when  $S = 0$ .

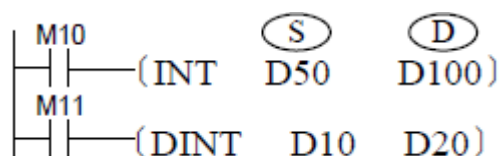
The borrow flag M8021 is set when  $|S| \leq 1$ .

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.

16-bit	32-bit	P	SIN	Floating-point SIN operation
	✓	✓		
	9 steps	9 steps	Instruction format: SIN S D	

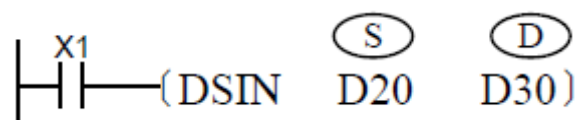
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
D													✓			

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  $0 \leq \alpha \leq 2\pi$ .

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

Example 1:



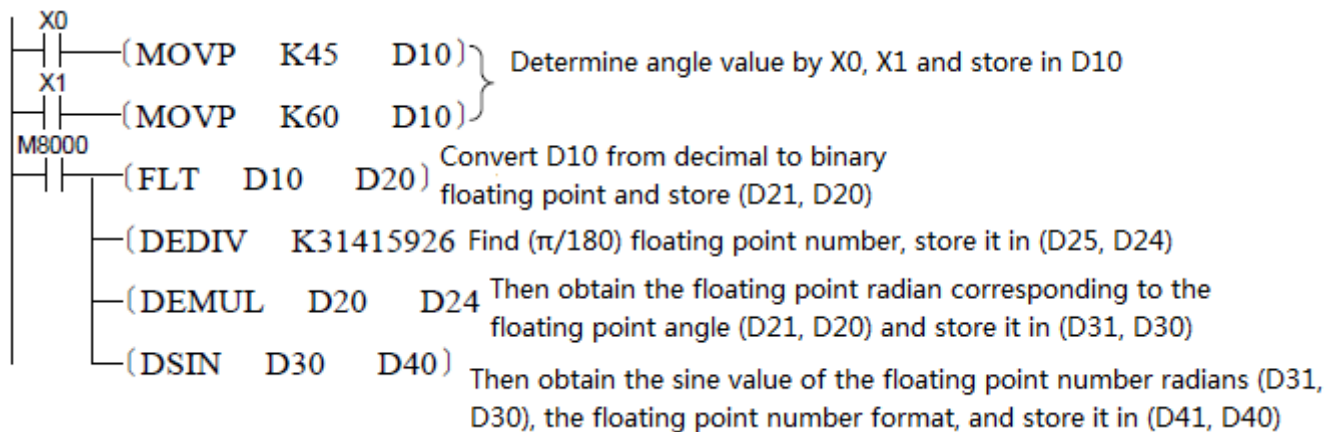
The tangent of the angle measured in radians in (D21, D20) is calculated. The result is stored in (D31, D30).

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

According to the equation Angle in radians = Angle in degrees  $\times \pi/180^\circ$ , an angle of  $360^\circ$  is converted to radians as follows:  $360^\circ \times \pi/180^\circ = 2\pi$ .

Example 2:

According to the angle value to find the corresponding SIN value program:



16-bit	32-bit	P	COS	Floating-point COS operation
	✓	✓		
	9 steps	9 steps	Instruction format: COS S D	

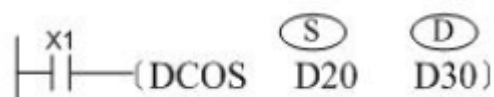
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
D													✓			

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

S is the angle variable for which the cosine value is to be calculated, the unit of RAD is expressed as a binary floating-point number. Value range  $0 \leq \alpha \leq 2\pi$ ;

D is the storage unit of the converted COS calculation result, in binary floating-point format.

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 degree  $\times \pi/180^\circ$ , an angle of  $360^\circ$  is converted to radians as follows:  $360^\circ \times \pi/180^\circ = 2\pi$ .

For details about the programming statement for calculating the cosine in degrees, see the example described in [“SIN: Floating-point SIN operation”](#).

16-bit	32-bit	P	TAN	Floating-point TAN operation
	✓	✓		
	9 steps	9 steps	Instruction format: TAN S D	

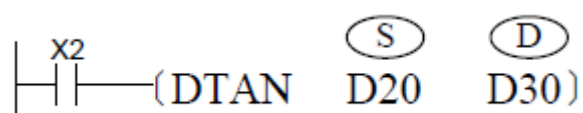
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓			
D													✓			

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

S is the angle variable whose tangent value is to be calculated, and the unit of RAD is expressed as a binary floating-point number. Value range  $0 \leq \alpha < 2\pi$ ;

D is the storage unit of the converted TAN calculation result, in binary floating-point format.

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 TAN operation result are in binary floating-point number format.

According to the equation Angle in radians = Angle in degree  $\times \pi/180^\circ$ , an angle of  $360^\circ$  is converted to radians as follows:  $360^\circ \times \pi/180^\circ = 2\pi$ .

For details about the programming statement for calculating the tangent in degrees, see the example described in [“SIN: Floating point SIN operation”](#).

16-bit	32-bit	P	ASIN	Binary floating-point ARCSIN operation
	✓	✓		
	9 steps	9 steps	Instruction format: ASIN S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓	✓	✓	
D													✓	✓	✓	

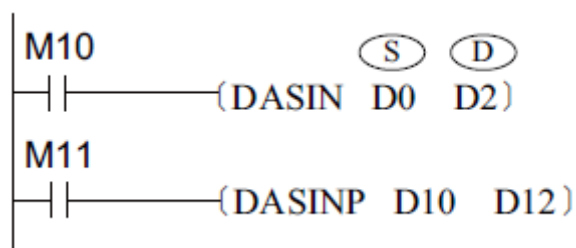
The sine in radians is calculated, where:

S is the binary floating-point variable that stores the  $\text{SIN}^{-1}$  (arc sine) to be calculated;

D is the storage unit of the calculation result ( $-\pi/2 \sim \pi/2$ ).

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.

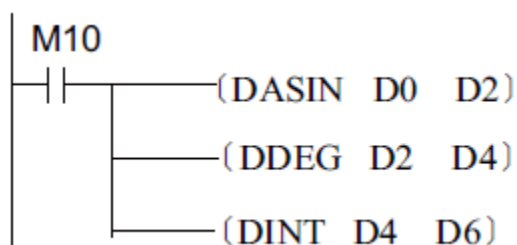
Example 1:



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

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

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.



16-bit	32-bit	P	ACOS	Binary floating-point ARCCOS operation
	✓	✓		
	9 steps	9 steps	Instruction format: ACOS S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓	✓	✓	
D													✓	✓	✓	

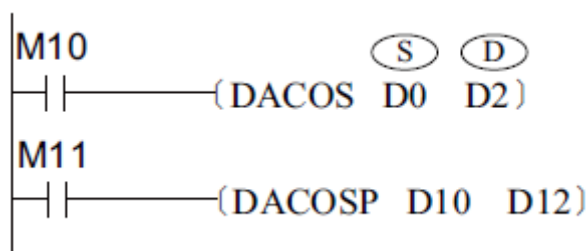
The cosine in radians is calculated, where;

S is the binary floating-point variable that stores the  $\text{COS}^{-1}$  (arc cosine) to be calculated;

D is the storage unit of the calculation result ( $0 \sim \pi$ ).

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.

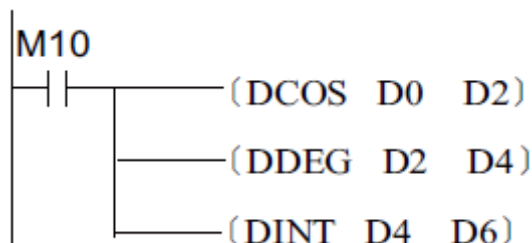
Example 1:



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

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

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.

16-bit	32-bit	P	ATAN	Binary floating-point ARCTAN operation
	✓	✓		
	9 steps	9 steps	Instruction format: ATAN S D	

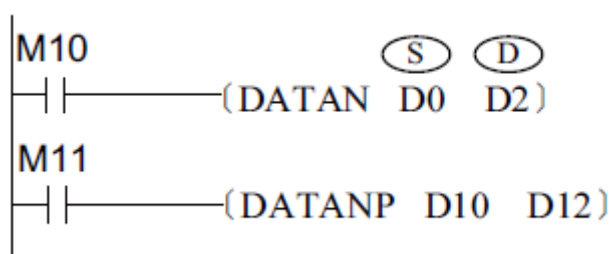
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓	✓	✓	
D													✓	✓	✓	

The tangent is radians calculated, where:

S is for storage pending  $TAN^{-1}$  (Arctangent) binary floating-point variable

D is the storage unit of the calculation result ( $-\pi/2 \sim \pi/2$ ).

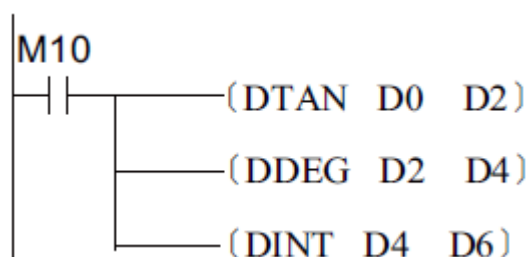
Example 1:



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

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

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.

16-bit	32-bit	P	RAD	Binary floating-point degree to radian conversion
	✓	✓		
	9 steps	9 steps	Instruction format: RAD S D	

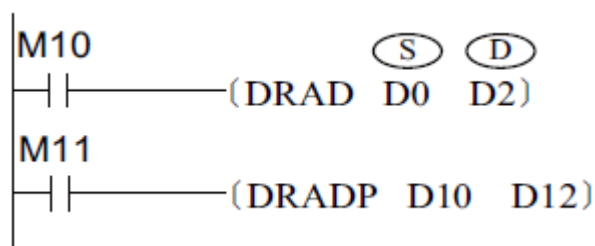
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓	✓	✓	
D													✓	✓	✓	

Binary floating-point degrees are converted to radians. The calculation formula is [Angle in radians = Angle in degree  $\times \pi/180$ ], where:

S is a binary floating-point angle variable that stores the radian to be determined;

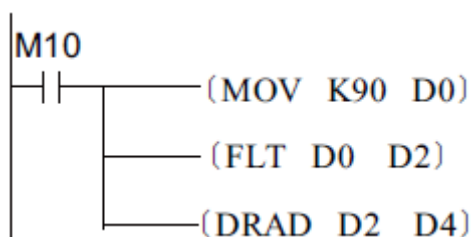
D is the storage unit of the calculation result.

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  $\pi/2$ , that is, 1.570796.

16-bit	32-bit	P	DEG	Binary floating-point radian to degree conversion
	✓	✓		
	9 steps	9 steps	Instruction format: DEG S D	

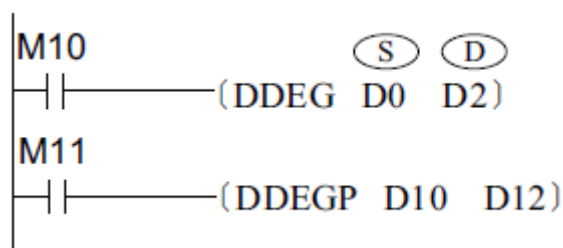
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓	✓	✓	
D													✓	✓	✓	

The DEG instruction converts binary floating-point radians to degrees. The calculation formula is [Angle in degrees = Angle in radians x  $\pi/180$ ], where:

S is a binary floating-point radian variable storing the angle to be determined;

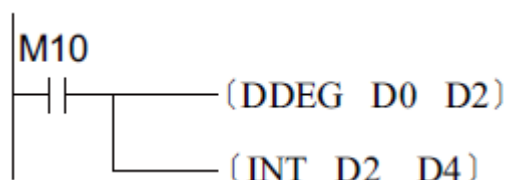
D is the storage unit of the calculation result.

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.

16-bit	32-bit	P	SINH	Binary floating-point SINH operation
	✓	✓		
	9 steps	9 steps	Instruction format: SINH S D	

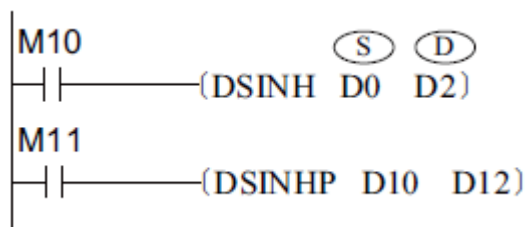
Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✓	✓	✓
(D)													✓	✓	✓

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

S is the binary floating-point variable that stores the SINH value to be calculated;

D is the storage unit of the calculation result.

Example:



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

16-bit	32-bit	P	COSH	Binary floating-point COSH operation
	✓	✓		
	9 steps	9 steps	Instruction format: COSH S D	

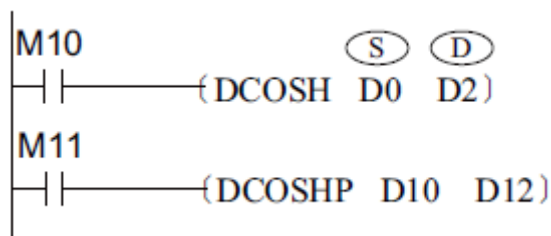
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S													✓	✓	✓	
D													✓	✓	✓	

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

S is the binary floating-point variable that stores the COSH value to be calculated;

D is the storage unit of the calculation result.

Example:



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

16-bit	32-bit	P	TANH	Binary floating-point TANH operation
	✓	✓		
	9 steps	9 steps	Instruction format: TANH S D	

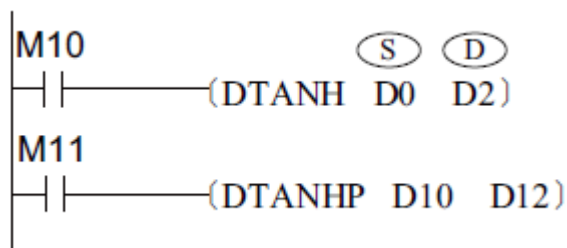
Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

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

S is the binary floating-point variable that stores the TANH value to be calculated;

D is the storage unit of the calculation result.

Example:



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

16-bit	32-bit	P	SWAP	Higher and lower byte swap
✓	✓	✓		
3 steps	5 steps		Instruction format: SWAP S	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S								✓	✓	✓	✓	✓	✓	✓	✓	

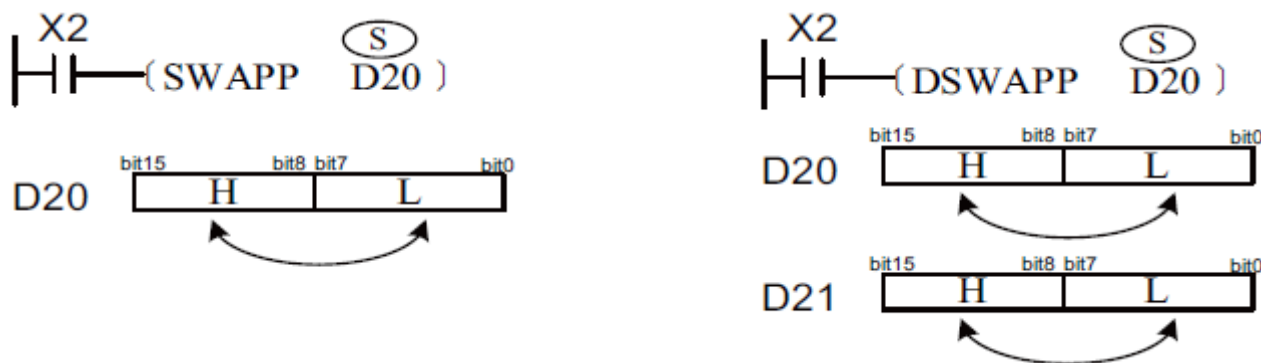
The higher and lower bytes of the variable in S are exchanged.

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.



## 4.3.11 Location control (155~159)

16-bit	32-bit	P	DSZR	DOG search return to origin
✓	✓			
3 steps	17 steps		Instruction format: DSZR S1 S2 D1 D2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1	✓		✓	✓												
S2	✓															
D1		✓														
D2		✓	✓													

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 or Y2

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)

Note: The M signal used by the above bit device has a range of M0~M1023.

List of related software components:

1) Special auxiliary relay:

Soft element			Name	Attributes
Y0	Y1	Y2		
M8029			Instruction execution end flag	Read
M8049			Instruction execution abnormal end flag	Read
M8040 *1	M8450	M8460	Monitoring during pulse output	Read
M8041 *1	M8451	M8461	The clear signal output function is valid *2	Read and write

Soft element			Name	Attributes
Y0	Y1	Y2		
M8042	M8452	M8462	Return to origin direction designation *2	Read and write
M8043	M8453	M8463	Forward limit	Read and write
M8044	M8544	M8464	Reverse limit	Read and write
M8045	M8545	M8465	Near-point signal logic inversion *2	Read and write
M8046	M8546	M8466	Zero signal logic inversion *2	Read and write
M8048	M8548	M8468	Positioning command driving	Read
M8049	M8549	M8469	Pulse stop command *2	Read and write
M8484	M8485	M8486	Clear signal device designation function is valid*2	Read and write

\*1: This port and related software component are reserved.

\*2: When the status changes from RUN to STOP, the signal is cleared.

## 2) Special data register:

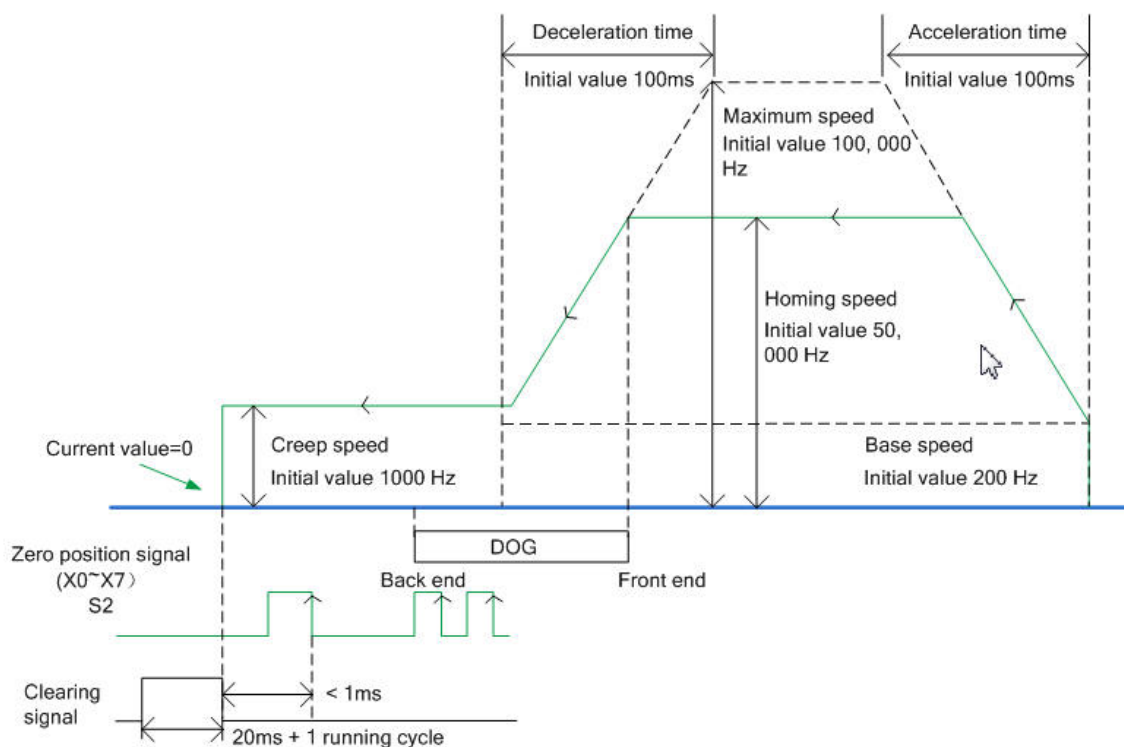
Soft element						Name	Default	Range *2
Y0		Y1		Y2				
D8440 *1	Lower bit	D8450	Lower bit	D8460	Lower bit	Current value register (Hz) (32 bits)	0	/
D8441 *1	Higher bit	D8451	Higher bit	D8461	Higher bit			
D8442		D8452		D8462		Base speed (Hz) (16-bit)	200	10~32767
D8443	Lower bit	D8453	Lower bit	D8463	Lower bit	Maximum speed (Hz) (32-bit)	100,000	715~100,000
D8444	Higher bit	D8454	Higher bit	D8464	Higher bit			
D8445		D8455		D8465		Creeping speed (Hz) (16-bit)	1000	10~32767
D8446	Lower bit	D8456	Lower bit	D8466	Lower bit	Origin return speed (Hz) (32-bit)	50,000	715~100,000
D8447	Higher bit	D8457	Higher bit	D8467	Higher bit			
D8448		D8458		D8468		Acceleration time (ms) (16-bit)	100	50~5000
D8449		D8459		D8469		Deceleration time (ms) (16-bit)	100	50~5000
D8484		D8485		D8486		Clear signal device designation	/	/

\*1: This port and related software component are reserved.

\*2: The following rules must be followed,

"creep speed  $\geq$  base speed", if creep speed  $< 0$ , creep speed  $<$  base speed, return speed  $<$  base speed, maximum speed  $<$  creep speed, an error will be reported [Report component overrun (6706)]; When other speeds are related, the system will automatically adjust to the close value.

Detailed analysis of parameters:



### 3) Logic inversion of near point signal and zero signal

D1 Pulse Output End	Near Point Signal (DOG) Logic Specified by S1 (Near Point Signal Logic Inversion Flag Bit)	Zero Signal Logic Specified by S2 (Zero Signal Logic Inversion Flag Bit)	Remarks
Y0	M8445	M8446	OFF: positive logic (When the input is ON, the near point/zero signal is ON.) ON: negative logic (When the input is OFF, the near point/zero signal is ON.)
Y1	M8455	M8456	
Y2	M8465	M8466	

### 4) Clearing signal output

Pulse Output End	Clearing Signal Output Effectiveness Flag Bit	Effectiveness Flag Bit of the Function Specified by the Clearing Signal Soft Element	Soft Element Specified by the Clearing Soft Element	Soft Element Default Value Specified by the Clearing Soft Element
Y0	M8441	M8484	D8484	D8484 = 4 (Y4)
Y1	M8451	M8485	D8485	D8485 = 5 (Y5)
Y2	M8461	M8486	D8486	D8486 = 6 (Y6)

## [Note]

a) After the origin regression ends, the clearing signal will be output. Therefore, the clearing signal output effectiveness flag bit needs to be set to ON.

b) If the effectiveness flag bit of the function specified by the clearing signal soft element is OFF, the clearing signal is output by the default Y port. Otherwise, the clearing signal is output by the Y port specified by the clearing soft element.

## 5) Velocity control

For the maximum speed, origin regression speed, creeping speed, and base speed, see definitions in the description of special data registers. Observe the following requirements:

Base speed = origin regression speed = maximum speed

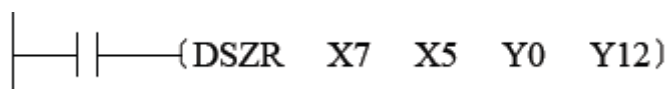
Base speed = creeping speed = maximum speed

Select proper parameters based on the preset parameter range. It is recommended that the creeping speed be smaller than or equal to the origin regression speed.

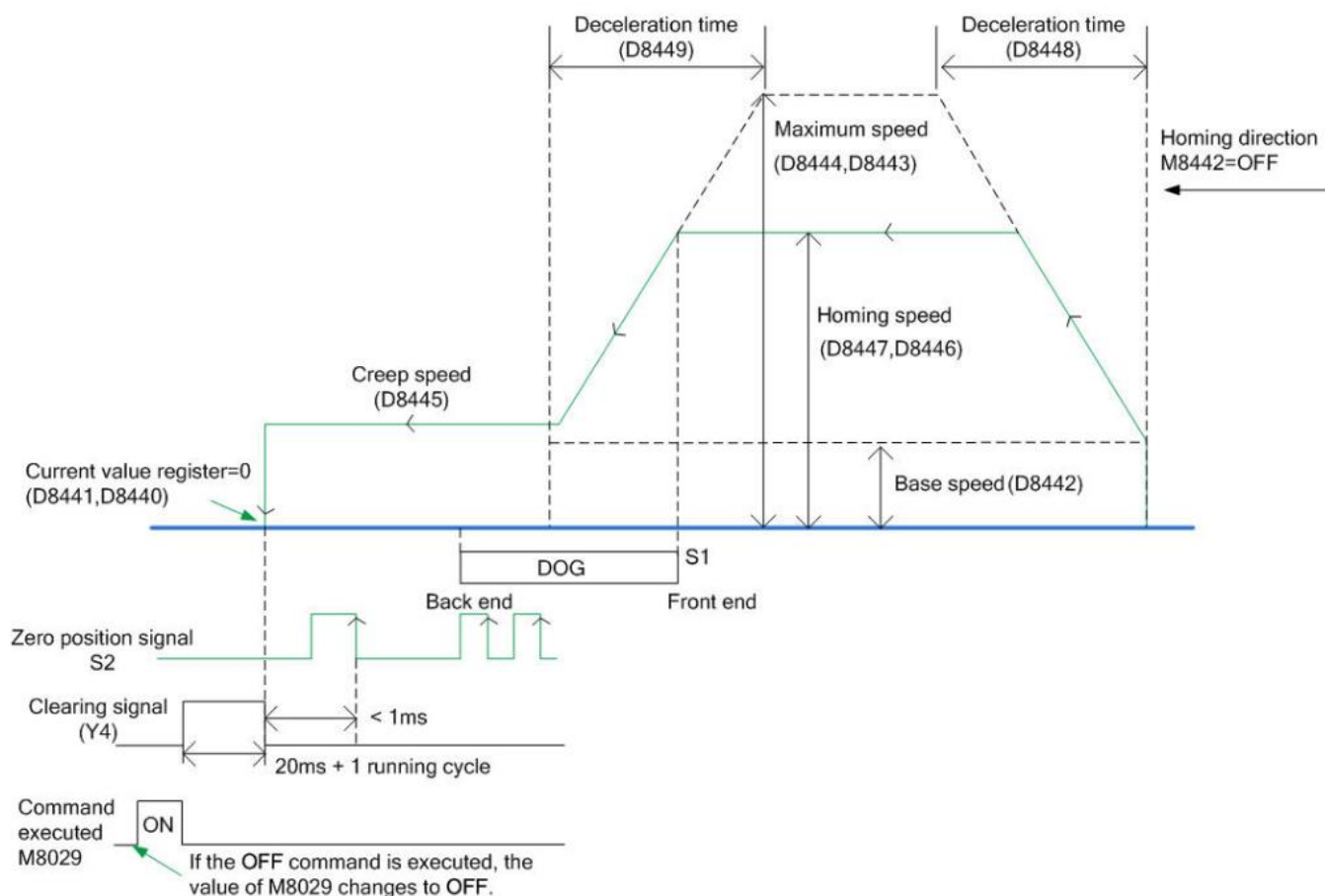
Note: If the base speed is set to a very low value, the output frequency jumps fiercely in the start, acceleration, and deceleration phases (front several output pulses). A lower base speed indicates fiercer frequency jump. It is recommended to set the base speed to a value greater than 50 (to ensure smooth acceleration).

**Origin regression:**

Example:



The following uses the pulse output port Y0 as an example. Modify soft elements when another port is used.

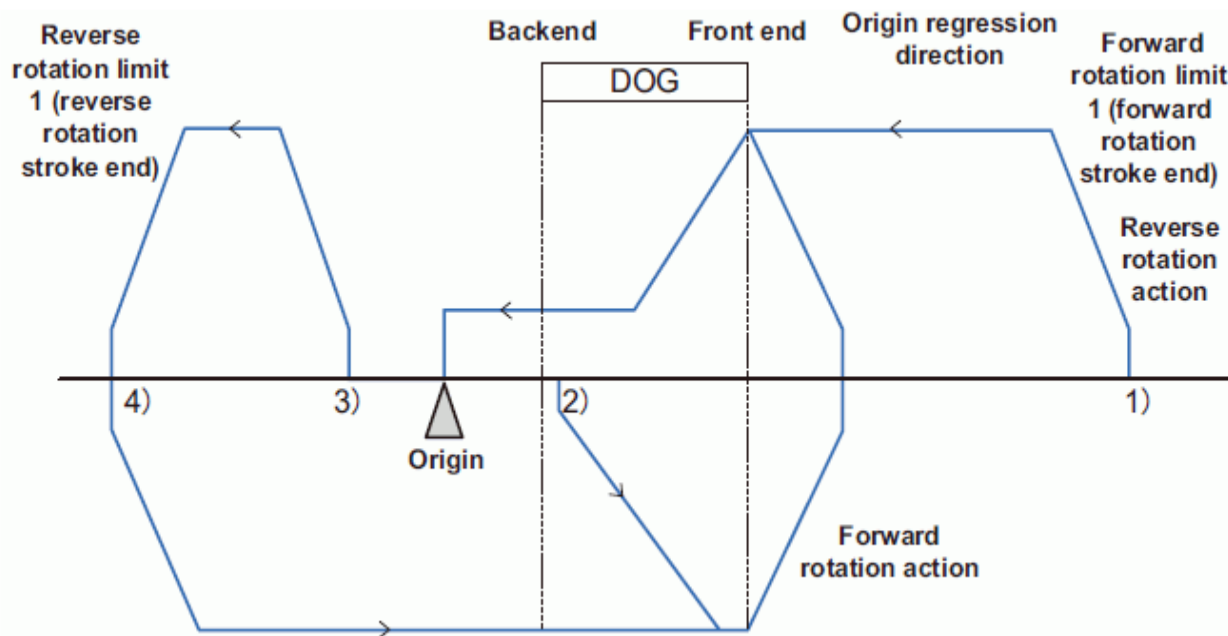


- 6) Designation of the origin regression direction: Specify the origin regression direction by setting the origin regression direction flag bit (M8442) to ON or OFF.
- 7) The base speed (D8442) increases to the origin regression speed (D8447, D8446), and the actuator moves in the direction specified by the origin regression direction flag bit (M8442).
- 8) When it is detected that near point signal (DOG) specified by S1 is ON, the output port starts deceleration to the creeping speed (D8445).
- 9) After the near point signal (DOG) specified by S1 turns from ON to OFF, if it is detected that the zero signal specified by S2 turns from OFF to ON, the output port immediately stops the pulse output.
- [Note] If the same input is specified by the near point signal and zero signal, this instruction does not use the zero signal just as the ZRN instruction does. After the near point signal turns from ON to OFF, the instruction immediately stops the pulse output. If the same input is specified by the near point signal and zero signal and the logic inversion flag bit (M8445 or M8446) is ON, the logic of the near point signal (M8445) shall prevail.
- 10) When the clearing signal output function (M8441) is valid (ON), after the pulse output is stopped (within 1ms), the clearing signal (Y4) keeps in the ON state within [20 ms + 1 operation period].
- 11) When the instruction execution end flag bit (M8029) is ON, the origin regression action ends.
- [Note] The preceding is based on the case in which the near point signal logic inversion flag bit (M8445) and the zero signal logic inversion flag bit (M8446) are OFF. If the logic flag bits are ON, ON needs to be changed to

OFF and OFF needs to be changed to ON for the near point signal and zero signal.

### DOG search function:

When the positive and reverse limits are designed, the origin regression using the DOG search function is executed. The start position of origin regression is different, and therefore the origin regression action is different.



- 1) When the start position is in front of DOG, including when the forward limit 1 is ON:
  - The origin regression instruction is executed to start the origin regression action.
  - The actuator moves in the origin regression direction at the origin regression speed.
  - When the DOG front end is detected, the output port decelerates to the creeping speed.
  - After the DOG rear end is detected, the pulse output is stopped when the first zero signal is detected.
- 2) When the start position is within DOG:
  - The origin regression instruction is executed to start the origin regression action.
  - The actuator moves in a direction opposite to the origin regression direction at the origin regression speed.
  - When the DOG front end is detected, the output port decelerates and stops pulse output. (Leave DOG.)
  - The actuator moves in the origin regression direction at the origin regression speed. (Enter DOG again.)
  - When the DOG front end is detected, the output port decelerates to the creeping speed.
  - After the DOG rear end is detected, the pulse output is stopped when the first zero signal is detected.
- 3) When the start position is at the near point signal OFF (after DOG):
  - The origin regression instruction is executed to start the origin regression action.
  - The actuator moves in the origin regression direction at the origin regression speed.
  - When reverse limit 1 (reverse limit) is detected, the output port decelerates and stops pulse output.
  - The actuator moves in a direction opposite to the origin regression direction at the origin regression

speed.

- When the DOG front end is detected, the output port decelerates and stops pulse output. (Leave DOG.)
  - The actuator moves in the origin regression direction at the origin regression speed. (Enter DOG again.)
  - When the DOG front end is detected, the output port decelerates to the creeping speed.
  - After the DOG rear end is detected, the pulse output is stopped when the first zero signal is detected.
- 4) When the position limit switch (reverse limit 1) of the origin regression direction is ON:
- The origin regression instruction is executed to start the origin regression action.
  - The actuator moves in a direction opposite to the origin regression direction at the origin regression speed.
  - When the DOG front end is detected, the output port decelerates and stops pulse output. (Leave DOG.)
  - The actuator moves in the origin regression direction at the origin regression speed. (Enter DOG again.)
  - When the DOG front end is detected, the output port decelerates to the creeping speed.
  - After the DOG rear end is detected, the pulse output is stopped when the first zero signal is detected.
  - [Note] When the near point signal (DOG) is counted, the ON duration should be sufficient for the output port to decelerate to the creeping speed. Keep the creeping speed as low as possible. If the output port stops pulse output without deceleration, the high creeping speed will cause position offset.

#### Notes:

- 1) Positioning instructions (ZRN/PLSV/DRVI/DRVA) can be used multiple times in the program. Do not execute the instructions on the same high-speed Y output port to perform the output operation at the same time.
- 2) After the instruction-driven flow turns OFF, if the driven flow needs to turn ON, it can be turned ON only after the state bit (Y000: [M8147], Y001: [M8148]) is OFF and one operation period elapses.
- 3) When the positioning instructions are driven again, there must be more than one OFF period. If re-driving is performed in less than the preceding time, an operation error will occur at the initial instruction execution (operation).

16-bit	32-bit	P	ABS	ABS reading location data
	✓			
	13 steps		Instruction format: ABS S D1 D2	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	✓	✓	✓	✓											
D1		✓	✓	✓											
D2								✓	✓	✓	✓	✓	✓	✓	✓

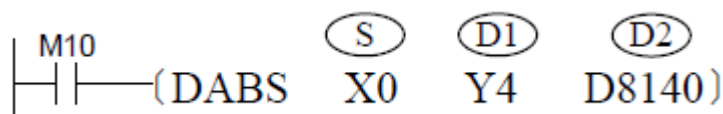
This instruction is to read the absolute position (ABS) data of the motor in the servo drive through the high-speed input port. where:

S is the input signal of the read servo device, occupying a total of 3 units in the following;

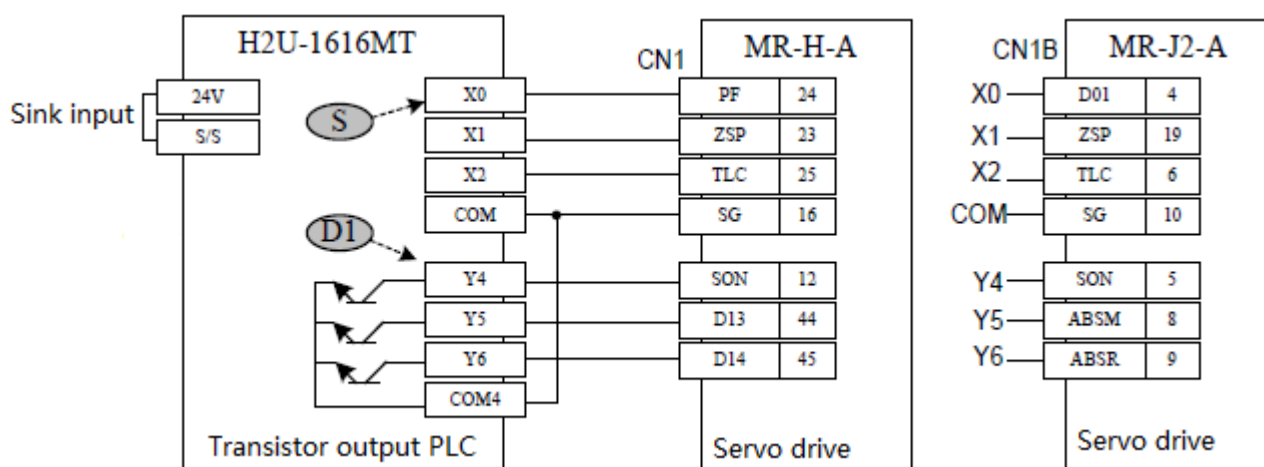
D1 is the control signal sent to the servo device, which occupies a total of 3 subsequent units;

D2 is the storage unit for reading data from the servo. It has a 32-bit width and occupies D2+1 and D2 units. Usually D8140 is designated.

Example:



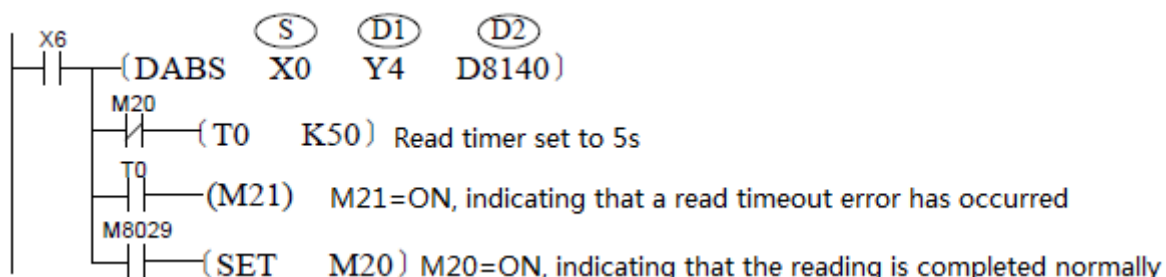
The corresponding wiring method is as shown in the figure below, in which the servo drive is a commercially available Mitsubishi company product, which is matched with a servo motor with an encoder for absolute position detection:



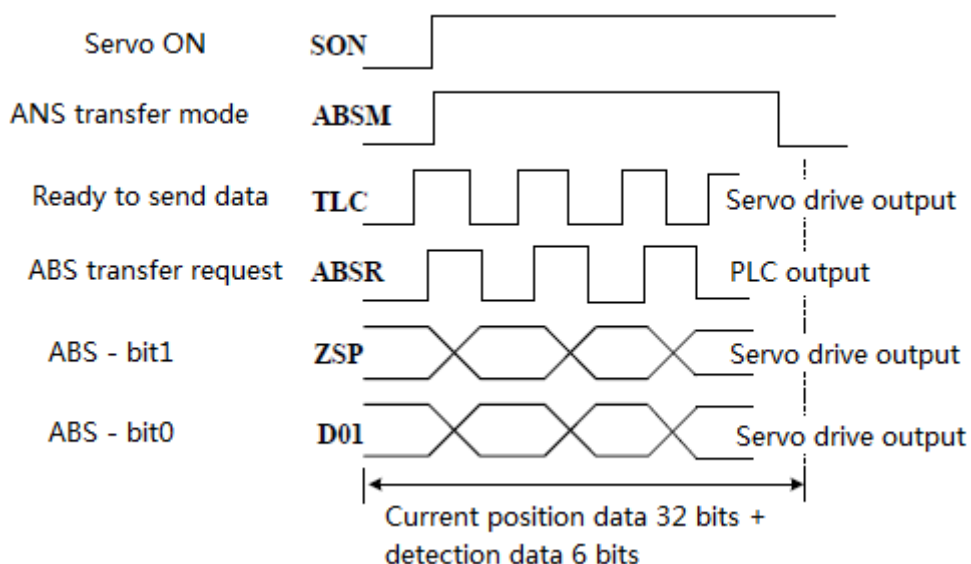
When the instruction drives M10 to turn ON, start to read. After reading is completed, the M8029 flag is set to ON;



When the instruction is executed, the drive flag becomes OFF, and the read operation is interrupted; An example of programming to read ABS data is as follows. ABS data is read only when the X6 terminal is closed. If the read operation is not completed within 5s, the timeout flag M21 is set. The programming is as follows:



The signal timing of the ABS read operation is as shown in the figure below. When the instruction is executed, the PLC will automatically complete the access operation with the servo drive according to this realization:



This instruction can only be executed in 32-bit mode, that is, the instruction is DABS.

16-bit	32-bit	P	ZRN	Regression through the origin
✓	✓			
9 step	17 step		Instruction format: ZRN S1 S2 S3 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S3	✓	✓	✓	✓												
D		✓														

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.

In the instruction:

S1 indicates the starting speed of the origin regression action. When the 16-bit instruction is executed, the value ranges from 10 Hz to 32,767 Hz; when the 32-bit instruction is executed, the value ranges from 10 Hz to 100,000 Hz.

S2 indicates the creeping speed after the origin signal turns ON. The value ranges from 10 Hz to 32,767 Hz.

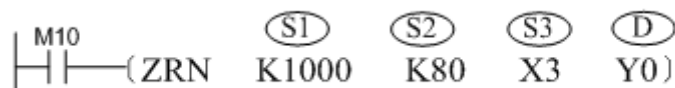
S3 indicates the origin signal (DOG) input. Though the X, Y, M, S signals are acceptable, only the X signal has the best timeliness.

D indicates the start address for pulse output.

For 3624MT/2416MT model, only Y0 or Y1 can be specified; for other MT models, Y0, Y1, or Y2 can be specified; for MTQ models, Y0, Y1, Y2, Y3, or Y4 can be specified.

When the relative position control instruction DRVI (FNC158) and absolute position control instruction DRVA are executed, the controller calculates the number of forward or reverse pulses transmitted by itself, and stores it in [D8141, D8140] (Y000) and [D8143, D8142] (Y001) in the register. Data in the register will be lost when a power failure occurs. Therefore, the origin regression instruction ZRN must be executed during power-on and initial running, to write the data of the origin position of the mechanical action to the register.

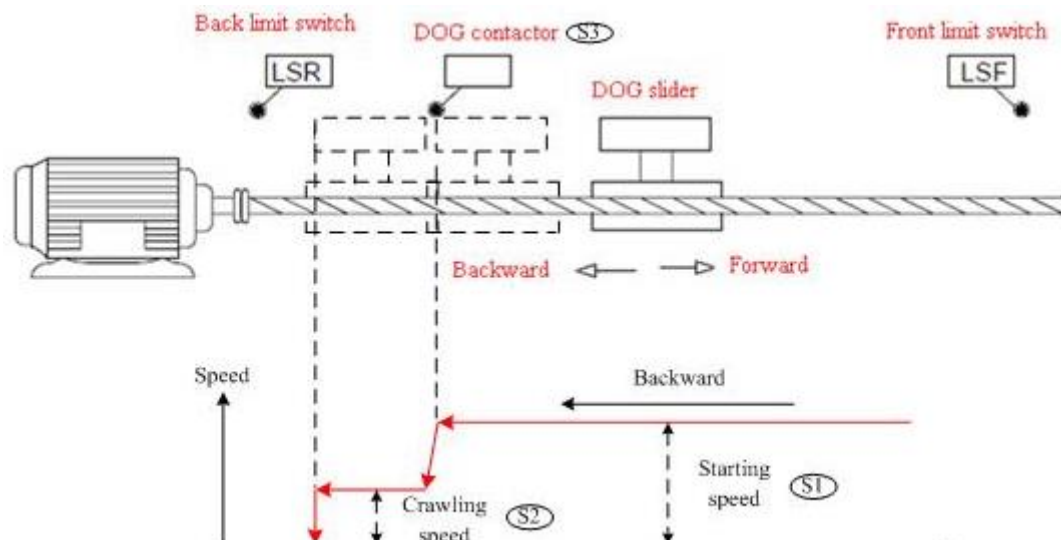
Example:



When M10 turns ON, the PLC transmits pulses at 1000 Hz through the Y0 high-speed output port, to make the stepper motor return to the origin. When the DOG signal turns ON (the DOG slider right touches the DOG contact), the output frequency decreases to 80 Hz and low-speed creeping is performed. When the DOG signal turns OFF again, Port Y0 stops outputting pulses and writes 0 to the current value register

(Y000:[D8141,D8140],Y001:[D8143, D8142]). In addition, when M8140 (clearing signal output function) is ON, the clearing signal is outputted. When the execution completion flag (M8029) is set to ON, the pulse output monitoring flag (Y000: [M8147], Y001: [M8148]) turns OFF.

See the figure below.



System variables involved in the execution of this instruction are as follows:

1. D8141 (high bit), D8140 (low bit): current value register outputted by Y000 (the 32-bit instruction is executed)
2. D8143 (high bit), D8142 (low bit): current value register outputted by Y001 (the 32-bit is executed)
3. M8145: Y000 pulse output stop (immediate stop)
4. M8146: Y001 pulse output stop (immediate stop)
5. M8147: Y000 pulse output monitoring (BUSY/READY)
6. M8148: Y001 pulse output monitoring (BUSY/READY)

The servo drive can store position information even if a power failure occurs. This instruction does not need to be executed at each power-on. Only unidirectional movement (backward direction) is allowed during the instruction execution. Therefore, the origin regression action must be executed prior to the DOG signal.

Note:

Positioning instructions (ZRN/PLSV/DRVI/DRVA) can be used multiple times in the program, but do not operate the same high-speed Y output port at the same time;

After the commanded drive power flow is OFF, when the drive is turned ON again, the status bit (Y000: [M8147], Y001; [M8148]) must be turned off and after an operation cycle, it can be driven again.

When the positioning instruction is re-driven, there must be an OFF time of more than one cycle. If the re-drive is executed in a shorter time than the above conditions, an "operation error" will occur during the first command execution (operation).

16-bit	32-bit	P	PLSV	Variable speed pulse output
✓	✓			
7 steps	13 steps		Instruction format: PLSV S D1 D2	

V	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D1		✓													
D2		✓	✓	✓											

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.

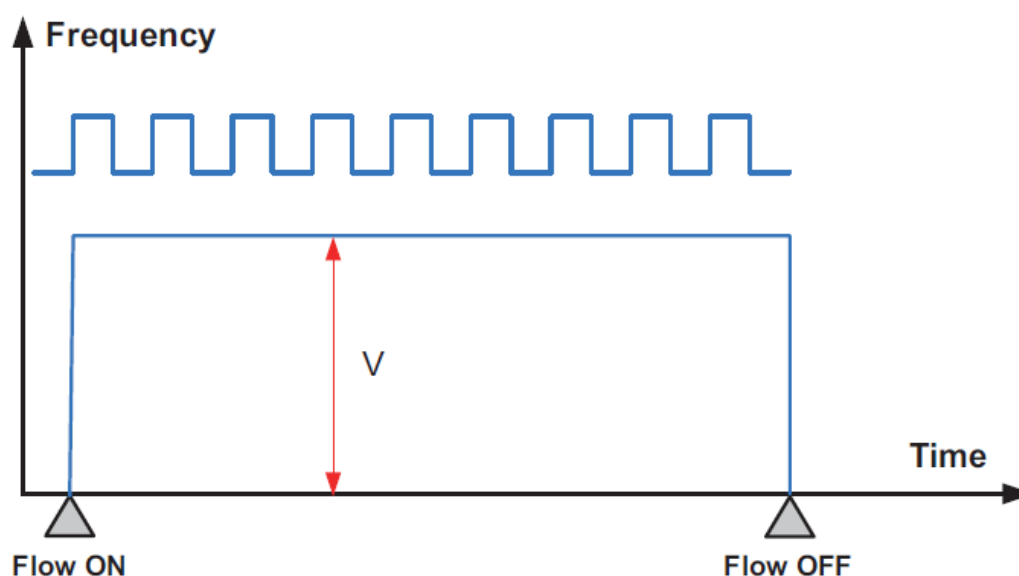
S indicates the specified pulse output frequency. For a 16-bit instruction, the range is 1 to 32,767 Hz and -1 to -32,768 Hz. For a 32-bit instruction, the range is 1 to 200,000 Hz and -1 to -200,000 Hz. The "-" indicates an instruction signal running in the reverse direction.

D1 indicates the pulse output port which can be Y0, Y1 or Y2

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.



Example:



When M1 is ON, Port Y1 outputs pulses at a frequency of 10 kHz and Y4 is used to control the operation direction. Y4=ON indicates forward direction.

System variables involved in the execution of this instruction are as follows:

D8141 (higher byte), D8140 (lower byte): number of pulses output by Y000. The current pulse position decreases when the instruction is executed in the reverse direction. (The 32-bit instruction is used.)

D8143 (higher byte), D8142 (lower byte): number of pulses output by Y001. The current pulse position decreases when the instruction is executed in the reverse direction. (The 32-bit instruction is used.)

D8151 (higher byte), D8150 (lower byte): number of pulses output by Y002. The current pulse position decreases when the instruction is executed in the reverse direction. (The 32-bit instruction is used.)

D8153 (higher byte), D8152 (lower byte): number of pulses output by Y003. The current pulse position decreases when the instruction is executed in the reverse direction. (The 32-bit instruction is used.)

D8155 (higher byte), D8154 (lower byte): number of pulses output by Y004. The current pulse position decreases when the instruction is executed in the reverse direction. (The 32-bit instruction is used.)

M8145: Y000 pulse output stop (immediate output) 7. M8146: Y001 pulse output stop (immediate output)

M8152: Y002 pulse output stop (immediate output) 9. M8153: Y003 pulse output stop (immediate output)

M8154: Y004 pulse output stop (immediate output)11. M8147: Y000 pulse output monitoring

(BUSY/READY)

M8148: Y001 pulse output monitoring (BUSY/READY)13. M8149: Y002 pulse output monitoring

(BUSY/READY)

M8150: Y003 pulse output monitoring (BUSY/READY)15. M8151: Y004 pulse output monitoring

(BUSY/READY)

16-bit	32-bit	P	DRVI	Relative positionig
✓	✓			
9 steps	17 steps		Instruction format : DRVI S1 S2 D1 D2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D1		✓														
D2		✓	✓	✓												

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, where:

S1 indicates the specified output pulse quantity. When the 16-bit instruction is executed, the value ranges from -32,768 to 32,767; when the 32-bit instruction is executed, the value ranges from -2,147,483,648 to 2,147,483,647. The minus sign indicates the reverse direction.

S2 indicates the specified output pulse frequency. When the 16-bit instruction is executed, the value ranges from 10 Hz to 32,767 Hz; when the 32-bit instruction is executed, the value ranges from 10 Hz to 100,000 Hz.

D1 indicates the pulse output port. For 3624MT/2416MT model, only Y0 or Y1 can be specified; for other MT models, Y0, Y1, or Y2 can be specified; for MTQ models, Y0, Y1, Y2, Y3, or Y4 can be specified.

D2 indicates the operation direction output port or bit variable. It indicates forward operation when the output is ON and indicates reverse operation when the output is OFF.

The output pulse quantity is based on the relative position of the following current value registers:

When pulses are output to [Y000], the current register is [D8I41 (high byte), D8I40 (low byte)] (32 bit).

When pulses are output to [Y001], the current register is [D8I43 (high byte), D8I42 (low byte)] (32 bit).

When pulses are output to [Y002], the current register is [D8I51 (high byte), D8I50 (low byte)] (32 bit).

When pulses are output to [Y003], the current register is [D8I53 (high byte), D8I52 (low byte)] (32 bit).

When pulses are output to [Y004], the current register is [D8I55 (high byte), D8I54 (low byte)] (32 bit).

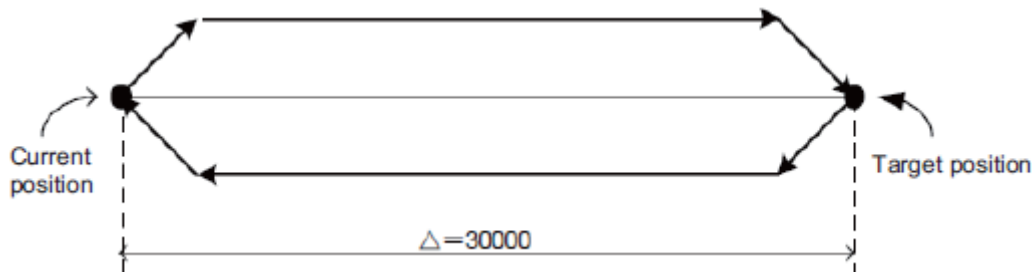
The value of the current value register decreases when the instruction is executed in reverse direction.

If the operand content is changed during instruction execution, the changed operand content cannot be reflected in the current operation. It takes effect in the next execution of the instruction. If the instruction-driven contact turns OFF during the instruction execution, the output port decelerates and stops pulse output. In this case, the execution completion flag M8029 does not act. After the instruction-driven contact turns OFF, when the pulse output interrupt flags M8147 (Y000), M8148 (Y001), M8149 (Y002), M8150 (Y003), M8151 (Y004) turn ON, the contact will not be driven by the instruction.

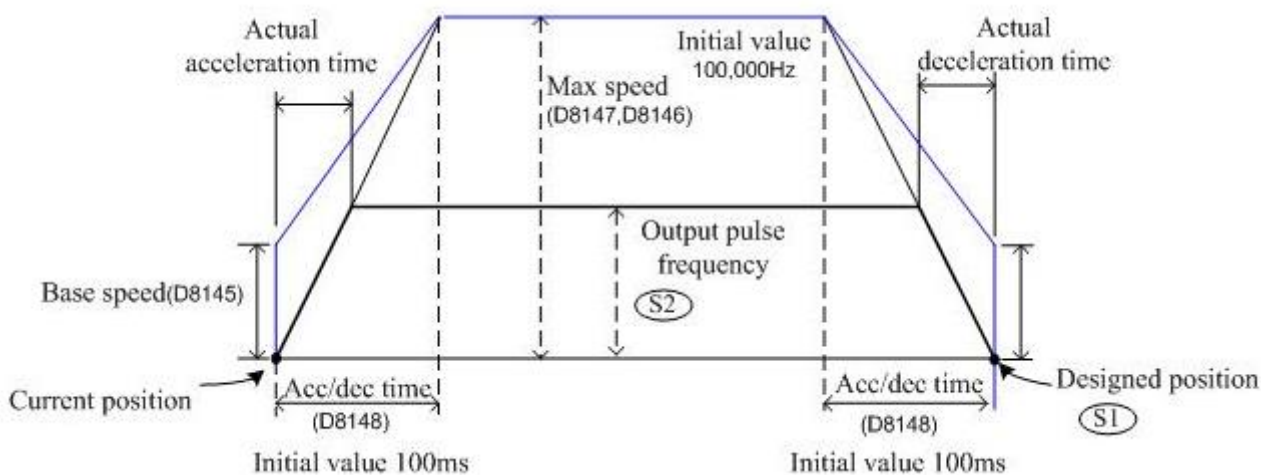
Example:



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.



The frequency accelerates or decelerates based on the preset value during pulse output.



The actual minimum frequency for pulse output is determined based on the following formula:

$$\text{Minimum frequency for pulse output} = \sqrt{\text{Max speed}[\text{D8147, D8146}]\text{HZ} \div (2 \times \text{Acc/dec time}[\text{D8148}]\text{ms} \div 1000)}$$

Even a value lower than the calculated result above is specified, the calculated value is still used for frequency output. The frequency at the initial phase of acceleration and at the last phase of deceleration cannot be lower than the calculated result above.

System variables involved in the execution of this instruction are as follows:

[D8145]: base speed during the execution of the (DRVI) and (DRVA) instructions. When the stepper motor is controlled, the resonance region and automatic startup frequency of the stepper motor need to be considered in the speed setting. Value range: lower than 1/10 of the maximum speed (D8147, D8146). When the value is out of this range, the base speed automatically decreases to the 1/10 of the maximum speed.

[D8147 (higher byte), D8146 (lower byte)]: maximum speed during the execution of the (DRVI) and (DRVA) instructions. The specified output pulse frequency must be smaller than the maximum speed. Value range: 10 to 100,000 (Hz)

[D8148]: acceleration/deceleration time during the execution of the (DRVI) and (DRVA) instructions. The acceleration/deceleration time refers to the time required for reaching the maximum speed (D8147, D8146).

Therefore, when the output pulse frequency S2 is lower than the maximum speed (D8147, D8146), the actual acceleration/deceleration time will be shortened. Value range: 50 to 5,000 (ms)

[M8145]: Y000 pulse output stop (immediate stop)

[M8146]: Y001 pulse output stop (immediate stop)

[M8152]: Y002 pulse output stop (immediate stop)

[M8153]: Y003 pulse output stop (immediate stop)

[M8154]: Y004 pulse output stop (immediate stop)

[M8147]: Y000 pulse output monitoring (BUSY/READY)

[M8148]: Y001 pulse output monitoring (BUSY/READY)

[M8149]: Y002 pulse output monitoring (BUSY/READY)

[M8150]: Y003 pulse output monitoring (BUSY/READY)

[M8151]: Y004 pulse output monitoring (BUSY/READY)

Keynote:

Positioning instructions (ZRN/PLSV/DRVI/DRVA) can be used multiple times in the program, but do not perform output operations on the same output port at the same time;

After the commanded drive power flow is OFF, when the drive is turned ON again, the status bit (Y000: [M8147], Y001: [M8148], Y002: [M8149], Y003: [M8150], Y004: [M8151]) must be OFF. After that, it can be driven again after one calculation cycle.

When the positioning instruction is re-driven, there must be an OFF time of more than one cycle. If the re-drive is executed in a shorter time than the above conditions, an "operation error" will occur during the first command execution (operation).

Note:

In the new version of H1U/H2U series PLC, the functions of PLSR, DRVI, DRVA and other instructions are enhanced;

- 1) By using special bits M8135~M8137 (respectively corresponding to Y0~Y2) to be ON, the number of output pulses (large or small) can be changed during operation; at the same time, the acceleration time and deceleration time are respectively defined by the following registers:

Port	Special bit	Register used for modifying acceleration time	Register used for modifying deceleration time
Y000	M8135	D8104	D8165
Y001	M8136	D8105	D8166
Y002	M8137	D8106	D8167

- 2) By using special bits M8085~M8087 (corresponding to Y0~Y2 respectively) to ON, the following functions can be realized:

When the drive special bit is ON, the next pulse output command can be started immediately, without the need to disable the previous power flow;

- 3) By using special bits M8090~M8092 (respectively corresponding to Y0~Y2) to be ON, the following functions



can be realized:

Can realize the interruption of pulse output completion; the specific correspondence is as follows:

Port	Special bit	Corresponding user interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

For details, please refer to the description in Appendix 8.6.

16-bit	32-bit	P	DRVA	Absolute position control
✓	✓			
9 steps	17 steps		Instruction format : DRVA S1 S2 D1 D2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D1		✓														
D2		✓	✓	✓												

This instruction is used to output pulses through a specified output, at a specified frequency, in a specified operation direction, to enable the servo actuator to move to the specified destination. This instruction is applicable only to PLCs using output ports of the transistor type, where:

S1 indicates the specified target position (absolute position). When the 16-bit instruction is executed, the value ranges from -32,768 to 32,767; when the 32-bit instruction is executed, the value ranges from -2,147,483,648 to 2,147,483,647.

If D1 is [Y000], the [D8141 (higher byte), D8140 (lower byte)] (32-bit instruction) indicates absolute position.

If D1 is [Y001], the [D8143 (higher byte), D8142 (lower byte)] (32-bit instruction) indicates absolute position.

If D1 is [Y002], the [D8151 (higher byte), D8150 (lower byte)] (32-bit instruction) indicates absolute position.

If D1 is [Y003], the [D8153 (higher byte), D8152 (lower byte)] (32-bit instruction) indicates absolute position.

If D1 is [Y004], the [D8155 (higher byte), D8154 (low byte)] (32-bit instruction) indicates absolute position.

The minus sign indicates the reverse direction. The value of the current value register decreases when the instruction is executed in reverse direction.

S2 indicates the specified output pulse frequency. When the 16-bit instruction is executed, the value ranges from 10 Hz to 32,767 Hz; when the 32-bit instruction is executed, the value ranges from 10 Hz to 100,000 Hz.

D1 indicates the pulse output port. For 3624MT/2416MT model, only Y0 or Y1 can be specified; for other MT models, Y0, Y1, or Y2 can be specified; for MTQ models, Y0, Y1, Y2, Y3, or Y4 can be specified.

S2 indicates the operation direction output port or bit variable. It is determined based on the difference between S1 and the current position. It indicates forward operation when the output is ON, and indicates reverse operation when the output is OFF.

If the operand content is changed during instruction execution, the changed operand content cannot be reflected in the current operation. It takes effect in the next execution of the instruction.

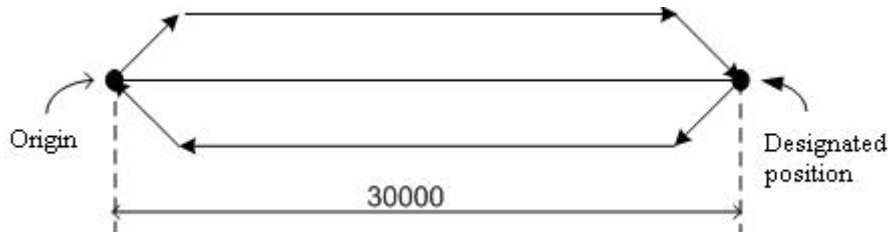
If the instruction-driven contact turns OFF during the instruction execution, the output port decelerates and stops pulse output. In this case, the execution completion flag M8029 does not act.

After the instruction-driven contact turns OFF, when the pulse output interrupt flags M8147 (Y000) and M8148 (Y001) turn ON, the contact will not be driven by the instruction.

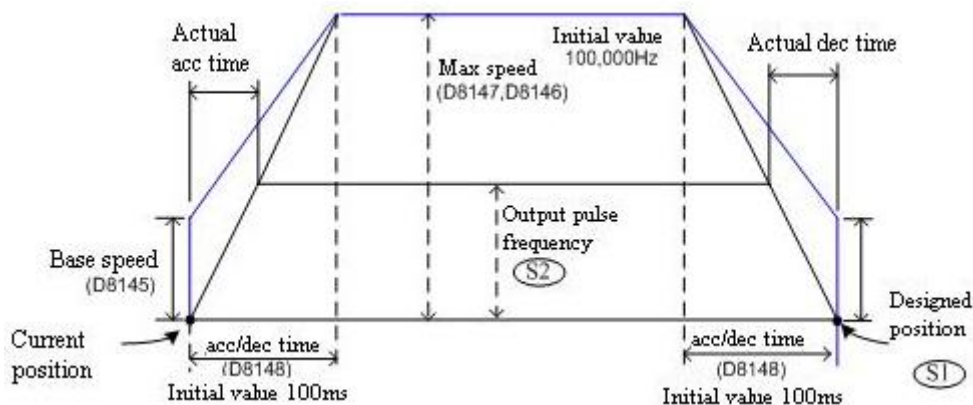
Example:



This instruction is used to control the actuator to move from the specified origin to the destination.



The frequency accelerates or decelerates based on the preset value during pulse output.



The actual minimum frequency for pulse output is determined based on the following formula:

$$\text{Minimum frequency for pulse output} = \sqrt{\text{Max speed}[\text{D8147, D8146}]\text{HZ} \div (2 \times \text{Acc/dec time}[\text{D8148}]\text{ms} \div 1000)}$$

Even a value lower than the calculated result above is specified, the calculated value is still used for frequency output. The frequency at the initial phase of acceleration and at the last phase of deceleration cannot be lower than the calculated result above.

System variables involved in the execution of this instruction are as follows:

[D8145]: base speed during the execution of the (DRVI) and (DRVA) instructions. When the stepper motor is controlled, the resonance region and automatic start frequency of the stepper motor need to be considered in the speed setting. Value range: lower than 1/10 of the maximum speed (D8147, D8146). When the value is out of this range, the base speed automatically decreases to the 1/10 of the maximum speed.

[D8147 (high byte), D8146 (low byte)]: maximum speed during the execution of the (DRVI) and (DRVA) instructions. The specified output pulse frequency must be smaller than the maximum speed. Value range: 10 to 100,000 (Hz)

[D8148]: acceleration/deceleration time during the execution of the (DRVI) and (DRVA) instructions. The acceleration/deceleration time refers to the time required for reaching the maximum speed (D8147, D8146).

Therefore, when the output pulse frequency is lower than the maximum speed (D8147, D8146), the actual acceleration/deceleration time will be shortened. Value range: 50 to 5,000 (ms)

[M8145]: Y000 pulse output stop (immediate stop)

[M8146]: Y001 pulse output stop (immediate stop)

[M8152]: Y002 pulse output stop (immediate stop)

[M8153]: Y003 pulse output stop (immediate stop)

[M8154]: Y004 pulse output stop (immediate stop)

[M8147]: Y000 pulse output monitoring (BUSY/READY)

[M8148]: Y001 pulse output monitoring (BUSY/READY)

[M8149]: Y002 pulse output monitoring (BUSY/READY)

[M8150]: Y003 pulse output monitoring (BUSY/READY)

[M8151]: Y004 pulse output monitoring (BUSY/READY)

Keynote:

Positioning instructions (ZRN/PLSV/DRVI/DRVA) can be used multiple times in the program, but do not perform output operations on the same output port;

After the commanded drive power flow is OFF, when the drive is turned ON again, the status bit (Y 000: [M8147], Y001: [M8148], Y002: [M8149]) must be turned off, and after an operation cycle, the drive will be turned on again.

When the positioning instruction is re-driven, there must be an OFF time of more than one cycle. If the re-drive is executed in a shorter time than the above conditions, an "operation error" will occur during the first command execution (operation).

Note:

In the new version of H1U/H2U series PLC, the functions of PLSR, DRVI, DRVA and other instructions are enhanced;

4) By using the special bits M8135~M8137 (respectively corresponding to Y0~Y2) to be ON, the number of output pulses (large or small) can be changed during operation; the acceleration time and the deceleration time is defined by the following registers:

Port	Special bit	Register used for modifying acceleration time	Register used for modifying deceleration time
Y000	M8135	D8104	D8165
Y001	M8136	D8105	D8166
Y002	M8137	D8106	D8167

5) By using special bits M8085~M8087 (corresponding to Y0~Y2 respectively) to be ON, the following functions can be realized:

When the drive special bit is ON, the next pulse output command can be started immediately, without the need to disable the previous power flow;

6) By using special bits M8090~M8092 (respectively corresponding to Y0~Y2) to be ON, the following functions can be realized:

Can realize the interruption of pulse output completion; the specific correspondence is as follows:

Port	Special bit	Corresponding user interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

For details, please refer to the description in Appendix 8.6.

## 4.3.12 Clock operation (160~169)

16-bit	32-bit	P	TCMP	Clock data comparison
✓		✓		
11 steps		11 steps	Instruction format : TCMP S1 S2 S3 S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
S3					✓	✓	✓	✓	✓	✓	✓	✓	✓			
S											✓	✓	✓			
D		✓	✓	✓												

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.

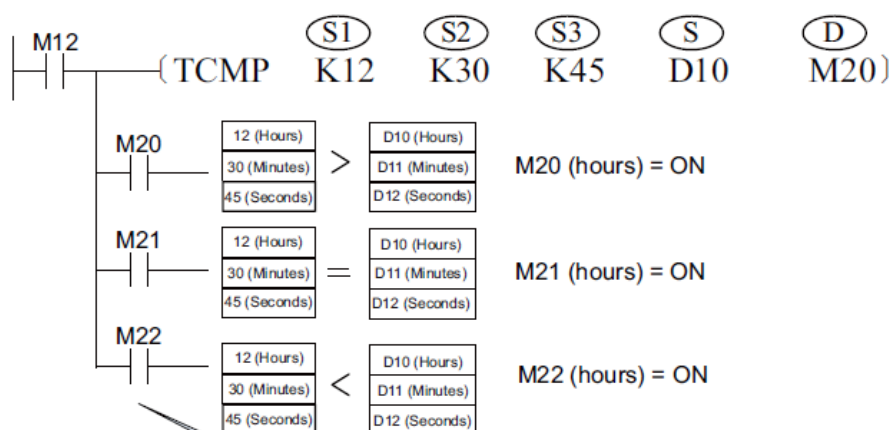
S2 represents minutes, in the range 0 to 59.

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  $\geq$ ,  $\leq$ , and  $\neq$  results.

16-bit	32-bit	P	TZCP	Clock data range comparison
✓		✓		
9 steps		9 steps	Instruction format : TZCP S1 S2 S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1											✓	✓	✓			
S2											✓	✓	✓			
S											✓	✓	✓			
D		✓	✓	✓												

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, where:

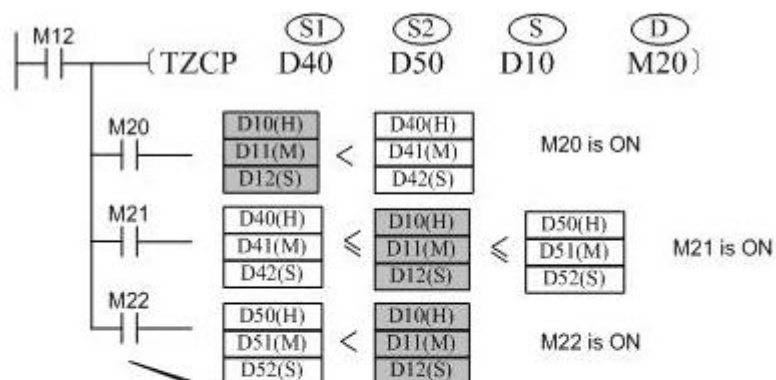
S1 is the set time comparison lower limit, which occupies 3 continuous variable units, and stores hour, minute, and second data in sequence:

S2 is the upper limit of the set time comparison, occupies 3 continuous variable units, and stores the hour, minute, and second data in sequence:

S is the start address of the time register of the real-time clock, usually the storage unit after the clock is read by the TRD or MOV instruction;

D is the starting address of the storage variable of the comparison result, occupying a total of 3 variable units in the following;

Example:



If M12=ON, one of M20~M22 will be turned ON.  
 When M12 is turned from ON to OFF, TZCP instruction will not be executed and the states in M20/M21/M22 before M12=OFF will be maintained.  
 RST or ZRST can be used to clear the comparison result of M20~M22.

16-bit	32-bit	P	TADD	Clock data addition operation
✓		✓		
7 steps		7 steps	Instruction format : TADD S1 S2 D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1											✓	✓	✓	✓	✓
S2											✓	✓	✓	✓	✓
D											✓	✓	✓	✓	✓

Two time values in **hours: minutes: seconds** format are added together. The result is stored in designated variables, where:

S1 is the time addend, which occupies 3 continuous variable units and stores hour, minute, and second data in sequence:

S2 is the time addend, which occupies 3 continuous variable units and stores hour, minute, and second data in sequence:

D is the sum of time, the storage unit, which occupies 3 continuous variable units, stores hour, minute, and second data in sequence:

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:



The following operation is completed:

(S1)		(S2)		(D)
D10(h) 09	+	D20(h) 08	=	D40(h) 18
D11(m) 50		D21(m) 56		D41(m) 46
D12(s) 16		D22(s) 09		D42(s) 25
9:50:16		8:56:09		18:46:25

If the result of addition operation is higher than 24 hours, the carry flag M8022 will be set to ON.

(S1)		(S2)		(D)
D10(h) 15	+	D20(h) 12	=	D40(h) 4
D11(m) 50		D21(m) 56		D41(m) 46
D12(s) 16		D22(s) 09		D42(s) 25
15:50:16		12:56:09		4:46:25



16-bit	32-bit	P	TSUB	Clock data subtraction operation
✓		✓		
7 steps		7 steps	Instruction format : TSUB S1 S2 D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1											✓	✓	✓	✓	✓
S2											✓	✓	✓	✓	✓
D											✓	✓	✓	✓	✓

One time value is subtracted from another in **hours: minutes: seconds** format. The result is stored in designated variables, where:

S1 is the time to be subtracted, occupying 3 continuous variable units, storing hour, minute, and second data in sequence:

S2 is the time minus number, which occupies 3 continuous variable units and stores hour, minute, and second data in sequence:

D is the time subtraction difference, the storage unit, which occupies 3 continuous variable units, stores hour, minute, and second data in sequence:

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:



The following operation is completed.

(S1)	(S2)	(D)
D10(H) 09	D20(H) 08	D40(H) 00
D11(M) 50	D21(M) 56	D41(M) 54
D12(S) 16	D22(S) 09	D42(S) 07
9:50:16	8:56:09	00:54:07

If the addition operation result is negative, then carry flag M8021 will be set to ON.

(S1)	(S2)	(D)
D10(H) 09	D20(H) 12	D40(H) 20
D11(M) 50	D21(M) 56	D41(M) 54
D12(S) 16	D22(S) 09	D42(S) 07
9:50:16	12:56:09	20:54:07

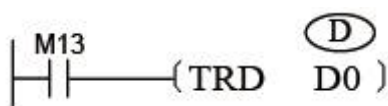
16-bit	32-bit	P	TRD	Clock data read
✓		✓		
3 steps		3 steps	Instruction format : TRD D	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D											✓	✓	✓	✓	✓

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:



The table lists the operations.

Item	System Variable		Data source D
Year (2000 to 2099)	D8018	→	D0
Month (1 to 12)	D8017	→	D1
Day (1 to 31)	D8016	→	D2
Hour (0 to 23)	D8015	→	D3
Minute (0 to 59)	D8014	→	D4
Second (0 to 59)	D8013	→	D5
Week [0 (day) to 6]	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.

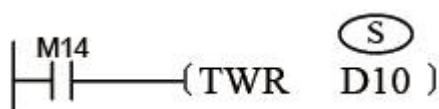
16-bit	32-bit	P	TWR	Clock data write
✓		✓		
3 steps		3 steps	Instruction format : TWR S	

Operand	Bit element				Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											✓	✓	✓	✓	✓

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:



The table below lists the operations.

Item	Data source D		System Variable
Year (2000 to 2099)	D0	→	D8018
Month (1 to 12)	D1	→	D8017
Day (1 to 31)	D2	→	D8016
Hour (0 to 23)	D3	→	D8015
Minute (0 to 59)	D4	→	D8014
Second (0 to 59)	D5	→	D8013
Week [0 (day) to 6]	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.

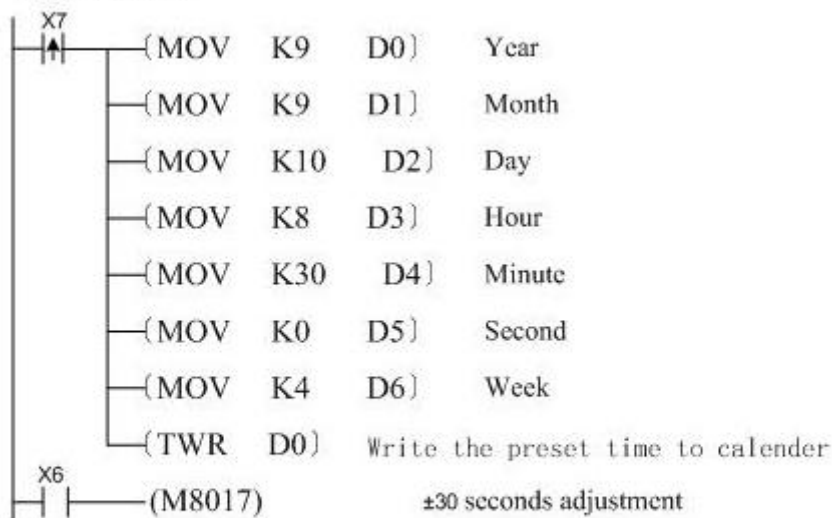
The clock is calibrated by  $\pm 30s$  each time M8017 is set to ON. When the number of seconds of the clock is within the range 1 to 29, the seconds are cleared while the minutes remain unchanged; when the number of seconds is within the range 30 to 59, the seconds are cleared while the number of minute is incremented by 1.

The clock stops when M8015 is set to ON.

The clock calibration method is described as follows.

## Example 2:

Change current time in PLC to Thursday, 0 second, 8:30,  
Sept. 10. 2009



Write the time to D0~D6 in ahead of a period of time. X7 will be turned ON to write the correct time to PLC when the actual time is due.

± 30 seconds adjustment can be made as soon as M8017 is turned ON.

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.

16-bit	32-bit	P	HOUR	Hour meter
✓	✓			
7 steps	13 steps		Instruction format : HOUR S D1 D2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D1													✓			
D2		✓	✓	✓												

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.

16-bit	32-bit	P	GRY	Grey code conversion (BIN→GRY)
✓	✓	✓		
7 steps	9 steps		Instruction format : GRY S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D								✓	✓	✓	✓	✓	✓	✓	✓	

Binary numbers are converted to grey 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.

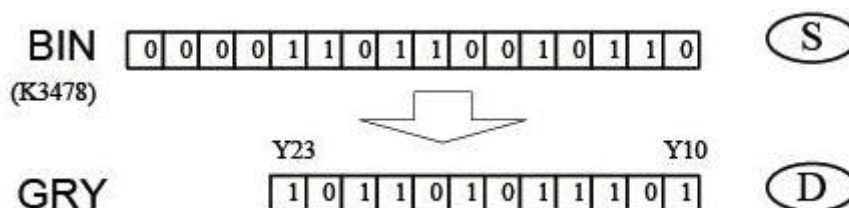
D is the unit that stores the grey code value after conversion.

Algorithm for BIN-to-grey code conversion: XOR operation is performed on every bit starting from the rightmost bit and a left-hand bit to get the grey code value. The leftmost bit remains unchanged (with a value of 0).

Example:



Execution result:



16-bit	32-bit	P	GBIN	Grey code inverse conversion (GRY→BIN)
✓	✓	✓		
7 steps	9 steps		Instruction format : GBIN S D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D								✓	✓	✓	✓	✓	✓	✓	✓	

Grey code values are converted to binary equivalents.

S is the grey 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.

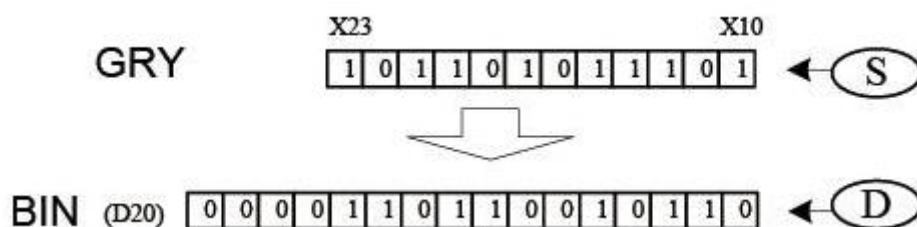
D is the unit that stores the BIN value after conversion.

Algorithm for grey 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:



16-bit	32-bit	P	RD3A	Analog module read
✓		✓		
7 steps		7 steps	Instruction format : RD3A m1 m2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
m1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
m2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D								✓	✓	✓	✓	✓	✓	✓	✓	

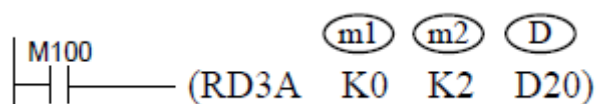
This instruction is used to read analog input values of Mitsubishi FX0N~3A modules., where;

m1 is the special module number; (K0~K7)

m2 is the analog input channel; (K1~K2)

D is the storage address of the value after reading

Example:



The analog input value of FX0N~3A can also be read with FROM instruction.



16-bit	32-bit	P	WR3A	Analog module write
✓		✓		
7 steps		7 steps	Instruction format : WR3A m1 m2 D	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
m1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
m2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
D								✓	✓	✓	✓	✓	✓	✓	✓	

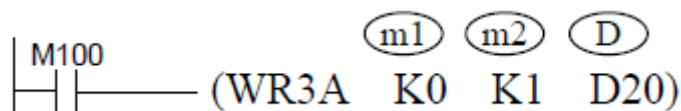
This instruction is used for writing analog output values of Mitsubishi FX0N~3A modules, where:

m1 is the special module number; (K0~K7)

m2 is the analog input channel; (K1)

D is the address of the parameter register in the main module, and its parameter is used as the source of write operation data.

Example:



The analog output value of FX0N~3A can also be written with TO instruction.

#### 4.3.14 Contact comparison (224~246)

Instruction	Instruction description
LD=	LD contact equal comparison
LD>	LD contact greater than comparison
LD<	LD contact smaller than comparison
LD< >	LD contact unequal comparison
LD>=	LD contact greater than or equal comparison
LD<=	LD contact smaller than or equal comparison
AND=	AND contact equal comparison
AND>	AND contact greater than comparison
AND<	AND contact smaller than comparison
AND< >	AND contact unequal comparison
AND>=	AND contact greater than or equal comparison
AND<=	AND contact smaller than or equal comparison
OR=	OR contact equal comparison
OR>	OR contact greater than comparison
OR<	OR contact smaller than comparison
OR< >	OR contact unequal comparison
OR>=	OR contact greater than or equal comparison
OR<=	OR contact smaller than or equal comparison

16-bit	32-bit	P	LD	Contact comparison
✓	✓			
5 steps	9 steps		Instruction format : LD S1 S2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The LD\* instruction compares the two operands and outputs the comparison result in a logical state. The variables in the comparison are processed as signed numbers. The \* comparison operator can be =, >, <, >=, <= or <>, where;

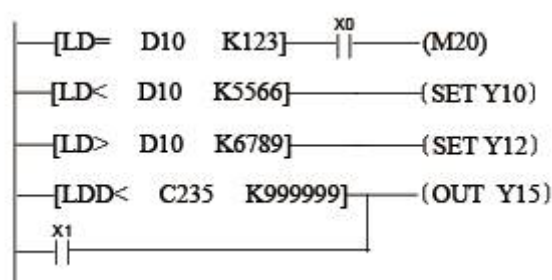
S1 is the data source or data variable unit 1 to be compared;

S2 is the data source or data variable unit 2 to be compared.

LD contact type comparison methods are:

16-bit Instruction	32-bit Instruction	Enabled state condition	Disabled state condition
LD=	LDD=	S1 = S2	S1 ≠ S2
LD>	LDD>	S1 > S2	S1 ≤ S2
LD<	LDD<	S1 < S2	S1 ≥ S2
LD<>	LDD<>	S1 <> S2	S1 = S2
LD≤	LDD≤	S1 ≤ S2	S1 > S2
LD≥	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 instructions LDD\* to compare 32-bit counters (C200~C255). If a different instruction is used, an error will occur.

16-bit	32-bit	P	AND*	Data comparison
✓	✓			
5 steps	9 steps		Instruction format : AND S1 S2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The AND\* instruction compares two operands and outputs the comparison result as a logical state. The variables in comparison are processed as signed numbers, where:

S1 is the data source or data variable unit 1 to be compared

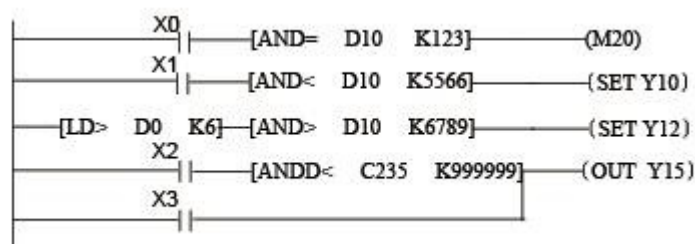
S2 is the data source or data variable unit 2 to be compared

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 <>.

AND contact type comparison methods are:

16-bit Instruction	32-bit Instruction	Enabled state condition	Disabled state condition
AND=	ANDD=	S1 = S2	S1 ≠ S2
AND>	ANDD>	S1 > S2	S1 ≤ S2
AND<	ANDD<	S1 < S2	S1 ≥ S2
AND<>	ANDD<>	S1 <> S2	S1 = S2
AND≤	ANDD≤	S1 ≤ S2	S1 > S2
AND≥	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.

16-bit	32-bit	P	OR*	Data comparison
✓	✓			
5 steps	9 steps		Instruction format : OR S1 S2	

Operand	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S1					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S2					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The OR\* instruction compares two operands and outputs the comparison result as a logical state. The variables in comparison are processed as signed numbers. The ☼ comparison operator can be =, >, <, >=, <=, or <>.

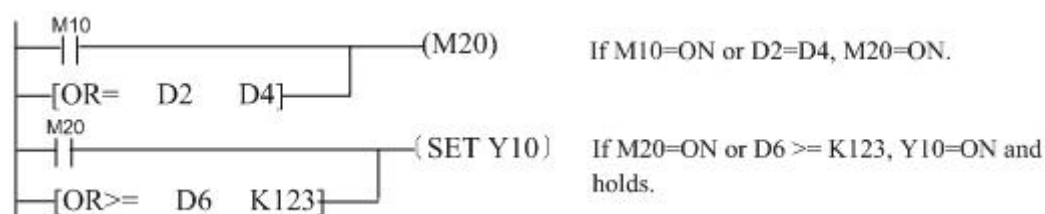
S1 is the data source or data variable unit 1 to be compared

S2 is the data source or data variable unit 2 to be compared

LD contact type comparison methods are:

16-bit Instruction	32-bit Instruction	Conduction Condition	Non-conduction Condition
OR=	ORD=	S1 = S2	S1 ≠ S2
OR>	ORD>	S1 > S2	S1 ≤ S2
OR<	ORD<	S1 < S2	S1 ≥ S2
OR<>	ORD<>	S1<>S2	S1 = S2
OR<=	ORD<=	S1 ≤ S2	S1 > S2
OR>=	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.



# Chapter 5 Communication Function

## Contents

5.1 Summary .....	- 275 -
5.1.1 Hardware and communication connection .....	- 275 -
5.1.2 H2U communication protocol switching logic description .....	- 276 -
5.2 HMI monitoring protocol .....	- 283 -
5.3 Parallel Protocol (1:1).....	- 284 -
5.4 N:N protocol.....	- 286 -
5.5 N:N Enhanced Protocol.....	- 290 -
5.6 Computer Link protocol .....	- 293 -
5.7 MODBUS protocol description .....	- 301 -
5.7.1 MODBUS master station communication application .....	- 302 -
5.7.2 MODBUS slave communication application.....	- 308 -
5.7.3 The programming method of using MODBUS protocol communication between multiple PLCs.....	- 316 -
5.7.4 MODBUS configuration and programming method.....	- 319 -
5.7.5 Inovance QLink protocol and programming method .....	- 323 -
5.7.6 Communication programming method between PLC and MD500 series inverter.....	- 326 -
5.7.7 Function code setting of MD500 with PLC application.....	- 327 -
5.7.8 PLC program to modify MD500 function code .....	- 334 -
5.9 CAN communication command description .....	- 337 -

---

5.9.1 Introduction to CANlink3.0 Application .....	- 337 -
5.9.2 Master-Slave network .....	- 337 -
5.9.3 Heartbeat mechanism.....	- 340 -
5.9.4 Offline detection.....	- 340 -
5.9.5 Use of CANlink3.0 .....	- 342 -
5.9.6 Software configuration CANlink3.0 master-slave communication configuration method .....	- 344 -
5.9.7 How to set CAN network configuration table.....	- 347 -
5.9.8 CANlink common faults and troubleshooting.....	- 364

-

## Chapter 5 Communication Function

### 5.1 Summary

The H1U/H2U series PLC main module contains four independent physical serial communication ports, namely COM0, COM1, COM2 and COM3. COM0 has programming and monitoring functions; COM1, COM2 and COM3 functions are customized, defined by users. The PLC comes standard with mini-USB communication port, which has the functions of uploading and downloading, monitoring and configuration.

The hardware standard configuration of the main module, the COM0 hardware is standard RS422, the interface terminal is mini DIN 8-pin female socket; the COM1 and COM2 hardware are standard RS485, and the interface is a wiring terminal; COM3 is reserved.

In addition to the standard communication hardware of the main module, users can also choose from three communication expansion cards. H2U-485-BD communication expansion card interface mode is terminal block, which can support 2-wire half-duplex RS485 standard communication and 4-wire full-duplex RS422 communication, requiring user wiring. The H2U-422-BD communication expansion card interface mode is mini DIN 8-pin, and a dedicated cable required. It only supports full-duplex RS422 communication. H2U-232-BD communication expansion card interface mode is DB9, supports 3-wire RS232 standard communication, standard RS232 communication can be used to connect user equipment.

#### 5.1.1 Hardware and communication connection

The hardware standard configuration of the whole machine: the COM0 hardware is standard RS422, and the interface terminal is an 8-hole mouse head female socket. The USB communication port is a MiniUSB interface.

Interface definition:



COM0 and mini-USB program download port

Pin number	Signal	Description
1	RXD-	Receive data negative
2	RXD+	Receive data positive
3	GND	Ground, no electrical connection
4	TXD-/RXD-	Externally data sends negative, 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.

Pin number	Signal	Description
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 connected

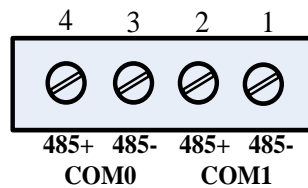
There are two ways to connect PLC and computer or touch screen through COM0.

Method 1: RS422 on the PLC side and USB on the computer side. The computer is connected to the program download port of COM0 through a dedicated USB download cable.

Method 2: The PLC side is RS422, and the computer side is RS232. The computer is connected to the program download port of COM0 through a dedicated serial port download cable.

H2U-XP version 24136 and above, H1U-XP version 26127 and above, support USB download. Please refer to the "usb" folder in the PLC software installation directory for the USB driver installation program.

The COM1 and COM2 hardware is RS485, the interface is the terminal block, the interface diagram:



Communication terminals

COM1 communicates and connects with other devices. Through the terminal block, users can wire on site. If the two serial ports communicate via RS485, both only support half-duplex communication mode. COM3 is reserved.

### 5.1.2 H2U communication protocol switching logic description

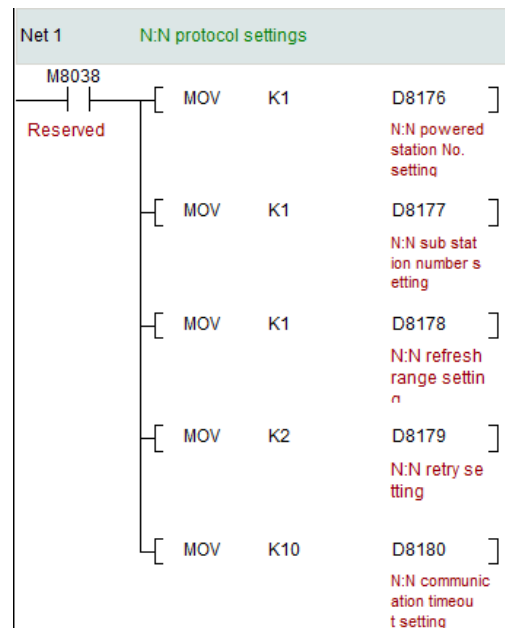
#### 1) COM0 protocol

COM0 protocol is fixed as download protocol/HMI monitoring protocol.

#### 2) COM1 protocol switching logic

- a) The protocol can be switched at any time during the shutdown state or the first operation. The system checks the protocol according to the priority. If there is a protocol with a higher priority, it will no longer check the protocol with a lower priority. The protocol priority is as follows: N:N, parallel master station, parallel slave station, computer link.
- b) N:N protocol trigger mode: starting from step 0, there are the following commands, and the system is set to N:N protocol.





N:N protocol configuration

- c) For the parallel protocol, M8070 is set as the master and M8071 is set as the slave. M8070 and M8071 cannot be set at the same time, and the parallel protocol is invalid if set at the same time.
- d) Computer link protocol, bit14 of D8120 = 1, the system sets COM1 protocol as computer link protocol.
- e) The protocol is determined by D8126. The value determined by D8126 during the first scan cycle of the PLC is valid for the protocol. After running, the change of D8126 cannot change the protocol. The corresponding relationship between D8126 and the protocol is shown in the protocol setting table.
- f) After the PLC is running, the protocol cannot be changed.

Remarks: The protocol switching logic of COM2 and COM3 is similar to COM1, only the D component is different, and the settings are described below.

### 3) Serial communication format setting

#### a) Protocol setting table

COM0 protocol	D8116 setup	Semi/Full duplex mode	COM0 Communication
Download protocol/HMI monitor protocol	01h	Full duplex	Fixed
Parallel link protocol master station	Not supported	Not supported	Not supported
Parallel link protocol slave station	Not supported	Not supported	Not supported
N:N Protocol Master	Not supported	Not supported	Not supported
N:N protocol slave station	Not supported	Not supported	Not supported
Computer link protocol	Not supported	Not supported	Not supported
MODBUS-RTU slave/Qlink protocol	02h	Half-duplex	Determined by D8110
MODBUS-ASC slave station	Not supported	Not supported	Not supported
RS instruction	Not supported	Not supported	Not supported
MODBUS RTU instruction	Not supported	Not supported	Not supported
MODBUS-ASC instruction	Not supported	Not supported	Not supported

COM1 communication protocol is set by D8126, communication mode and format are set by D8120.

COM2 communication protocol is set by D8266, communication mode and format are set by D8260.

COM3 communication protocol is set by D8276, communication mode and format are set by D8270

The protocol setting table of COM1 is as follows: (D component is different for COM2 and COM3, others are the same)

COM1 protocol	D8126 setup	Semi/Full duplex mode	COM1 Comm.
RS instruction	00h	Determined by D8120*	Determined by D8120*
HMI monitor protocol	01h	Half-duplex	Fixed
Parallel link protocol master station	50h	Half-duplex	Fixed
Parallel link protocol slave station	05h	Half-duplex	Fixed
N:N Protocol Master	40h	Half-duplex	Fixed
N:N Protocol Slave	04h	Half-duplex	Fixed
Computer link protocol	06h	Half-duplex	Determined by D8120
MODBUS-RTY slave station	02h	Half-duplex	Determined by D8120
MODBUS-ASC slave station	03h	Half-duplex	Determined by D8120
RS instruction	10h	Determined by D8120*	Determined by D8120
MODBUS RTU instruction	20h	Half-duplex	Determined by D8120
MODBUS-ASC instruction	30h	Half-duplex	Determined by D8120

\*RS command half-duplex/full-duplex mode is set by Bit10 of D8120;

1: Half-duplex RS485;

0: Full duplex RS232C/RS422;

(The standard configuration of COM1 is RS485. If RS485 is used, Bit10 of D8120 must be set to ON; if users need to use full-duplex RS422 or RS232C, they need to purchase a communication expansion board and set bit10 of D8120 to OFF).

Protocol and communication format comparison table:

Protocol Name	Baudrate	Data bit	Check bit	Stop bit
N:N Protocol	Default is 38400	7	Even parity check	1
N:N enhanced protocol	Default is 38400	7	Even parity check	1
Parallel Protocol	Default is 19200	7	Even parity check	1
HMI Monitoring Protocol	Set at 9600	7	Even parity check	1
Computer link protocol	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	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.	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)	Serial port 0 is checked through D8110, and serial port 1 is checked through bit 3 of D8120. 0: 1 bit 1: 2 bits
MODBUS-RT Slave				
MODBUS-AS Slave				
RS instruction				
MODBUS RTU instruction				
MODBUS-AS command				

## 4) List of serial communication format setting devices

COM0 serial port settings			
M8110	Reserved	D8110 *1	Communication format, the interface configuration settings, the default is 0
M8111	Waiting to send (RS instruction)	D8111 *1	Station number setting, the interface configuration settings, the default is 1
M8112	Send flag (RS instruction) instruction execution state (MODBUS)	D8112	The amount of remaining data transmitted (RS instruction only)
M8113	Receive complete flag (RS) Communication error flag (MODBUS)	D8113	The amount of received data (RS instruction only)
M8114	Receiving (only RS instruction)	D8114	Start character STX (only RS instruction)
M8115	Reserved	D8115	Termination character ETX (only RS instruction)
M8116	Reserved	D8116	Protocol settings, the interface configuration settings, the default is 0
M8117	Reserved	D8117	Computer link protocol of data starting address
M8118	Reserved	D8118	Computer link protocol sending data amount
M8119	Time-out judgement	D8119	Communication timeout judgment, interface configuration setting, the default is 10 (100ms)
COM1 serial port settings			
M8120	Reserved	D8120 *1	Communication format, the interface configuration with a default of 0
M8121	Waiting to send (RS instruction)	D8121 *1	Station number settings, the interface configuration settings with a default of 1
M8122	Send flag (RS instruction) instruction execution state (MODBUS)	D8122	Amount of remaining data to be transmitted (Only to RS instruction)
M8123	Receiving complete flag (RS) Communication error flag (MODBUS)	D8123	Amount of data already received (Only to RS instruction)
M8124	Receiving (only to RS instruction)	D8124	Start character STX (Only to RS instruction)
M8125	Reserved	D8125	End character ETX (Only to RS instruction)
M8126	485BD expansion card is valid when ON	D8126	Communication protocol, the interface configuration with a default of 0
M8127	Reserved	D8127	The computer link protocol connection requires the data start address number; SETEX protocol: The starting address of the buffer zone sent from the slave station (PLC) to the master station (display screen), the buffer size is 18.
M8128	Reserved	D8128	The number of data to be sent when the computer link protocol is connected; SETEX protocol: The master station (display screen) sends the starting address of the buffer zone to the slave station

			(PLC), the buffer size is 19.
M8129	Time-out judgement	D8129	Communication timeout judgment, interface configuration setting, the default is 10 (100ms)

COM2 serial port settings			
M8260	Reserved	D8260 *1	Communication format, the interface configuration with a default of 0
M8261	Sending and waiting (RS2 instruction)	D8261 *1	Station number settings, the interface configuration settings with a default of 1
M8262	Sending flag (RS2 instruction) Instruction execution status (MODBUS2)	D8262	Amount of remaining data to be transmitted (Only to RS2 instruction)
M8263	Receiving complete flag (RS2) Communication error flag (MODBUS2)	D8263	Amount of data already received (Only to RS2 instruction)
M8264	Receiving (only to RS2 instruction)	D8264	Start character STX (Only to RS2 instruction)
M8265	Reserved	D8265	End character ETX (Only to RS2 instruction)
M8266	485BD expansion card is valid when ON	D8266	Communication protocol, the interface configuration with a default of 0
M8267	Reserved	D8267	The computer link protocol connection requires the data start address number; SETEX protocol: The starting address of the buffer zone sent from the slave station (PLC) to the master station (display screen), the buffer size is 18.
M8268	Reserved	D8268	The number of data to be sent when the computer link protocol is connected; SETEX protocol: The master station (display screen) sends the starting address of the buffer zone to the slave station (PLC), the buffer size is 19.
M8269	Time-out judgement	D8269	Communication timeout judgment, interface configuration setting, the default is 10 (100ms)
COM3 serial port settings			
M8270	Reserved	D8270 *1	Communication format, the interface configuration with a default of 0
M8271	Sending and waiting (RS2 instruction)	D8271 *1	Station number settings, the interface configuration settings with a default of 1
M8272	Sending flag (RS2 instruction) Instruction execution status (MODBUS2)	D8272	Amount of remaining data to be transmitted (Only to RS2 instruction)
M8273	Receiving complete flag (RS2) Communication error flag (MODBUS2)	D8273	Amount of data already received (Only to RS2 instruction)
M8274	Receiving (only to RS instruction)	D8274	Start character STX (Only to RS2 instruction)
M8275	Reserved	D8275	End character ETX (Only to RS2 instruction)
M8276	485BD expansion card is	D8276	Communication protocol, the interface configuration with a

	valid when ON		default of 0
M8277	Reserved	D8277	The computer link protocol connection requires the data start address number; SETEX protocol: The starting address of the buffer zone sent from the slave station (PLC) to the master station (display screen), the buffer size is 18.

COM3 serial port settings			
M8278	Reserved	D8278	The number of data to be sent when the computer link protocol is connected; SETEX protocol: The master station (display screen) sends the starting address of the buffer zone to the slave station (PLC), the buffer size is 19.
M8279	Time-out judgement	D8279	Communication timeout judgment, interface configuration setting, the default is 10 (100ms)

\*1: Save when power off.

#### 5) List of communication error codes

Type	Error code	Description	Remedies
Parallel connection communication error M8063 (D8063) continues to run	0000	Normal	Check whether PCs are powered on, whether the adapter is correctly connected to the controller, and whether adapters are correctly connected to each other.
	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 to 6311	None	
	6312	Parallel character incorrect	
	6313	Parallel sum incorrect	
	6314	Parallel format incorrect	
	6330	Incorrect MODBUS slave station address	COM0 communication error: Check whether COM0 communication cables are correctly connected, whether communication formats are matched, whether communication protocols are matched, whether devices are powered on (COM0 can be used as a free port only when devices are powered on; otherwise, it can be only used for monitoring or download), and whether the JP0 jumper is inserted (COM0 can be used as an RS485 free port only when the JP0 jumper is disconnected; if the JP0
	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 port protocol error		

Type	Error code	Description	Remedies
			jumper is connected, it can be only used for monitoring or download in RS422 mode).
	6340	Incorrect MODBUS slave station address	COM1 communication error: Check whether COM1 communication cables are correctly connected and 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 port protocol error	
	6350	Incorrect MODBUS slave station address	COM2 communication error: Check whether COM2 communication cables are correctly connected and whether communication formats are matched
	6351	Incorrect data frame length	
	6352	Incorrect address	
	6353	CRC error	
	6354	Unsupported instruction code	
	6355	Receiving timeout	
	6356	Data error	
	6357	Buffer overflow	
	6358	Frame error	
	6359	Serial port protocol error	
	6360	Incorrect MODBUS slave station address	COM3 communication error: Check whether COM3 communication cables are correctly connected and whether communication formats are matched
	6361	Incorrect data frame length	
	6362	Incorrect address	
	6363	CRC error	
	6364	Unsupported instruction code	
	6365	Receiving timeout	
	6366	Data error	
	6367	Buffer overflow	
	6368	Frame error	
	6369	Serial port protocol error	

Note: M8063 and D8063 will remain ON after the fault has been corrected, until the user clears it forcibly.

## 5.2 HMI monitoring protocol

### 1) Hardware configuration and software setting:

COM0 communication connection method uses RS422, and it can only be connected through the download port.

COM1 communication method uses RS485, you need to set D8126=01h, use wired terminal block, or wired H2U-485-BD expansion card.

COM1 communication method uses RS422, D8126=81h needs to be set, use wired H2U-485-BD expansion card or connection through H2U-422-BD expansion card. H2U-485-BD expansion card needs to be configured as a 4-wire system. H2U-422-BD needs to be connected through a dedicated cable.

The communication format and baud rate of this protocol are fixed.

Remarks: The hardware configuration and software settings of COM2 and COM3 are similar to those of COM1, only the D component is different. COM2 needs to set D8266; COM3 needs to set D8276.

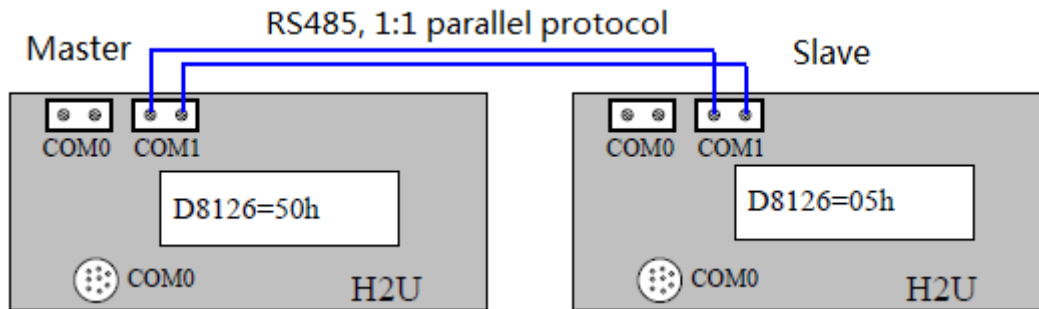
### 2) Program description:

The HMI monitoring protocol is the PLC internal protocol, used for AutoShop software to communicate with the PLC. AutoShop can erase, read and download user programs through this protocol. It can implement local, remote adjustment and remote control of PLC. Specifically, it can monitor the status of any element in the PLC, change any element forcibly, and control the start and stop of the PLC.

In order to protect the user program, it is recommended that other devices not use this protocol to connect to the PLC except for the debugging program during PLC work.

## 5.3 Parallel Protocol (1:1)

When it is necessary to exchange data between two H1U/H2U series PLC main modules, the 1:1 parallel communication protocol is one of the easiest ways to communicate. The hardware only needs to use a twisted pair to connect the COM1 ports of the two PLC's. The RS485 signal terminal can be connected in parallel to form a communication network.



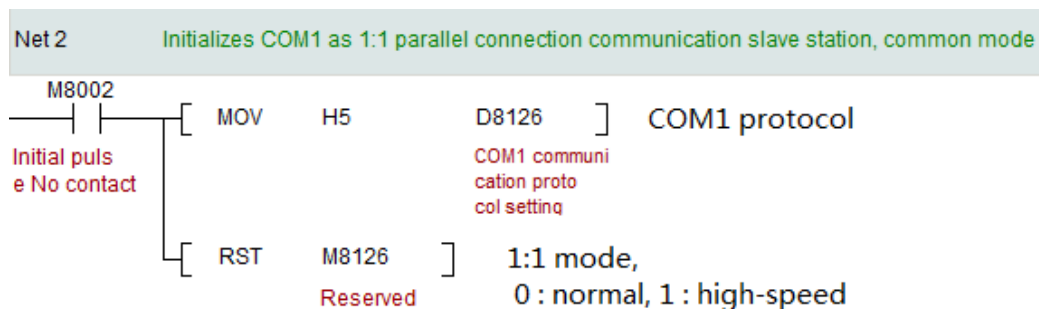
1) Hardware configuration and software setting:

Parallel protocol can only be used for COM1. There are two ways to set COM1 as parallel protocol:

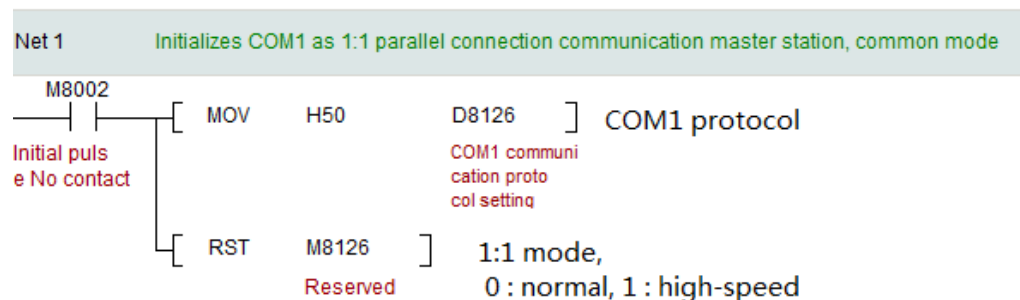
1. PLC is the master when D8126=50h and slave when D8126=05h
2. PLC is master when M8070=ON and slave when M8071=ON

The communication format and baud rate of this protocol are fixed.

The PLC system software has built-in parallel protocol. You only need to set it in D8126. Configure one PLC as the parallel protocol master station and set common communication mode using the following statements:



Configure another PLC as the parallel protocol slave station and set the common communication mode using the following statements:





Connect the COM1 serial ports of two PLC's to implement data exchange between the PLC's. The data area of communication exchange has fixed address to receive and send the fixed variable zones. Depending on the size of data to be exchange, there are two communication modes:

	Send by master station (Slave receives)	Send by slave station(Master receives)
Common Mode M8162=0	M800~M899 D490~D499	M900~M999 D500~D509
High-speed Mode M8162=1	D490~D491	D500~D501

Variables related to parallel protocol communication and control:

M8070: The value 1 indicates the parallel master station. If the value is 0 and D8126=50h, it is also the parallel master station. (The setting of M8070 is prior to the setting of D8126.)

M8071: The value 1 indicates the parallel slave station. If the value is 0 and D8126=05h, it is also the parallel slave station. (The setting of M8071 is prior to the setting of D8126.)

M8162: high-speed parallel connection

M8072: parallel connection running

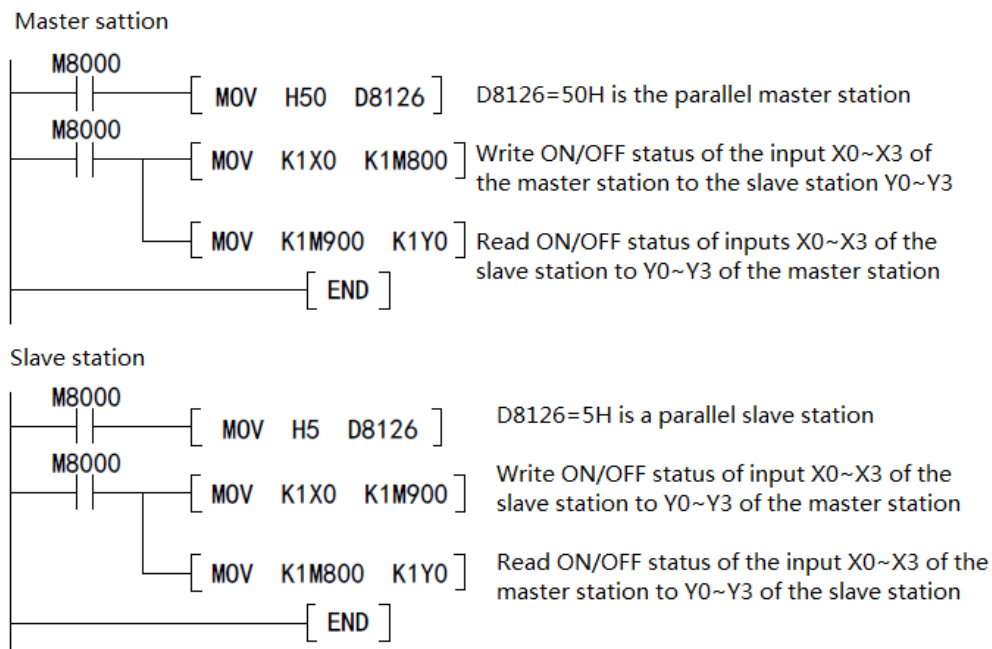
M8073: parallel connection setting abnormal

M8063: serial communication error

D8070: error time, 500 by default

D8063: serial communication error code

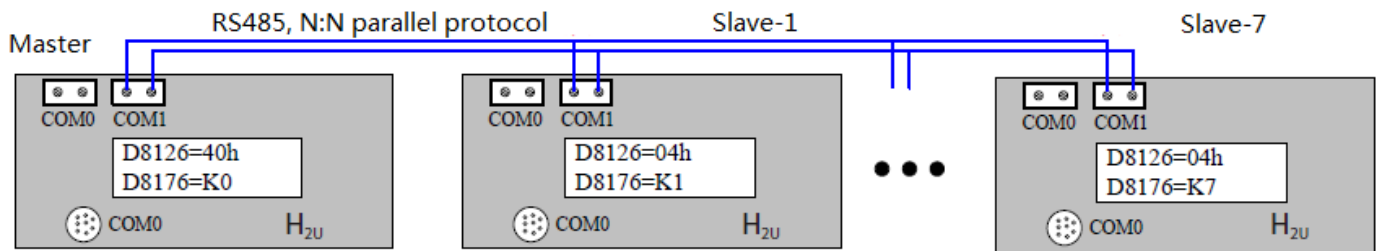
2) Parallel example reference:



Note: Another setting: M8070 is turned ON as the master station; M8071 is turned ON as the slave station. XP type COM2/COM3 can also use parallel protocol, the program instruction are same as COM1, please refer to the corresponding serial port settings for setting M and D.

## 5.4 N:N protocol

When there are multiple (2-8) PLC's in a system that need to exchange information with each other and work in co-ordination, the built-in N:N network protocol of PLC can be used to realize multi-party communication between PLC's. The hardware only needs to use twisted pair to connect all PLC COM1 port RS485 signal ends in parallel to form a communication network.



1) Hardware configuration and software setting:

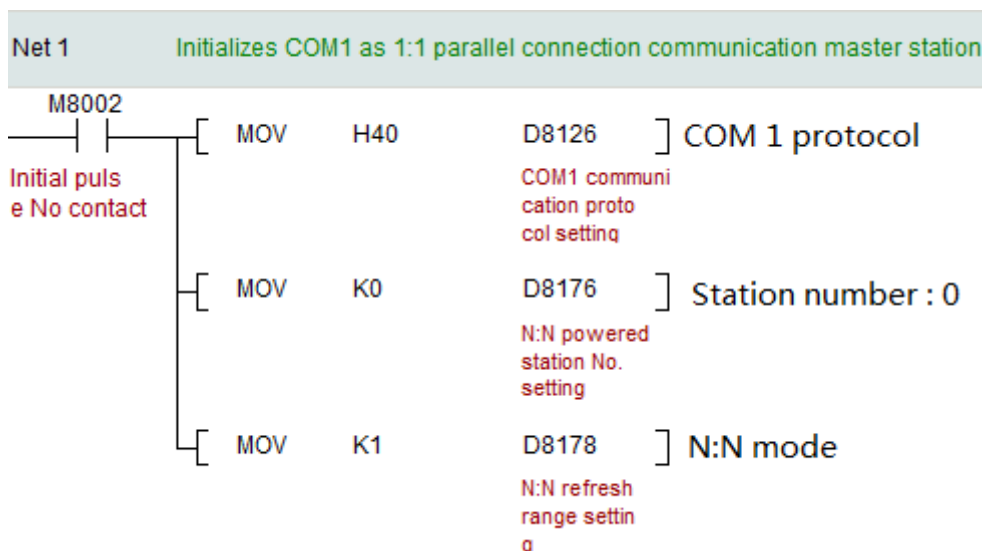
This protocol can only be used for COM1. There are two ways to set COM1 to N:N protocol:

1. Set D8126=40h, PLC is N:N protocol for master and set D8126=04h, PLC is N:N protocol slave station.

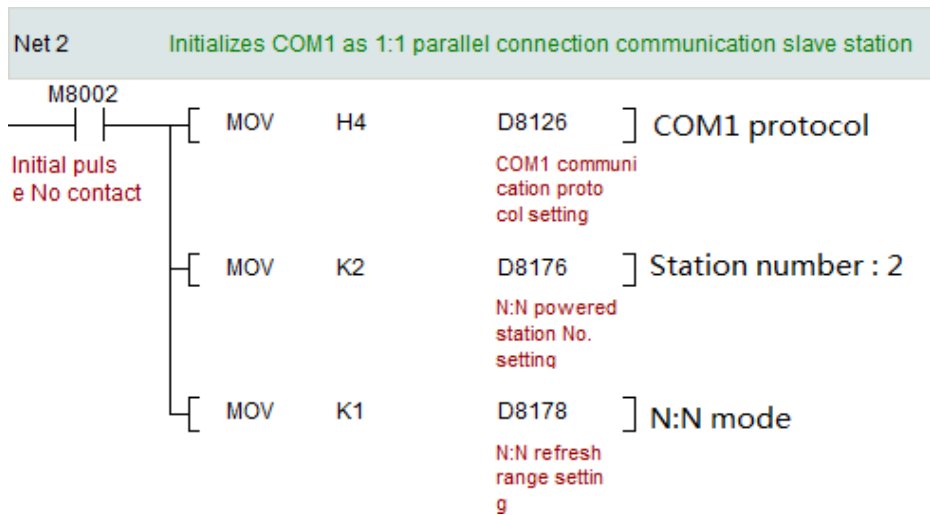
2. Add a program at the beginning of the program.

The communication format and baud rate of this protocol are fixed.

The user needs to set a PLC as N:N protocol master station, set the station number of the machine D8176 to 0, and the speed mode D8178, for example:



The other PLCs are set as N:N protocol slaves, and the station number of this machine is D8176, and the speed mode is D8178, for example:



When the PLC is running, data can be exchanged between multiple PLC's. The user program can read the sending status data of other PLC's in the specific data area of the PLC; the user program of this machine copies the data that needs to be broadcast to the specific data unit. It can be read by other PLC.

According to the two index requirements of the required communication interaction information volume and communication refresh speed, there are three modes to choose from, and the corresponding variable areas are defined as follows:

Mode	Station No.	Soft Component	
		Bit element (M)	Soft Word element (D)
		0	4
Mode 0 D8178=0 Exchange data 0 M component 4 D component	No. 0	None	D0 to D3
	No. 1	None	D10 to D13
	No. 2	None	D20 to D23
	No. 3	None	D20 to D23
	No. 4	None	D40 to D43
	No. 5	None	D50 to D53
	No. 6	None	D60 to D63
	No. 7	None	D70 to D73
Mode 1 D8178=1 Exchange data 32 M component 4 D component	No. 0	M1000 to M1031	D0 to D3
	No. 1	M1064 to M1095	D10 to D13
	No. 2	M1128 to M1159	D20 to D23
	No. 3	M1192 to M1223	D30 to D33
	No. 4	M1256 to M1287	D40 to D43
	No. 5	M1320 to M1351	D50 to D53
	No. 6	M1384 to M1415	D60 to D63
	No. 7	M1448 to M1479	D70 to D73
Mode 2 D8178=2 Exchange data 64 M component 8 D component	No. 0	M1000 to M1063	D0 to D7
	No. 1	M1064 to M1127	D10 to D17
	No. 2	M1128 to M1191	D20 to D27
	No. 3	M1192 to M1255	D30 to D37
	No. 4	M1256 to M1319	D40 to D47
	No. 5	M1320 to M1383	D50 to D57
	No. 6	M1384 to M1447	D60 to D67
	No. 7	M1448 to M1511	D70 to D77

Set the relevant register description of N:N connection protocol:

D8126: COM1 communication port and protocol configuration, set to 40h means N:N master station; set to 04h means N:N slave station.

D8176: station number, range 0~7, 0 means main station;

D8177: total number of slave stations, ranging from 1 to 7, only the master station needs to be set;

D8178: refresh range (mode) setting, range 0~2, only the master station needs to set;

D8179: retry times setting, only the master station needs to set;

D8180: communication timeout setting, the unit is 10ms, only the master station needs to set;

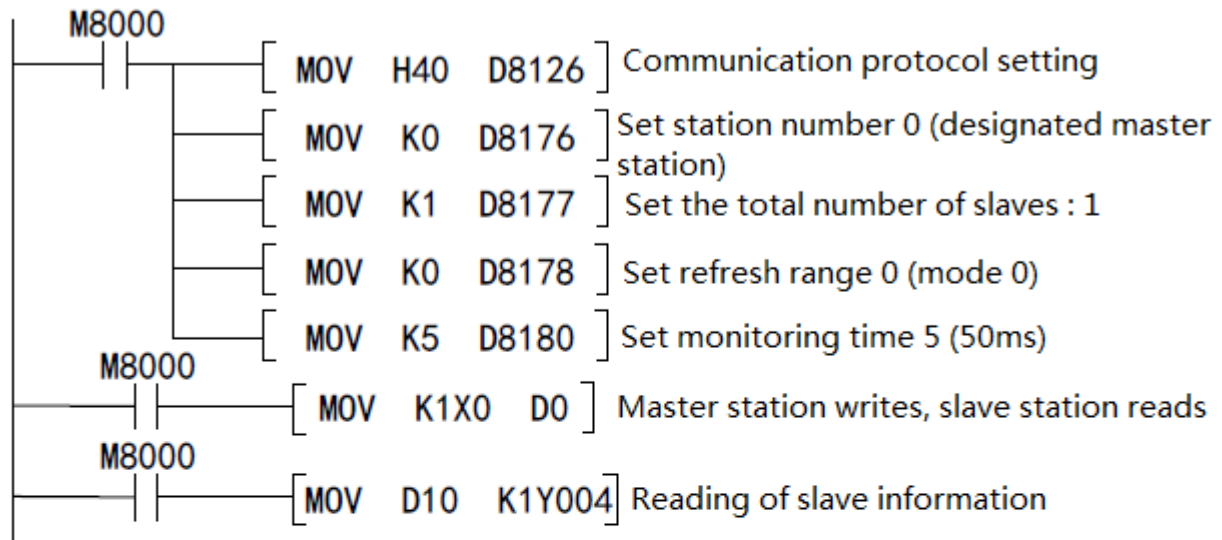
M8183~M8190: communication error flag, M8183 corresponds to station No. 0 (master station), M8184 corresponds to station No. 1, and so on, M8190 corresponds to station No. 7;

M8191: Data transmission is being performed.

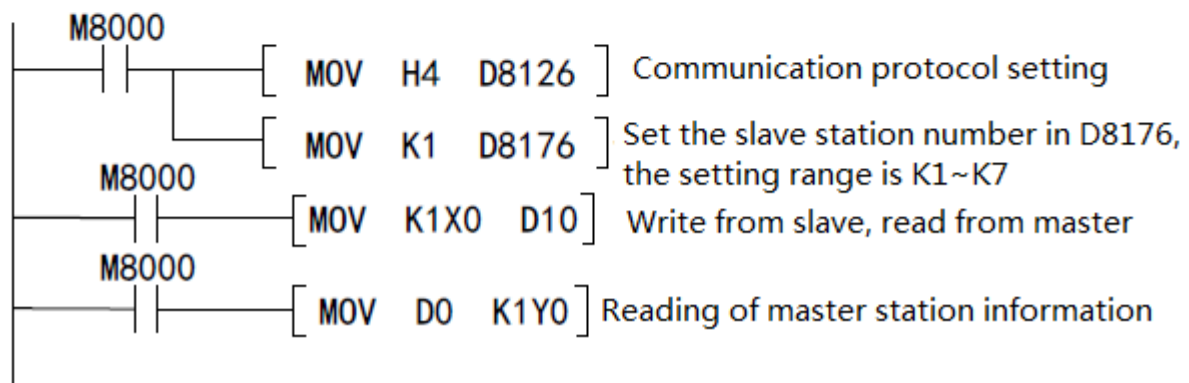
D8063	Serial communication error code (see the list of communication error codes)
D8070	Parallel connection error time, the default is 500ms
D8173	Station number setting status of this station
D8174	Communication slave station setting status
D8175	Refresh range setting status
D8201	Current connection scan time
D8202	Maximum connection time
D8203	Master station communication error times
D8204	Slave 1 communication error times
D8205	Slave 2 communication error times
D8206	Slave 3 communication error times
D8207	Slave 4 communication error times
D8208	Slave 5 communication error times
D8209	Slave 6 communication error times
D8210	Slave 7 communication error times
D8211	Master station communication error code
D8212	Slave 1 communication error code
D8213	Slave 2 communication error code
D8214	Slave 3 communication error code
D8215	Slave 4 communication error code
D8216	Slave 5 communication error code
D8217	Slave 6 communication error code
D8218	Slave 7 communication error code

## 2) N: N communication example reference:

## Master station program



## Slave station program



Special note: XP type COM2/COM3 can also use N:N protocol, the program instructions are similar to COM1, only need to replace the communication protocol register accordingly.

For example: To use COM2 port as N:N protocol master station, you need to set D8266=40h, and other registers do not need to be modified. It should be noted that, because of the shared registers, at most one COM port can be set to N:N or N:N enhanced protocol, otherwise it will cause unknown errors.

## 5.5 N:N Enhanced Protocol

The N:N enhanced protocol increases the number of PLC's that can be communicate in network compared to the N:N protocol from 8 to 16, and the amount of data that can be exchanged has also been expanded.

### 1) Communication principle:

Communication hardware setting: The baud rate is fixed at 38400bps, and the communication format is 7E1. When the PLC is running, data can be exchanged between multiple PLC's. The user program can read the sending status data of other PLC's in the specific data area of the PLC; the user program of this machine copies the data that needs to be broadcast to the specific data unit , It can be read by other PLC.

Host (address 0)	Data area 0: Host can write, other machines can read
Slave 1 data (address 1)	Data area 1: Slave 1 can write; other machines can read
Slave 2 data (address 2)	Data area 2: Slave 2 can write; other machines can read
Slave 3 data (address 3)	Data area 3: Slave 3 can write; other machines can read
Slave 4 data (address 4)	Data area 4: Slave 4 can write; other machines can read
Slave 5 data (address 5)	Data area 5: Slave 5 can write; other machines can read
Slave 6 data (address 6)	Data area 6: Slave 6 can write; other machines can read
Slave 7 data (address 7)	Data area 7: Slave 7 can write; other machines can read
Slave 8 data (address 8)	Data area 8: Slave 8 can write; other machines can read
Slave 9 data (address 9)	Data area 9: Slave 9 can write; other machines can read
Slave 10 data (address 10)	Data area 10: Slave 10 can write; other machines can read
Slave 11 data (address 11)	Data area 11: Slave 11 can write; other machines can read
Slave 12 data (address 12)	Data area 12: Slave 12 can write; other machines can read
Slave 13 data (address 13)	Data area 13: Slave 13 can write; other machines can read
Slave 14 data (address 14)	Data area 14: Slave 14 can write; other machines can read
Slave 15 data (address 15)	Data area 15: Slave 15 can write; other machines can read

Compared with the N:N protocol, the refresh mode of the N:N enhanced protocol is fixed and not selectable. The M element and D element planning for data interaction of each networked PLC is shown in the following table.

Station	Soft element	
	Bit address (M)	Word address (D)
	64	32
No. 0	M500 to M563	D200 to D231
No. 1	M570 to M633	D240 to D271
No. 2	M640 to M703	D280 to D311
No. 3	M710 to M773	D320 to D351
No. 4	M780 to M843	D360 to D391
No. 5	M850 to M913	D400 to D431
No. 6	M920 to M983	D440 to D471
No. 7	M990 to M1053	D480 to D511
No. 8	M1060 to M1123	D520 to D551

Station	Soft element	
	Bit address (M)	Word address (D)
	64	32
No. 9	M1130 to M1193	D560 to D591
No. 10	M1200 to M1263	D600 to D631
No. 11	M1270 to M1333	D640 to D671
No. 12	M1340 to M1403	D680 to D711
No. 13	M1410 to M1473	D720 to D751
No. 14	M1480 to M1543	D760 to D791
No. 15	M1550 to M1613	D800 to D831

2) Definition of related registers:

D8126: Set to 41h to indicate N:N enhanced protocol master station; set to 14h to indicate N:N enhanced protocol slave station.

D8176: station number, range 0-15, 0 means main station;

D8177: total number of slave stations, ranging from 1 to 15, only the master station needs to be set;

D8179: retry times setting, only the master station needs to set, the default is 3;

D8180: communication timeout setting, the unit is 10ms, only the master station needs to set, the default is 5;

M8183~M8190 / M8310~M8317: communication error flag, M8183 corresponds to station 0 (master station), M8184 corresponds to station 1, and so on, M8317 corresponds to station 15;

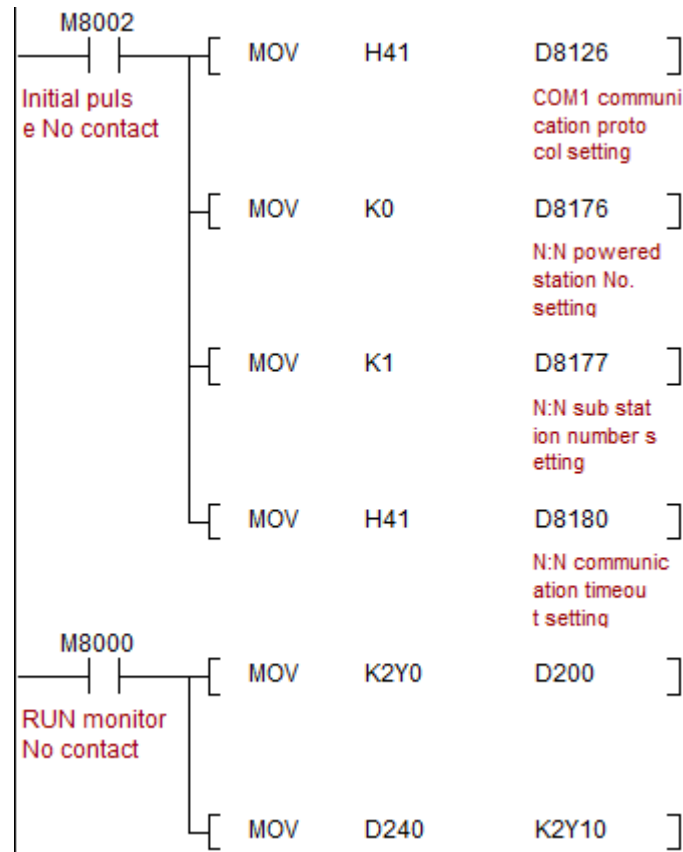
See the table below for other related registers

D8063	Serial communication error code (see the list of communication error codes)
D8070	Parallel connection error time, the default is 500ms
D8173	Station number setting status of this station
D8174	Communication slave station setting status
D8175	Refresh range setting status
D8201	Current connection scan time
D8202	Maximum connection time
D8203	Master station communication error times
D8204	Slave 1 communication error times
D8205	Slave 2 communication error times
D8206	Slave 3 communication error times
D8207	Slave 4 communication error times
D8208	Slave 5 communication error times
D8209	Slave 6 communication error times
D8210	Slave 7 communication error times
D8211	Master station communication error code
D8212	Slave 1 communication error code
D8213	Slave 2 communication error code
D8214	Slave 3 communication error code
D8215	Slave 4 communication error code
D8216	Slave 5 communication error code
D8217	Slave 6 communication error code
D8218	Slave 7 communication error code

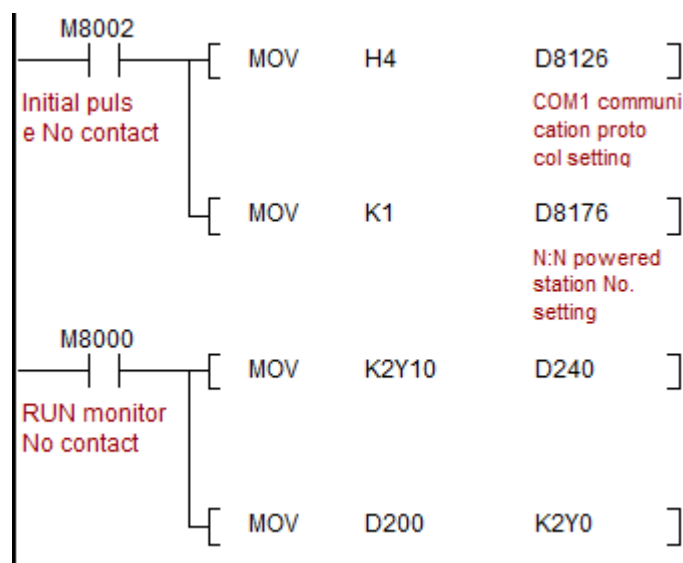
3) N:N enhanced protocol example:

Communication requirements: exchange the values of Y0-Y7 and Y10-Y17 of two PLC's through the N:N enhanced protocol. The ladder diagram can be written as follows.

Master:



Slave:





## 5.6 Computer Link protocol

Hardware configuration and software settings:

This protocol can only be used for COM1. There are two ways to set the COM1 computer link protocol:

1. Set D8126 = 06h, PLC is the computer link protocol.
2. Bit14 of D8120 = 1, the baud rate and communication format are determined by D8120.

Protocol description:

### 1) Data sequence

Format	Control Code	Station No.	PC No.	Command	Wait Time	Content Data	[Verification Sum]	[LF CR]
Bytes	1	2	2	2	1	X	2	2

#### a) Control Code

Signal	Code (hexadecimal)	Description
STX	02H	Start text
ETX	03H	End text
EOT	04H	End of Transmission
ENQ	05H	Enquiry
ACK	06H	Acknowledge
LF	0AH	Next line
CL	0CH	Clear
CR	0DH	Enter
NAK	15H	Not acknowledged

b) Station number: the address, used to distinguish the communication between this PLC and other PLC's, set by D8121, the value is 00~0FH.

c) PC number: for A series, fixed as "FF" for FX series.

d) Command:

			Command		Description	Maximum communication unit
			Symbol	ASCII code		
Soft component memory	Read	Bit unit	BR	42H,52H	One group of bit soft component (X, Y, M, S, T, C) is read, result uses 1 point as the unit.	256
		Word unit	WR	57H,52H	One group of bit soft component (X, Y, M, S) is read, result uses 16 points as the unit.	32 words 512 points
	One group of word soft component (D, T, C) is read, result uses 1 component as the unit.				64 points	
	Write	Bit unit	BW	42H,57H	One group of bit soft component (X, Y, M, S, T, C) is written, result uses 1 point as the unit.	160 points

			Command		Description	Maximum communication unit
			Symbol	ASCII code		
	Word unit	WW	57H,57H	One group of bit soft component (X, Y, M, S) is written, result uses 1 point as the unit.	10 words 160 points	
				One group of word soft component (D, T, C) is written, result uses 1 component as the unit.	64 points	
	Test	Bit unit	BT	42H,54H	Selectively use 1 point as the unit setting/reset independent bit soft component (X, Y, M, S, T, C)	20 points
		Word unit	WT	57H,54H	Selectively use 16 point as the unit setting/reset bit soft component (X, Y, M, S)	10 words 160 points
					Selectively use 1 component as the unit word writing soft component (D, T, C (besides high-speed C200~255 counter))	10 points
		PC	Remote run	RR	52H,52H	Transmit remote run/stop request to PLC
Remote stop	RS		52H,53H			
PC type readout	PC		50H,43H	PC type name readout (code)		
Grounding		GW	47H,57H	Configure/reset grounding marks to all connected PLC (M8126 for FX series)	1 point	
Connection requirement				Request can only be transmitted from the PLC when system configuration is 1:1	Maximum 64 words	
Circle loopback test		TT	54H,54H	Characters received from the computer will be transmitted directly back to the computer.	254 characters	

Note: The 16-bit and 32-bit components of the C word must be written separately and cannot be written consecutively.

e) Message waiting time. The minimum delay between PLC receiving a message and before responding, the value is "0" ~ "F", representing 0 ~ 150ms.

f) For example, sequence of content data: "M00160510101":

"M0016": means starting from M0016, "05": Means operating 5 variables, "10101": Represents the variable value.

g) The checksum is optional. Whether to add is determined by b13 of D8120.

h) LF CR

Optional, whether to add is determined by the selected protocol, and the protocol selection is determined by b15 of D8120.

## 2) Detailed command

## a) Batch reading of bit components (BR command).

Computer command (frame minimum 15 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	BR command	2	BR
5	Message wait time	1	A
6	First component	5	X0040
7	Component number N (1~255)	2	05
8	Check sum (optional)	2	47
9	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFBRAX00400547\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	STX	1	\STX
2	Station No.	2	05
3	PC No.	2	FF
4	Data of designated component	N	01101
5	ETX	1	\ETX
6	Check sum (optional)	2	E7
7	CR, LF (optional)	2	\CR\LF

Communication data example: \STX05FF01101\ETXE7\CR\LF

Computer acknowledgement:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication Data example: \ACK05FF

## b) Word soft component's batch readout (WR command).

Computer command (minimum 15 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	WR command	2	WR
5	Message wait time	1	0
6	First component	5	X0040
7	N(1~255/200/100) Component numbers N(1~255/200/100)	2	02
8	Check sum (optional)	2	48
9	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFWR0X00400248\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	STX	1	\STX
2	Station No.	2	05
3	PC No.	2	FF
4	Designated component data	N*4(16-Bit element)/ N*8(32-Bit element)	1234ABCD
5	ETX	1	\ETX
6	Check sum (optional)	2	08
7	CR, LF (optional)	2	\CR\LF

Communication data example: \STX05FF1234ABCD \ETXC08\CR\LF

Computer acknowledgement:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication data example: \ACK05FF

c) Bit soft component batch write (BW command)

Computer command (minimum 16 bytes)

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	BW command	2	BW
5	Message wait time	1	0
6	First component	5	M0903
7	Component number N(1~160)	2	05
8	Designated component data	N	01101
9	Check sum (optional)	2	2B
10	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFBW0M090305011012B\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication data example: \ACK05FF

## d) Word soft component batch write (WW command)

Computer command (minimum 19 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	WW command	2	WW
5	Message wait time	1	0
6	First component	5	M0640
7	Component number N(1~64)	2	02
8	Designated component data	N*4(16-Bit element)/ N*8(32-Bit element)	2347AB96
9	Check sum (optional)	2	0A
10	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFWW0M0640022347AB960A\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication data example: \ACK00FF

## e) Bit soft component testing (BT command)

Computer command (minimum 16 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	BT command	2	BT
5	Message wait time	1	0
6	Component number N(1~20)	2	03
7	Element 1	5	M0500
	Component 1 data (0/1)	1	1
	Component 2	5	S0100
	Component 2 data (0/1)	1	0
	Component 3	5	Y0001
	Component 3 data (0/1)	1	1
	...		
8	Check sum (optional)	2	EC
9	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFBT003M05001S01000Y00011EC\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication data example: \ACK05FF.

f) Word soft component testing (WT command)

Computer command (minimum 19 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	WT command	2	WT
5	Message wait time	1	0
6	Component number N(1~10)	2	03
7	Component 1	5	D0500
	Component 1 data (0/1)	4	1234
	Component 2	5	Y0100
	Component 2 data (0/1)	4	BCA9
	Component 3	5	CN100
	Component 3 (0/1)	4	0064
	...		
8	Check sum (optional)	2	07
9	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFWT003D05001234Y0100BCA9CN100006407\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication data example: \ACK05FF

g) Remote run/stop (RR/RS command)

Computer command (minimum 8 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	RR or RS command	2	RR
5	Message wait time	1	0
6	Check sum (optional)	2	C5
7	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFRR0C5\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication data example: \ACK05FF.

Conditions for effective execution of commands:

Remote operation: the programmable controller should be in a stopped state;

Remote stop: The programmable controller should be in forced operation mode.

Note: The forced operation mode will not be restored after power failure. When the programmable controller is in forced operation mode, if the power is turned off and then turned on, the special auxiliary relays M8035, M8036, and M8037 will be reset to off, and the programmable controller will remain stopped.

h) Read programmable controller type (PC command)

Computer command (frame minimum 8 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	0F
3	PC No.	2	FF
4	PC command	2	PC
5	Message wait time	1	0
6	Check sum (optional)	2	C5
7	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ0FFFPC0C5\CR\LF

PLC Response:

Serial	Name	Data length	Example
1	STX	1	\STX
2	Station No.	2	0F
3	PC No.	2	FF
4	Content PC No.	2	02
5	ETX	1	\ETX
6	Check sum (optional)	2	C5
7	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ0FFRR\ETXC5\CR\LF

Computer acknowledgement:

Serial	Name	Data length	Example
1	ACK	1	\ACK
2	Station No.	2	05
3	PC No.	2	FF

Communication Data example: \ACK05FF.

## i) Global function (GW command)

Computer command (minimum 9 bytes):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	FF
3	PC No.	2	FF
4	GW command	2	GW
5	Message wait time	1	0
	Control marks (1: ON, 0: OFF)	1	1
6	Check sum (optional)	2	17
7	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQFFFFGW01C5\CR\LF

PLC no response:

Note: The station number FF means all stations. M8126 is set when the global variable flag is turned on, and M8126 is reset when the global variable flag is turned off.

QUESTION: Regardless of turning ON or OFF the global variable mark, PLC still cannot be controlled or

Question: Regardless of whether the global variable flag is turned on or off, the PLC cannot be controlled or set by the address "FF", and the PLC can only be controlled or set by the specific station number of the PLC.

## j) Loopback test (TT command)

Computer command (minimum 11 bytes of frame):

Serial	Name	Data length	Example
1	ENQ	1	\ENQ
2	Station No.	2	05
3	PC No.	2	FF
4	TT command	2	TT
5	Message wait time	1	0
6	Character number N	2	08
7	Character series	N	12345678
8	Check sum (optional)	2	FB
9	CR, LF (optional)	2	\CR\LF

Communication data example: \ENQ05FFTT00812345678FB\CR\LF

PLC response:

Serial	Name	Data length	Example
1	STX	1	\STX
2	Station No.	2	05
3	PC No.	2	FF
4	Character number	2	08
5	Content PC No.	2	02
6	ETX	1	\ETX
7	Check sum (optional)	2	C5
8	CR, LF (optional)	2	\CR\LF

Communication data example: \STX05FF0812345678\ETXC5\CR\LF



## k) Error response

PLC error response:

Serial	Name	Data length	Example
1	NAK	1	\ NAK
2	Station No.	2	05
3	PC No.	2	FF
4	Error code	2	02
5	CR, LF (optional)	2	\CR\LF

Communication data example: \ NAK 05FF02\CR\LF

Note:

I) The error response does not exist and is checked;

II) Error code:

"02": Check sum error, sum check error;

"03": COMM Mode error, communication format error;

"06": char buf error, character area error;

"07": char error, the character error is not between 0 and F;

"10": PC code error, PC number error;

"18": control error, remote control error.

3) Connect request (PLC acts as the master station to send data to the computer)

If D8128 is not 0, PLC will actively send data through the port. The content of the sent data is the device designated by D8127, and the data length is designated by D8128. The sum of D8127 and D8128 does not exceed 8000, that is, data after D8000 cannot be accessed.

During data transmission, M8127 is set, and M8127 is reset after sending. If the data range is wrong, M8128 is set and no data is sent. M8129 is ON, the data register is processed by 8 bits, and OFF, the data register is processed by 16 bits.

Example:

User program: D8127 = 5, D8128 = 3, M8129 = OFF;

D5 = 501h, D6 = 602h, D7 = 703h.

PLC processing: M8127 = ON;

Send the following data through serial port 1: \STX05FE 050106020703\ETX;

M8127 = OFF.

Data sent by PLC actively:

Serial	Name	Data length	Example
1	STX	1	\STX
2	Station No.	2	05
3	PC No. (always FE)	2	FE
4	Content (D component data)	N*4 (or 2)	050106020703
5	ETX	1	\ETX
6	Check sum (optional)	2	C5
7	CR, LF (optional)	2	\CR\LF

Note: XP type COM2/COM3 can also use computer link protocol, the usage method is the same as COM1, please refer to the corresponding serial port settings for setting M and D.

## 5.7 MODBUS protocol description

Note: When the XP type COM2/COM3 port uses MODBUS protocol, use RS2 and MODBUS2 commands, other methods are the same as COM1, please refer to setting M and D of the corresponding serial port settings.

The bottom layer of MODBUS communication is RS485 signal, which can be connected by twisted pair, so the transmission distance is up to 1000 meters, the anti-interference performance is better, and the cost is low. It is widely used in the communication of industrial control equipment. Most inverter and controller manufacturers have adopted this protocol.

There are two transmission data formats: HEX code data and ASCII code, which are called MODBUS-RTU and MODBUS-ASC protocol respectively. The former is for direct data transmission, while the latter requires data to be converted to ASCII code for transmission. Therefore, MODBUS-RTU protocol features high efficiency and simple processing, and thus is preferred by users.

MODBUS is a single-master and multiple-slave communication method, which uses a master question-and-answer method. Each communication is initiated by the master station and the slave station passively responds. Therefore, controlled devices such as inverters generally have built-in slave station protocols, while control devices such as PLC's need to have master station protocol and slave station protocol.

### 5.7.1 MODBUS master station communication application

The COM1 communication port of H1U/H2U series PLC can use MODBUS-RTU and MODBUS-ASC commands, just set the corresponding value to the system register D8126. The communication of MODBUS instruction (master station) includes RS extended instruction and MODBUS instruction, which are explained as follows:

#### 1) RS instruction

Hardware configuration and software settings:

The RS command can only be used for COM1 communication. The connection mode is full-duplex or half-duplex. It supports the standard communication port of this machine and all communication expansion cards. You can set D8126=10h to start. The communication format and baud rate of this protocol are set by D8120. The definition of D8120 is shown in the following table:

Tag	Name	Content	
		0 (OFF)	1 (ON)
b0	Data length	7bit	8bit
b2 b1	Parity	00: None 01: Odd parity (ODD) 11: Even parity (EVEN)	
b3	Stop bit	1-bit	2-bit
b7 b6 b5 b4	Baud rate	0011: 300 0100: 600 0101: 1200 0110: 2400 0111: 4800 1000: 9600 1001: 19200 1010: 38400 1011: 57600 1100: 115200	

Tag	Name	Content	
		0 (OFF)	1 (ON)
b8	Start character	None	Yes, D8124 is the start character
b9	End character	None	Yes, D8125 is the start character
b10	Full or half duplex	Full duplex	Half duplex
b11	Retain	Not available	
b12	Retain	Not available	
b13	Sun check	Not attached	Attached
b14	Protocol	Not used	Used
b15	Control sequence	1 way	4 way

Note: b12~b14 are only used for computer link protocol and have nothing to do with RS instruction.

#### Protocol description:

The RS instruction format is:

RS (TXDADDR, TXDLEN, RXDADDR, RXDLEN);

TXDADDR: To send the data address, it must be a D component;

TXDLEN: The length of the sent data, which can be a variable or a constant;

RXDADDR: Receive data address: must be D component;

RXDLEN: The length of the received data, which can be a variable or a constant.

Send request command: M8122, if the M8122 ON and the RS command is driven, that is, from the D component address specified by TXDADDR, send TXDLEN data to COM1 (if a start character or a stop character is specified, they will be sent together ). The system automatically resets M8122 after sending.

Receiving flag: M8123, after receiving data, M8123 will be automatically set to on, and the reset will enter the next receiving state.

Receiving timeout: If the received data is insufficient and the time is longer than the set time (D8129×10ms), the receiving timeout, M8129 is set to on.

Note: If there is no data received, M8129 will not be set. The user needs to judge whether the communication is normal according to the amount of data received (D8123).

## 2) MODBUS instruction

Hardware configuration and software settings:

RS command can only be used for COM1 communication, if D8126=20h, COM1 protocol is MODBUS-RTU master station (command); if D8126=30h, COM1 protocol is MODBUS-ASC master station (command);

#### Protocol description:

The MODBUS instruction is valid for the serial port COM1, and the user can program through the MODBUS instruction to use the PLC as the master station to communicate with the MODBUS slave device. There are two kinds of MODBUS commands, one conforms to MODBUS-RTU protocol, and the other conforms to MODBUS-ASC protocol, which is determined by D8126. Which command is used is determined by the protocol format supported by the slave. If the slave supports both protocols and the user requires faster

communication, it is recommended to use the RTU protocol. The two protocols are only different in communication format, and they are the same for user programming. The following only describes the RTU protocol.

Multiple MODBUS instructions can exist at the same time and all of them are driven. The system will coordinate the sequential execution of the instructions. The MODBUS protocol requires that no matter whether it is write or read, the slave must have a response (except broadcast). A MODBUS instruction may take a long time to execute, and generally requires multiple scan cycles. In a scan cycle, instructions are driven, but not necessarily executed.

If there are multiple MODBUS instructions, the execution sequence is as follows: from the start, scan the first MODBUS instruction, if scanned, record the MODBUS parameters and execute it in the background. After execution, return to the user program, scan the next MODBUS instruction from the position of the MODBUS instruction that was just executed, and execute it again and again.

Command format: RS (ADDR&CMD, REGADDR, REGLLEN, DATABUF)

ADDR&CMD: Slave address and MODBUS function code, the upper 8 bits represent the slave address, that is, the target device address. The lower 8 bits represent the MODBUS function code, which is defined by the standard MODBUS protocol. The currently supported function codes are 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0f, 0x10. For specific meaning, please refer to the standard MODBUS protocol or the target device MODBUS protocol.

REGADDR: The address of the slave coil (1 bit) or register (16 bits) to be read or written, the value refers to the slave MODBUS protocol. Can be a component or a constant.

REGLLEN: The number of slave coils or registers to be read and written, which can be components or constants.

DATABUF: can only be D components. The machine is used to store the starting register of data, namely the data buffer. The length of the buffer is related to REGLLEN and should be at least 1. If the MODBUS command is read, after the instruction is successfully executed, read the slave data into the buffer, if the MODBUS command is write, send the buffer to the slave. The user needs to calculate the length of the buffer when designing the program and reserve enough registers as the buffer.

Related status flags:

M8122: MODBUS instruction execution status indication. When it is OFF, it means the instruction has been executed, and when it is ON, it is executing. If M8122 is OFF, and the instruction is effective in a scan cycle, and M8122 is set to ON, the system will record the instruction parameters and transfer to the background to execute the communication requirements of the instruction. After the communication is executed, when it runs to the position of this instruction again, regardless of whether the power flow of the instruction is valid or not, M8122 will be reset to OFF, and the next effective power flow instruction will be scanned immediately, and the instruction parameters will be recorded and executed in the background. The communication requirements of the instruction.

M8123: Command communication status indication, ON means communication is abnormal, OFF means communication is normal;

M8063: Command error indication, the error code is stored in D8063;

D8063: Error code (see the list of communication error codes).

Note:

When reading and writing registers, H1U supports up to 50 16-bit registers at a time. H2U meets the requirements of the MODBUS protocol standard and supports up to 125 registers and 120 registers at a time. If the read/write exceeds the maximum value, a parameter error will be reported.

When reading and writing coils, H1U supports up to 800 coils at a time. H2U meets the requirements of the MODBUS protocol standard and supports up to 2000 coils for reading and 1968 coils for writing at a time. If the read/write exceeds the maximum value, a parameter error will be reported.

Example 1: Continuously read the register with the address of 100 from the slave, and the data is stored in D10. initialization:

D8126 = H0020 Set the communication protocol to MODBUS RTU instruction;

D8120 = H0081 Set the communication format of COM1 as: 9600, 8N1;

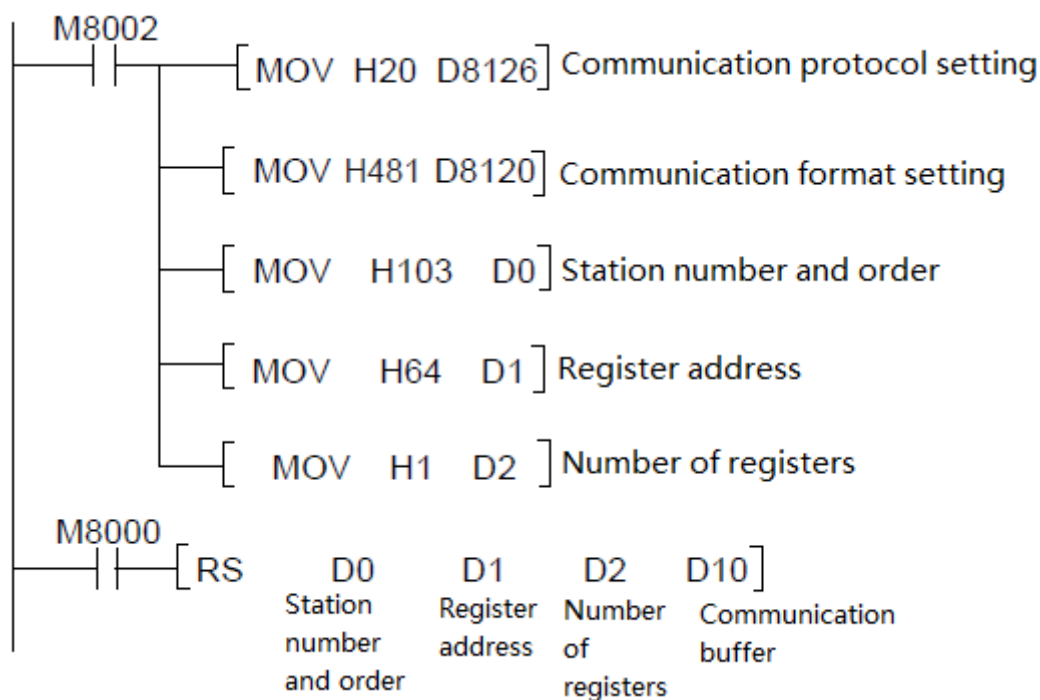
D0 = H0103 Addr & Cmd slave address is 01 and MODBUS command code is 03, read register;

D1 = H0064 RegAddr the register address of the slave to be operated;

D2 = H0001 RegLen the number of registers to be operated;

D10 = PLC data buffer. In this example, the data is stored in D10 after the read command communication is successful.

The ladder diagram is as follows:

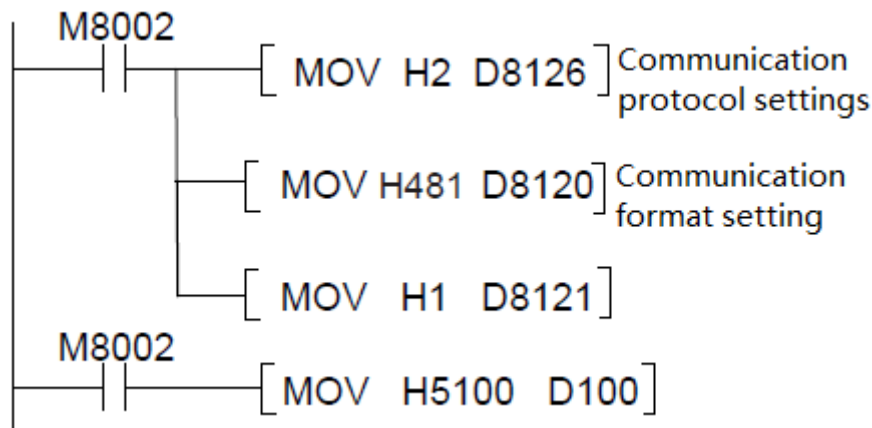


Execution result: After power-on, the PLC keeps reading the register with the slave address of 100, and sends

the following frame of data (hexadecimal) through COM1: 01 03 00 64 00 01 C5 D5

- 01: represents the slave address, the upper 8 bits of D0;
- 03: MODBUS command code, the lower 8 bits of D0, meaning to read the slave register;
- 00 64: The address of the slave register to be read, the value of D1;
- 00 01: The number of registers to be read, the value of D2;
- C5 D5: CRC check code.

If the slave is also H1U/H2U series PLC, it is set to MODBUS RTU slave station protocol, the ladder diagram is as follows:



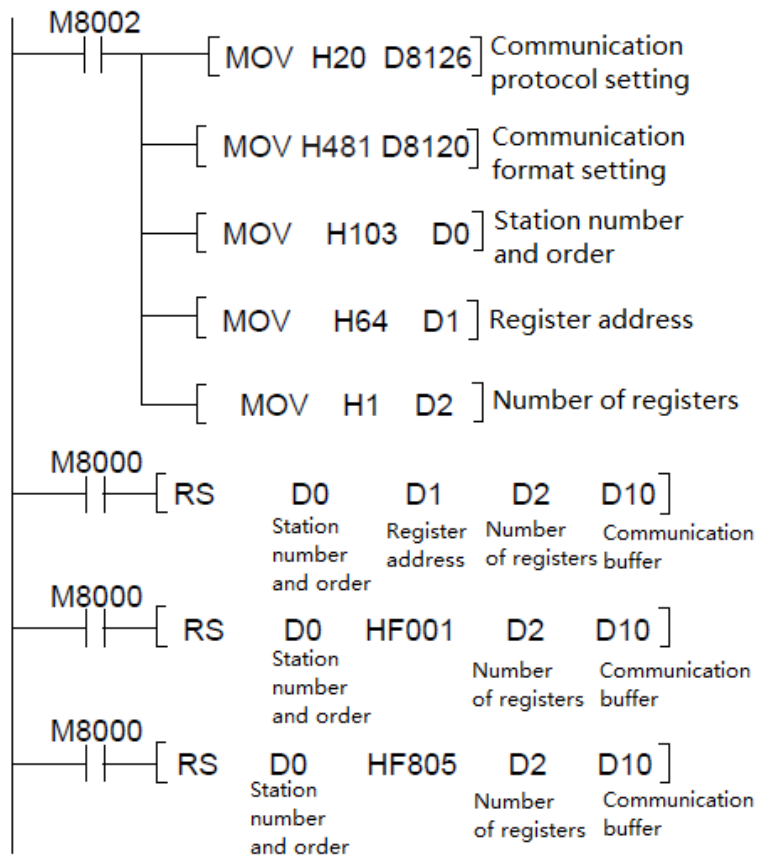
The slave responds correctly to the data frame (hexadecimal): 01 03 02 51 00 85 D4

The slave sends D100 (register address is H0064) to the master:

- 01: represents the slave address;
- 03: MODBUS command code;
- 02: Indicates that 2 bytes of valid data are returned;
- 51 00: Register data, that is, the value of D100;
- 85 D4: CRC check code.

Example 2: Use three MODBUS instructions to read the registers of the slave address H0064, F001 and F805 respectively, and the data is stored in D10, D20 and D30.

The ladder diagram is as follows:



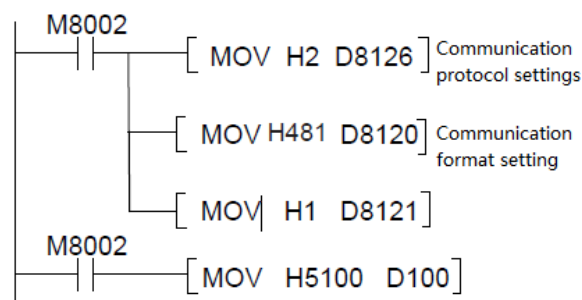
As a result of the execution, the PLC sends the following three frames of data (hexadecimal) sequentially through the serial port COM1:

01 03 00 64 00 01 C5 D5

01 03 F0 01 00 01 E6 CA

01 03 F8 05 00 01 A5 6B

Reply: The slave is still H1U/H2U series PLC, set to MODBUS RTU slave station protocol, the ladder diagram is as follows:



Slave response:

For the first frame number, the response data frame of the slave is (hexadecimal): 01 03 02 51 00 85 D4  
The meaning is: the slave sends the value H5100 of D100 (the address of the D100 register is H0064) to the master;

For the second frame number, the slave response data frame is (hexadecimal): 01 03 02 00 00 B8 44  
The meaning is: the slave sends the value H0000 of T1 (the T1 register address is F001, see "MODBUS Slave

Communication Application" in section 5.7.2) to the master;

For the third frame number, the response data frame of the slave is (hexadecimal): 01 83 02 C0 F1

The meaning is: read register error.

01: slave address;

83: Read register error;

02: Error code, address error, because the register at address HF805 does not exist;

C0 F1: CRC check code.

### 5.7.2 MODBUS slave communication application

In some industrial applications, as part of the industrial automation system, PLC controllers must be monitored by the automation control network. A typical host computer, such as a DCS or an industrial PC running configuration software, is used for this purpose and must communicate with the PLC with the MODBUS master protocol., at this time the PLC communication port needs to communicate with the host computer through the MODBUS slave station protocol. H1U/H2U series PLC has built-in MODBUS-RTU slave station protocol and MODBUS-ASC slave station protocol, and the protocol can be run on COM0 and COM1 ports.

#### 1) Hardware configuration and software setting:

COM0 setting: if D8116=02h, COM0 protocol is MODBUS-RTU slave; if D8116=03h, COM0 protocol is MODBUS-ASC slave;

COM1 setting: if D8126=02h, COM1 protocol is MODBUS-RTU slave; if D8126=03h, COM1 protocol is MODBUS-ASC slave;

#### 2) Protocol description:

The MODBUS slave station protocol includes MODBUS RTU protocol (hereinafter referred to as RTU protocol) and MODBUS ASC protocol (hereinafter referred to as ASC protocol). The difference between the two is in the data link. The data transmitted by the MODBUS RTU protocol is real data, and the data transmitted by the ASC protocol is data converted to ASC code. In addition, the two are also different in frame structure. The MODBUS RTU protocol distinguishes data frames by time. If the data is not received within a period of 3.5 bytes, it is considered that the other party's data has been transmitted; the ASC protocol is based on the ASC code. ":" is the frame start character, and \r\n(0D0Ah) is the frame end character. From the perspective of communication efficiency, the RTU protocol is about twice the ASC protocol.

For details, please refer to the relevant documents of the standard MODBUS protocol. These documents are open to download and can be downloaded on the internet or on the official website of MODBUS.

Table of variables and signs related to control:

Variable	Description	Remark
D8120	Communication format setting	Example: 81h: 9600bps, 8N1 91h: 19200bps, 8N1
D8121	Set the slave address of this PLC	Update valid at any time while the program is running
D8126	Slave station protocol setting	02h: MODBUS RTU slave station 03h: MODBUS ASC slave station
M8063	MODBUS communication error indication	Read only, user program is cleared or cleared



		once PLC power reboot
D8063	Communication error code (see the list of communication error codes)	Read only, user program is cleared or cleared once PLC power reboot

### 3) MODBUS slave station communication protocol description (MODBUS function code and data addressing)

When H1U/H2U is used as a MODBUS slave, it supports MODBUS protocol function codes 0x01, 0x03, 0x05, 0x06, 0x0f, 0x10; through these function codes, you can read and write M, S, T, C, X (read only) coils and D, T, C registers.

When the MODBUS communication master accesses (reads or rewrites) the internal variables of the PLC slave, it must follow the following communication command frame definition and the variable address index method in order to carry out normal communication operations.

#### MODBUS frame format (take MODBUS-RTU as an example)

##### a) Function code 0x01 (01), 0x02 (02): Read coil request frame format:

Slave address + 0x01 + coil start address + coil number + CRC check

Serial number	Data (byte) meaning	Byte quantity	Instructions
1	Slaver address	1 byte	Value range 1~247, defined by D8121
2	0x01 (function code)	1 byte	Read coil
3	Coil starting address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Coil number	2 bytes	Lower byte follow higher byte (N)
5	CRC check	2 bytes	Lower byte follow higher byte

Response frame format: slave address + 0x01 + byte count + coil status + CRC

Serial number	Data (byte) meaning	Byte quantity	Instructions
1	Slaver address	1 byte	Value range 1~247, defined by D8121
2	0x01 (function code)	1 byte	Read coil
3	Byte number	1 byte	Value: $[(N+7)/8]$
4	Coil state	$[(N+7)/8]$ byte	A byte consistent of eight coils, and if the last one has less than eight bits, the undefined bits should be set with 0. The first eight coils are in first byte, and the coil with min address is in lowest bit. And so forth.
5	CRC check	2 bytes	Lower byte follow higher byte

Error Response: Please refer to the error response frame.

Note: N, up to 2000(H2U), up to 800 (H1U).

## b) Function 0x03(03): read register

Request Frame Format: slave address + 0x03 + register start address + register count + CRC

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x03 (function code)	1 byte	Read register
3	Register starting address	2 bytes	Lower byte follow higher byte, refer to register addressing
4	Register number	2 bytes	Lower byte follow higher byte (N)
5	CRC check	2 bytes	Lower byte follow higher byte

Response Frame Format: slave address + 0x03 + byte count + register value + CRC

Serial number	Data(byte) meaning	Byte quantity	instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x03 (function code)	1 byte	Read register
3	Byte number	1 byte	Value: N*2
4	Register value	N*2 byte	One register value is expressed by two bytes, and the lower byte follows higher byte. The register with lower address is set in front
5	CRC check	2 bytes	Lower byte follow higher byte

Error Response: Please refer to the error response frame.

Note: N, up to 1250(H2U), up to 50 (H1U).

## c) Function code 0x05(05): write single coil

Request Frame Format: slave address + 0x05 + coil address + coil status + CRC

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x05 (function code)	1 byte	Write single coil
3	Coil address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Coil state	2 bytes	Lower byte follow higher byte, If it is not 0, it is valid.
5	CRC check	2 bytes	Lower byte follow higher byte

Response Frame Format: slave address + 0x05 + coil address + coil status + CRC check

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x05 (function code)	1 byte	Write single coil
3	Coil address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Coil state	2 bytes	Lower byte follow higher byte, If it is not 0, it is valid.
5	CRC check	2 bytes	Lower byte follow higher byte

Error Response: Please refer to the error response frame.

d) Function code 0x06 (06): write single register

Request Frame Format: slave address + 0x06 + register address + register value + CRC

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x06 (function code)	1 byte	Write single register
3	Register address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Register state	2 bytes	Lower byte follow higher byte, If it is not 0, it is valid.
5	CRC check	2 bytes	Lower byte follow higher byte

Response frame format: slave address + 0x06 + register address + register value + CRC check

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x06 (function code)	1 byte	Write single register
3	Register address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Register state	2 bytes	Lower byte follow higher byte, If it is not 0, it is valid.
5	CRC check	2 bytes	Lower byte follow higher byte

Error response error: referring to error response frame.

## e) Function code 0x0f (15): write multi-coil

Request frame format: slaver address + 0x0f + coil starting address + coil number + byte number + coil state + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x0f (function code)	1 byte	Write multi-coil
3	Coil starting address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Coil number	2 bytes	Lower byte follow higher byte.
5	Byte count	1 byte	Value: $\lceil (N+7) / 8 \rceil$
6	Coil status	$\lceil (N+7) / 8 \rceil$ byte	A byte consistent of eight coils, and if the last one has less than eight bits, the undefined bits should be set with 0. The first eight coils are in first byte, and the coil with min address is in lowest bit. And so forth.
7	CRC check	2 bytes	Lower byte follow higher byte

Response Frame Format: slave address + 0x05 + start address of coil + coil count + CRC check

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x0f (function code)	1 byte	Write multi-coil
3	Coil starting address	2 bytes	Lower byte follow higher byte, refer to coil addressing
4	Coil number	2 bytes	Lower byte follow higher byte
5	CRC check	2 bytes	Lower byte follow higher byte

Error response error: referring to error response frame.

Note: N, up to 1968 (H2U), up to 800 (H1U).

## f) Function code 0x10 (16): write multi-register

Request frame format: slaver address + 0x10 + register starting address + register number + byte number + register value + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x10 (function code)	1 byte	Write multi-register
3	Start address of register	2 bytes	Lower byte follow higher byte, refer to register addressing
4	Register count	2 bytes	Lower byte follow higher byte
5	Byte count	1 byte	Value: $N*2$
6	Register value	$N*2$ ( $N*4$ )	
7	CRC check	2 bytes	Lower bytes follow higher bytes

Response frame format: slaver address + 0x05 + coil starting address + coil number + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x10 (function code)	1 byte	Write multi-register
3	Start address of register	2 bytes	Lower byte follow higher byte, refer to register addressing
4	Register count	2 bytes	Lower byte follow higher byte
5	CRC check	2 bytes	Lower byte follow higher byte

Error response error: referring to error response frame.

Note: N, up to 120 (H2U), up to 50 (H1U).

g) Function code 0x60: Specify the PLC component to be read

HMI request frame format: slave address + 0x60 + data block identification + number of bit components + number of registers + bit component address + register address + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247
2	0x60 (function code)	1 byte	QLink customized
3	Data block identification	2 bytes	Only, 0xffff is not allowed.
4	Number of bit components N	2 bytes	Lower byte follow higher byte
5	Number of registers N	2 bytes	Lower byte follow higher byte
6	Scattered bit component address	N*2	Lower byte follow higher byte, refer to register addressing
7	Scattered register address	N*2	
8	CRC check	2 bytes	

PLC response frame format: slave address + 0x60 + data block identification + number of bit components + number of registers + bit component value + register value + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247
2	0x60 (function code)	1 byte	QLink customized
3	Data block identification	2 bytes	The only one is useful in the following data
4	Number of bit components N	1 byte	
5	Number of register N	1 byte	
6	Bit component value	(N+7)/8	Lower byte follow higher byte, refer to register addressing
7	Registers value	N*2 or N*4	
8	CRC check	2 bytes	

Error response: See error response frame.

Note: The maximum value of N is 120 for H2U and 50 for H1U.

h) Function code 0x61: data block read request

HMI request frame format: slave address + 0x61 + data block identification + number of bit components + number of registers + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247
2	0x61 (function code)	1 byte	QLink customized
3	Data block identification	2 byte	The only one is useful in the following data
4	Number of bit components N	1 byte	
5	Number of bit registers N	1 byte	
6	CRC check	2 byte	Lower byte follow higher byte, refer to register addressing

PLC response frame format: slave address + 0x60 + data block identification + number of bit components + number of registers + bit component value + register value + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247
2	0x61 (function code)	1 byte	QLink customized
3	Data block identification	2 bytes	The only one is useful in the following data
4	Number of bit components N	1 byte	
5	Number of registers N	1 byte	
6	Bit component value	(N+7)/8	Lower byte follow higher byte, refer to register addressing
7	Registers value	N*2 or N*4	
8	CRC check	2 bytes	

Error response: See error response frame.

Note: The maximum value of N is 120 for H2U and 50 for H1U.

i) Error response frame

Error response: slave address + (function code + 0x80) + error code + CRC check.

Serial number	Data(byte) meaning	Byte quantity	Instructions
1	Slave address	1 byte	Value range 1~247, defined by D8121
2	0x80 (function code)	1 byte	Error function code
3	Error code	1 byte	1~4
4	CRC check	2 bytes	Lower bytes follow higher bytes

### Variable addressing

a) Coil addressing

Coil: refers to a bit variable, there are only two states 0 and 1. This PLC contains variables such as M, S, T, C, X, Y, etc.

Variable name	Starting address	Coil count	Instructions
M0~3071	0 (0)	3072	
M8000~M8256	0x1F40 (8000)	256	
S0~S999	0xE000 (57344)	1000	
T0~T256	0xF000 (61440)	256	
C0~C255	0xF400 (62464)	256	
X0~X255 X0~X255	0xF800 (63488) 0xF800 (63488)	256	
Y0~Y255 Y0~Y255	0xFC00 (64512) 0xFC00 (64512)	256	

## b) Register addressing

Register: Refers to 16-bit or 32-bit variables. In this PLC, 16-bit variables include D, T, and C0~199; 32-bit variables are C200~255.

Variable name	Start address	Coil count	Instructions
D0~D8255	0 (0)	8256	
T0~T255	0xF000 (61440)	256	
C0~C199	0xF400 (62464)	200	
C200~C255	0xF700 (63232)	56	32-bit register

## Description:

When accessing C200~C255 32-bit registers through MODBUS, one register is treated as two registers, and one 32-bit register occupies two 16-register spaces. For example, if the user wants to read or write the 4 registers C205~C208, the MODBUS address is 0xF70A (0xF700+10), and the number of registers is 8 (4\*2). 32-bit registers do not support the 0x06 function code (writing a single register).

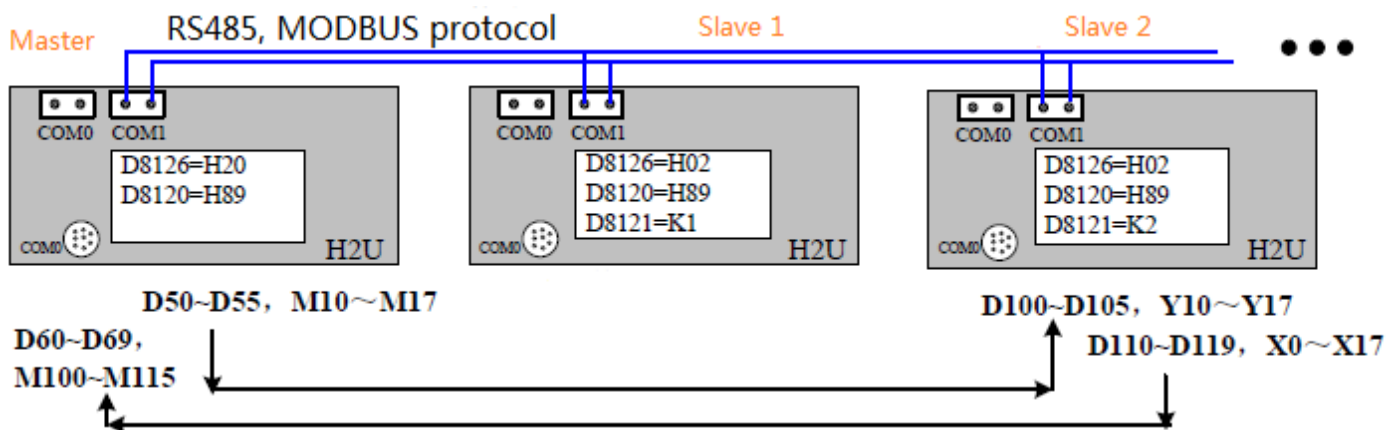
### 5.7.3 The programming method of using MODBUS protocol communication between multiple PLC's.

For a system with 2 or more PLCs in parallel, programming using MODBUS protocol is simple and flexible. The combined system is more convenient. The MODBUS communication system is a master and multiple slave. The data exchange required for communication is completely initiated by the master. All slaves receive and respond passively. The communication-related programming is mainly carried out in the program of the master. For the slaves, you only need to configure the communication protocol, communication format, and the station number of this machine, and then properly process the communication data.

Example 1: For the following multiple PLC communication connections, to realize the data exchange between the master PLC and #2 slave PLC, as shown in the figure below, the exchanged data is:

Master (D50~D55, M10~M17) → Slave (D100~D105, Y10~Y17);

Slave (D110~D119, X0~X17) → Master (D60~D69, M100~M115).



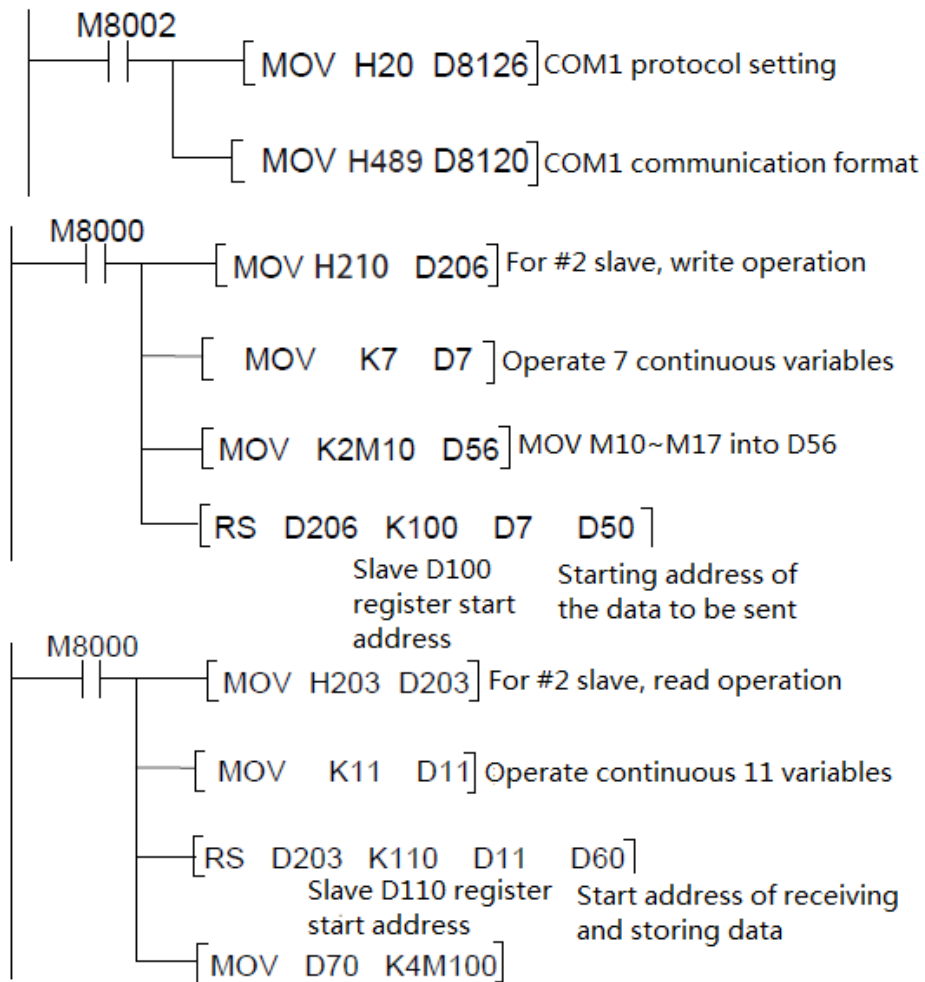
Programming method:

The COM1 communication port of the main PLC is configured as the MODBUS master station protocol, 9600bps, 8N2 format, and the data exchange (read and write) is all completed by the main PLC. Because some of the bit variables X, Y, and M need to be exchanged, these variables are integrated into D variables and exchanged in batches in a continuous D variable area. Master and slave respectively perform the combination and analysis of the bit variables. Such exchange features high efficiency and simple programming. Here, the variables M10~M17 in the master station are combined into D56, so the data to be sent by the master station is D100~D106, a total of 7 D variables; the variables X0~X17 in the slave station are combined into D120, and the data to be read by the master station is 11 D registers from D110 to D120.

An example of master station programming with RS extended instructions is:

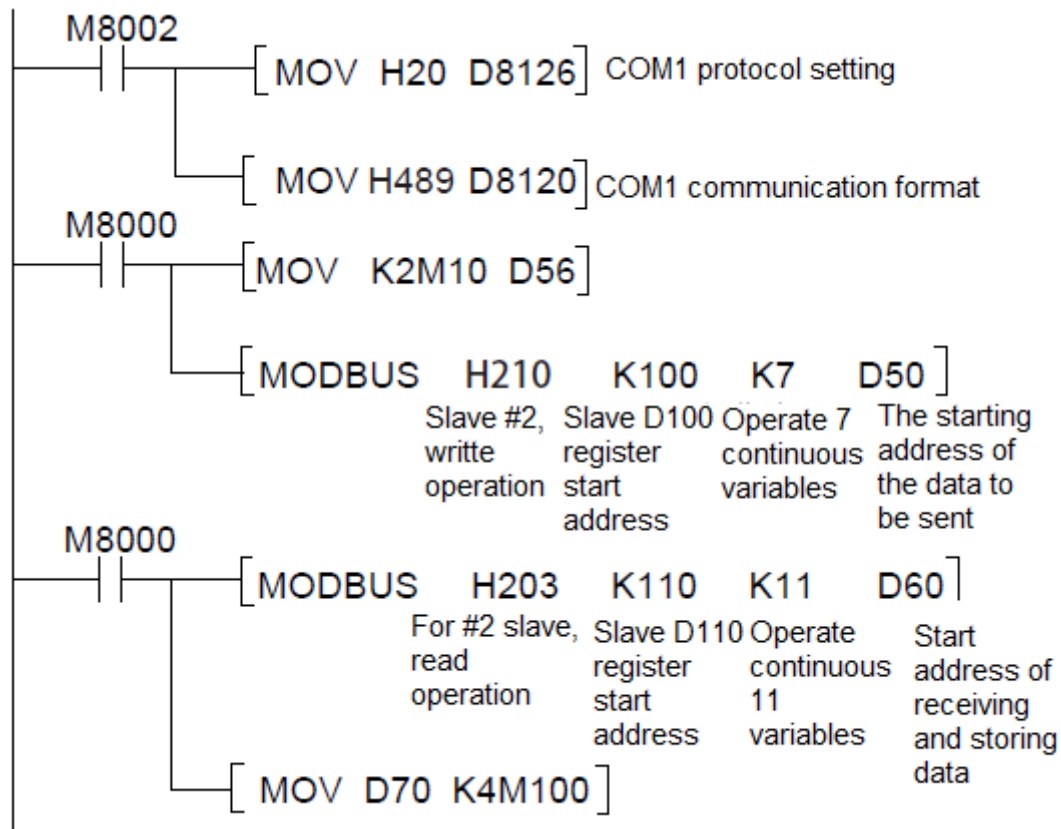
Initialize the COM1 port as MODBUS-RTU, 9600bps, 8N2





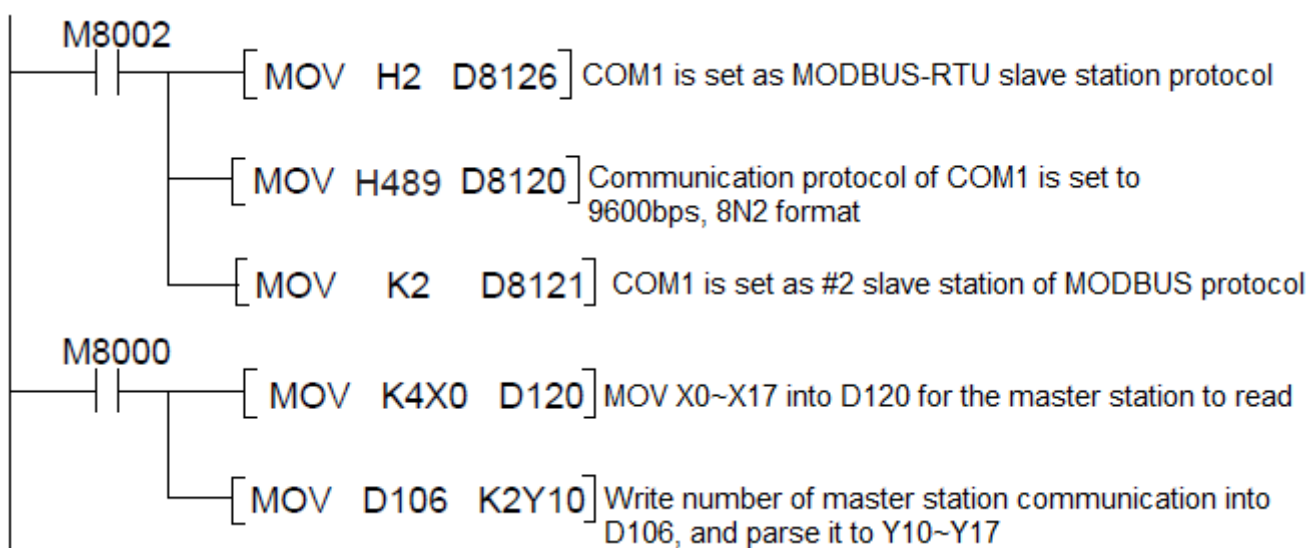
To achieve the same data exchange function, master station programming of MODBUS instruction enables refined statements and reduced register occupation:

Initialize the COM1 port as MODBUS-RTU, 9600bps, 8N2.



#2 The programming requirement of the slave station is to configure the COM1 port as a MODBUS-RTU slave station. The communication format is the same as that of the master station i.e. 9600bps, 8N2. Set the station number of the machine to 2, and refresh the information read by the master station in time. For the maintenance of data registers and the data written by the master station by communication, the programming examples are as follows:

Set the COM1 port to MODBUS-RTU slave station protocol, 9600bps, 8N2, and station number 2.



For the programming methods of other slave stations, please refer to the above-mentioned slave station programming methods, and pay attention to the set station number not to be repeated.

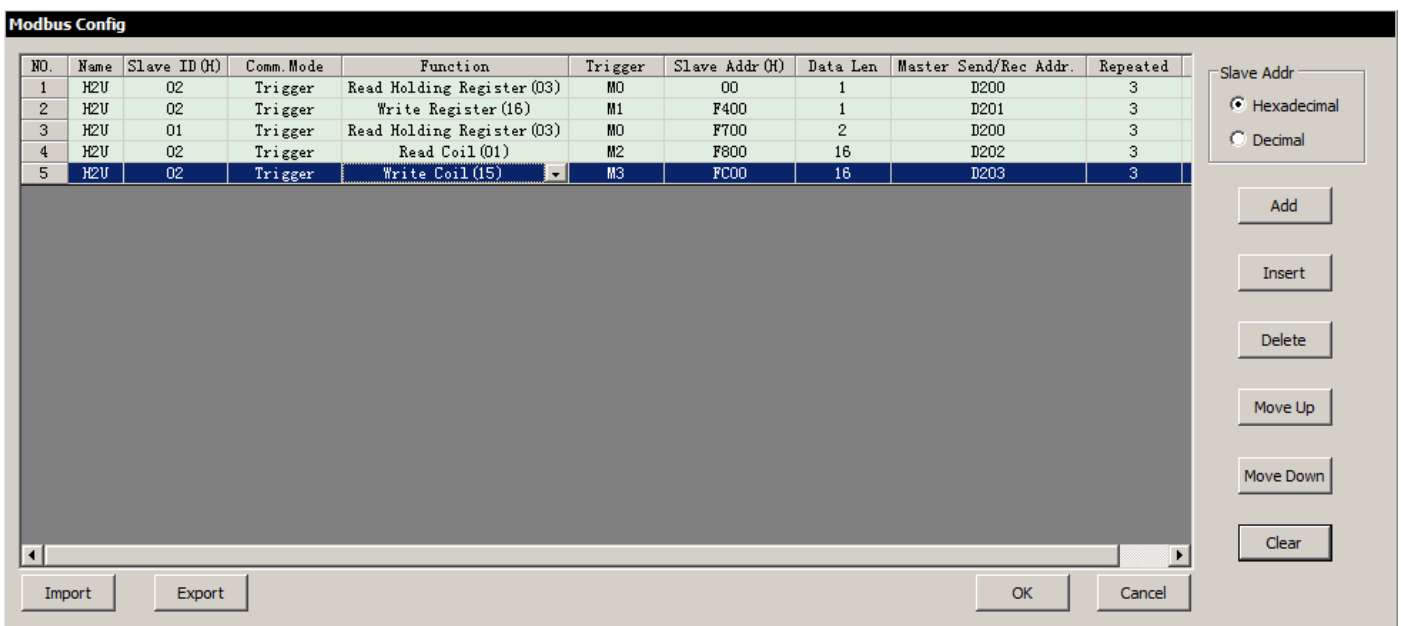
### 5.7.4 MODBUS configuration and programming method

Modbus configuration can realize the centralized configuration of Modbus communication data and download the configuration data to the Modbus master station. The PLC software realizes communication with the slave device according to the configuration information to achieve the purpose of data exchange; it reduces the difficulty of using Modbus, reduces the workload, and improves the effect of user experience.

#### 1. Master station Modbus configuration

##### 1.1 Configuration interface

Double-click "Modbus Configuration" in the "Project Management" window to open the "Modbus Config" window.



Configuration operation:

Add: Add a new configuration record at the end of the configuration data;

Insert: Add a configuration record at the current position;

Delete: delete the selected configuration record;

Move up: adjust the position of the selected row;

Move down: adjust the position of the selected row;

Clear: Clear all configuration information in the window;

OK: save the modification, close the window;

Cancel: Close the window without saving the changes.

##### 1.2 Configuration data

- 1) Slave station number (H): the hexadecimal representation of the slave station number, the slave station can be PLC, MD, IS;
- 2) Communication method: The communication method is divided into "trigger" and "cycle";

- a. Trigger: "Trigger Condition" is set to "ON", the communication operation is triggered; the trigger condition can be the following types of components: M, C, T, S
  - b. Cycle: "Trigger Condition" is "Empty", the communication operation is executed cyclically
- 3) COM: Modbus communication serial port of the master station;
  - 4) Slave register address (H): the starting operation address of the slave terminal of the communication operation, expressed in hexadecimal;

This document only lists PLC addressing, For MD series AC drive\IS series servo drive addressing please refer to the function code description of the corresponding device;

PLC Modbus protocol variable addressing table: coil addressing

Variable Name	Start Address	Number of Coils	Description
M0~M3071	0 (0)	3072	
M8000~M8255	0x1F40 (8000)	256	
S0~S999	0xE000 (57344)	1000	
T0~T255	0xF000 (61440)	256	
C0~C255	0xF400 (62464)	256	
X0~X255	0xF800 (63488)	256	
Y0~Y255	0xFC00 (64512)	256	

Register addressing:

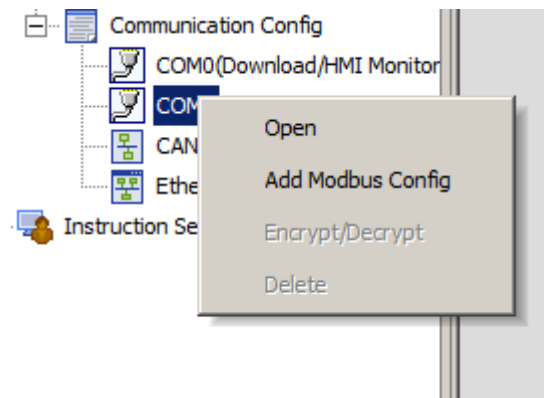
Variable Name	Start Address	Number of Registers	Description
D0~D8255	0 (0)	8256	16-bit register
T0~T255	0xF000 (61440)	256	16-bit register
C0 to C199	0xF400 (62464)	200	16-bit register
C200 to C255	0xF700 (63232)	56	32-bit register

- 5) Data length: The data length of read register and write register indicates the number of registers, that is, the n registers starting from "Slave Register Address"; the data length of read coil and write coil indicates the number of coils, that is, "Slave Register Address". N coils
- 6) Master station buffer address: the master station side buffer area when the master and slave stations exchange data; the buffer area is the D component of the PLC, and the number of components is determined by the data length

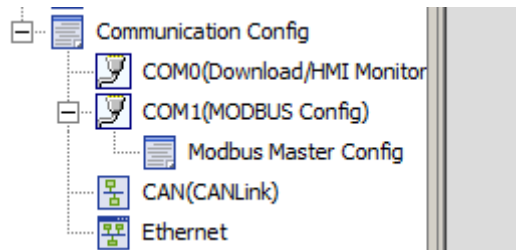
### 1.3 Configure encryption & decryption

In order to facilitate users to implement encryption protection on the project, the configuration encryption/decryption function is provided. Users can set according to their needs. After re-downloading to the device, the encrypted configuration parameters can only be viewed after entering the password as required. The specific operation instructions are as follows:

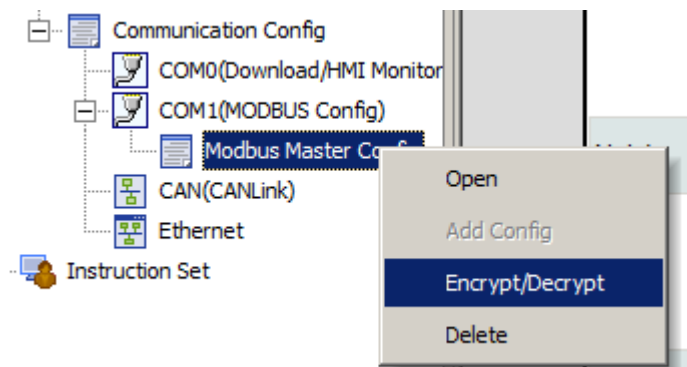
- Select a communication port configuration. Select "COM1 (Modbus master configuration)" as an example:



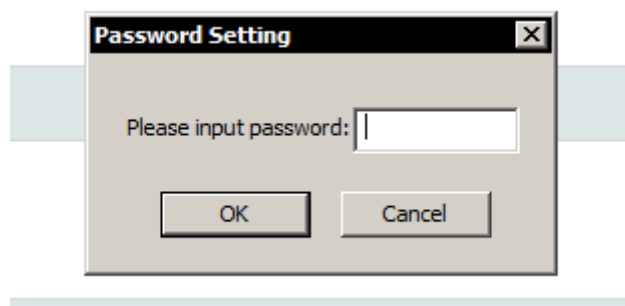
Right-click "Add Modbus Config" and add it as follows:



Click "Modbus Config", right-click and enter "Encrypt/Decrypt" to set specific passwords.



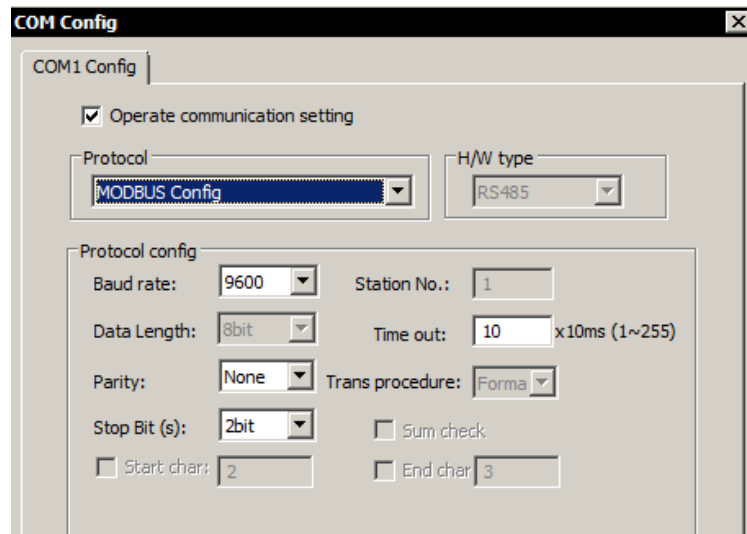
After configuring encryption, you must enter the password to view it. After the password is entered, the original program is decrypted. Through the above path, select decryption in "Encrypt/Decrypt", and then decrypt it after entering the password.



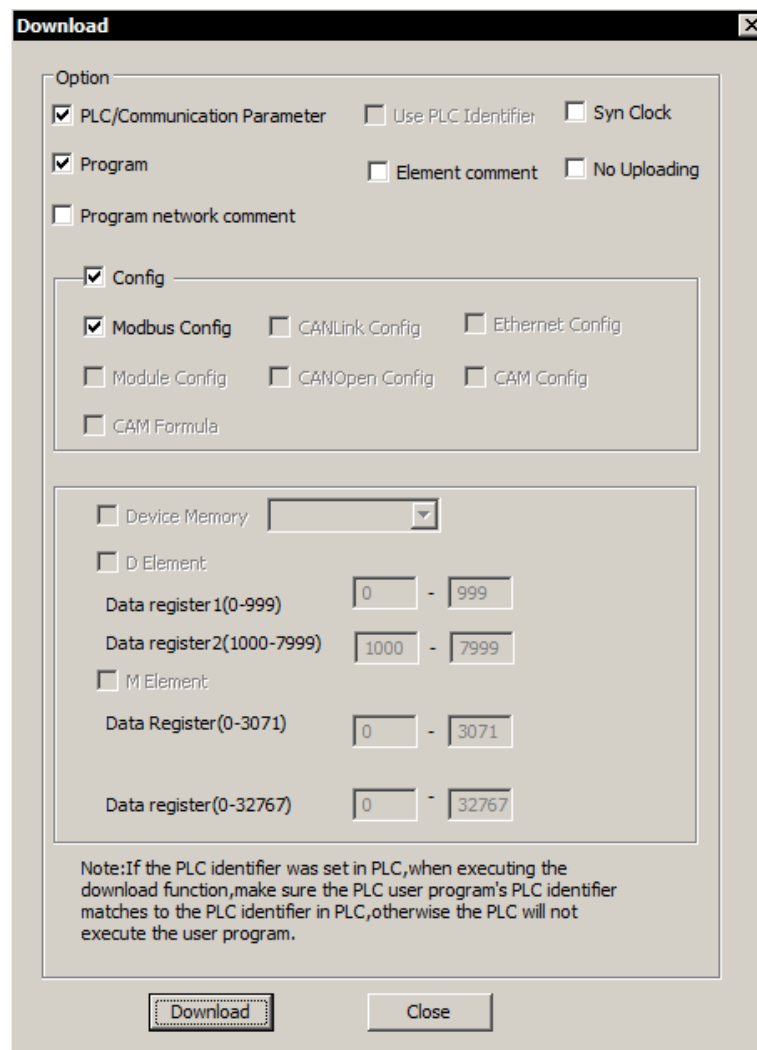
[Note] After encryption, you must enter the password to view the configuration. After you enter the password once, the configuration will be automatically decrypted (that is, you do not need to enter the password to view it on the device in the future).

#### 1.4 Master station communication settings

1) Select "Modbus Config" in "Communication Config-> COM1 Config-> Protocol", set the station number and other communication configuration information; click "OK" to save the communication configuration;



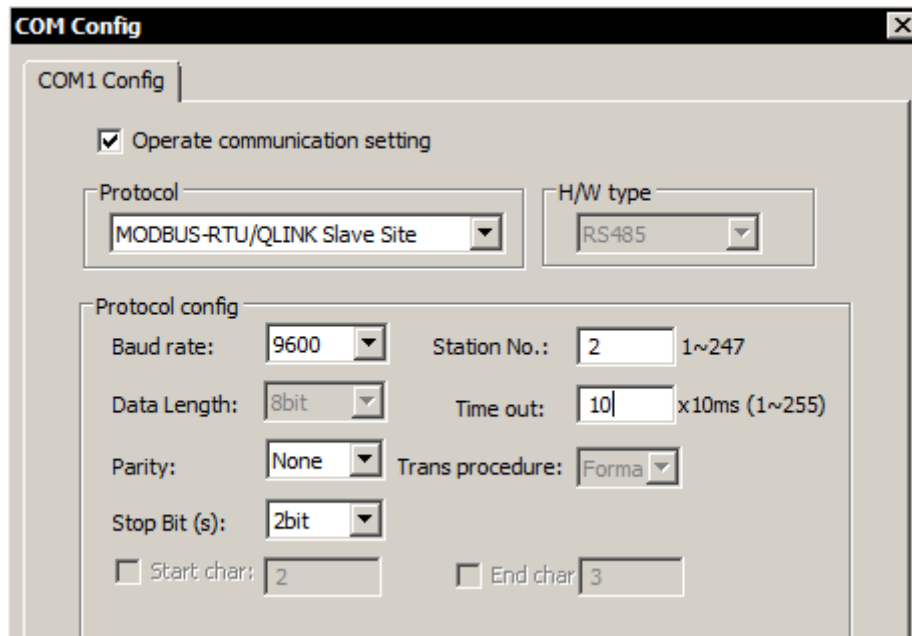
2) Click the menu item "PLC->Download", the toolbar "Download" button or the shortcut key F8, in the open "Download" window, check "Modbus config" and click "Download" to complete Master station communication configuration;



[Note] After the communication protocol is set, you only need to download "Modbus";

## 2. PLC slave communication configuration

Select "Modbus-RTU/QLINK Slave" in "COM Config-> COM1 Config ->Protocol", set communication configuration information such as station number, and click "OK" to save the communication configuration; check in the "Download" window "System parameters", and click download to complete the Modbus slave station protocol configuration;



[Note] The slave station has no "Modbus configuration" data;

Remarks

- 1) For more information on Modbus, please refer to "AutoShop Help -> PLC Communication Application";
- 2) For the Modbus slave communication configuration of MD\IS, please refer to the help file of the corresponding device

### 5.7.5 Inovance QLink protocol and programming method

#### 1) Protocol description:

In the HMI page, because the user program design cannot combine the variables, it takes multiple frames of data to read the PLC data, resulting in lower communication efficiency and slow data refresh. In order to solve this problem, the Qlink protocol is specially developed, and HMI can quickly read PLC data through this protocol.

This protocol is based on the MODBUS protocol. Each HMI page allows up to two configuration frames. The data length of a single configuration frame is set to a maximum of 240 bytes, that is, 120 registers or coils (C200~C255 occupy two registers).

The functions of write and continuous read can refer to the MODBUS slave station protocol to complete communication through normal functions. H1U/H2U PLC MODBUS slave station protocol is a subset of this protocol, which can respond to MODBUS frames normally.

#### 2) Programming method:

Step 1: The user needs to configure H2U as the MODBUS communication mode first, and select the protocol type as the fast protocol, which is easy to distinguish from the previous MODBUS protocol. The current physical communication serial port, if it is serial port 0, set D8110 (communication format, baud rate), D8111 (slave address), and D8116 (protocol). If it is serial port 1, Set D8120 (communication format, baud rate), D8121 (slave station address), D8126 (protocol). At the same time, the user needs to specify the H2U fast protocol in the HMI background to communicate normally.

Step 2: When the HMI switches to a new page, the HMI needs to download the configuration information to the PLC, and it can download two pieces of information at most. After the PLC terminal receives the configuration information, it returns the register and coil data corresponding to the information to the HMI, and stores the configuration information in the PLC memory until the PLC is reset or the HMI downloads the latest configuration information to cover the information. If the page read-through amount exceeds the amount that two pieces of configuration information can contain, the remaining registers or coils are processed using the standard MODBUS protocol.

Step 3: If the HMI only stays on this page, the PLC data in the configuration information paired with the data block identifier can be quickly read through the data block identifier.

Exception handling: When the HMI communicates with the PLC, if the data block identification reading information causes the communication timeout (excluding communication errors or check errors), the HMI needs to download the configuration frame information again, because the PLC may have restarted at this time, and the configuration information It may no longer exist.

### 3) Programming example:

If the communication baud rate is set to 9600bps, the communication data format is 8 data bits, no parity bit, 1 stop bit, and the PLC station number is set to station 1. Connect the COM1 port of the HMI and the COM0 port of the PLC through a suitable cable and set as shown in the figure below to communicate through the Inovance Qlink protocol.

#### a) InoTouch Editor software configuration:

Open the InoTouch Editor software: File>New>Device>Communication Settings. The HMI configuration is as follows:



**Device** [X]

Device

Device name:

Producer Name:  Connect mode:

Device model:

---

Connect port:   Preset station No.:

Use broadcast command

---

PLC addr.interval(words):

PLC max. read word counts:  PLC max. write word counts:

**Communication Setting** [X]

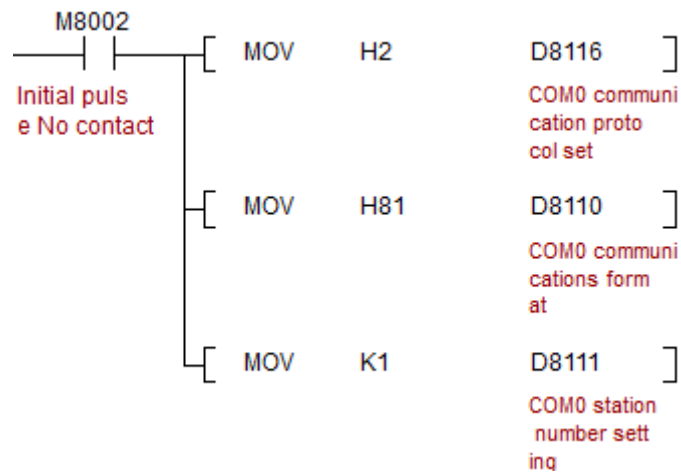
Comm. mode:

---

COM: <input type="text" value="COM1"/>	Timeout: <input type="text" value="0.2"/>
Baud rate: <input type="text" value="9600"/>	Comm. delay: <input type="text" value="0"/> ms
Data bit: <input type="text" value="8 bits"/>	Response delay: <input type="text" value="0"/> ms
Parity: <input type="text" value="None"/>	Resend counts: <input type="text" value="6"/>
Stop bit: <input type="text" value="2 bits"/>	

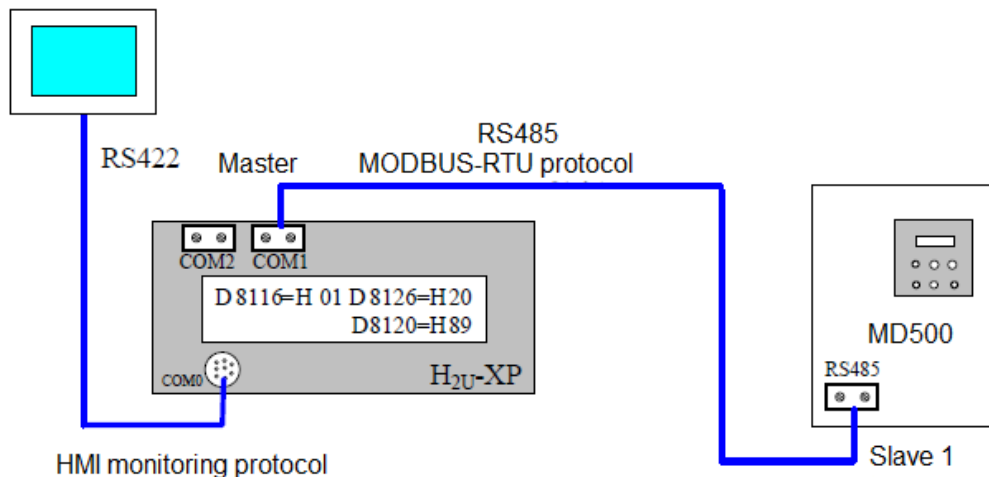
b) AutoShop software configuration:

Open the AutoShop software: File>New Project. The PLC configuration is as follows:



### 5.7.6 Communication programming method between PLC and MD500 series inverter

MD series inverters have built-in MODBUS-RTU slave station protocol. When data exchange between PLC and MDI is required, MODBUS protocol must be used for communication. For the following typical systems:



When programming, configure the PLC's COM1 port as the MODBUS-RTU master station, 9600bps, 8N2, so as to match the factory default values of the MD series inverter, and reduce the trouble of setting the inverter related function codes; the PLC completes the data required by the system exchange work, that is, read and write commands to the slave (the inverter defaults to address #1) to start and stop the inverter, set the frequency, read the operating parameters, etc., or send the HMI setting data to MDI.

Set the COM0 port of the PLC as the (default) HMI monitoring protocol, so that the PLC can communicate with the HMI in the running or stopped state, which is convenient for programming download; (COM0 port: protocol D8116= H01 ---- download and HMI monitoring protocol COM1 port: protocol D8126= H20; format D8120= H89; ----MODBUS master station, 9600bps, 8N2)

The communication protocol of the inverter is MODBUS slave station, and its default address is #1, 9600bps, 8N2. After initializing MD500, it will be this setting, and passively respond to external control (set FP-01=1 on the inverter panel, Press the ENT key to confirm, you can restore the default value)

Pay attention to when programming:

When accessing MD series inverter communication, each communication frame can only read or write

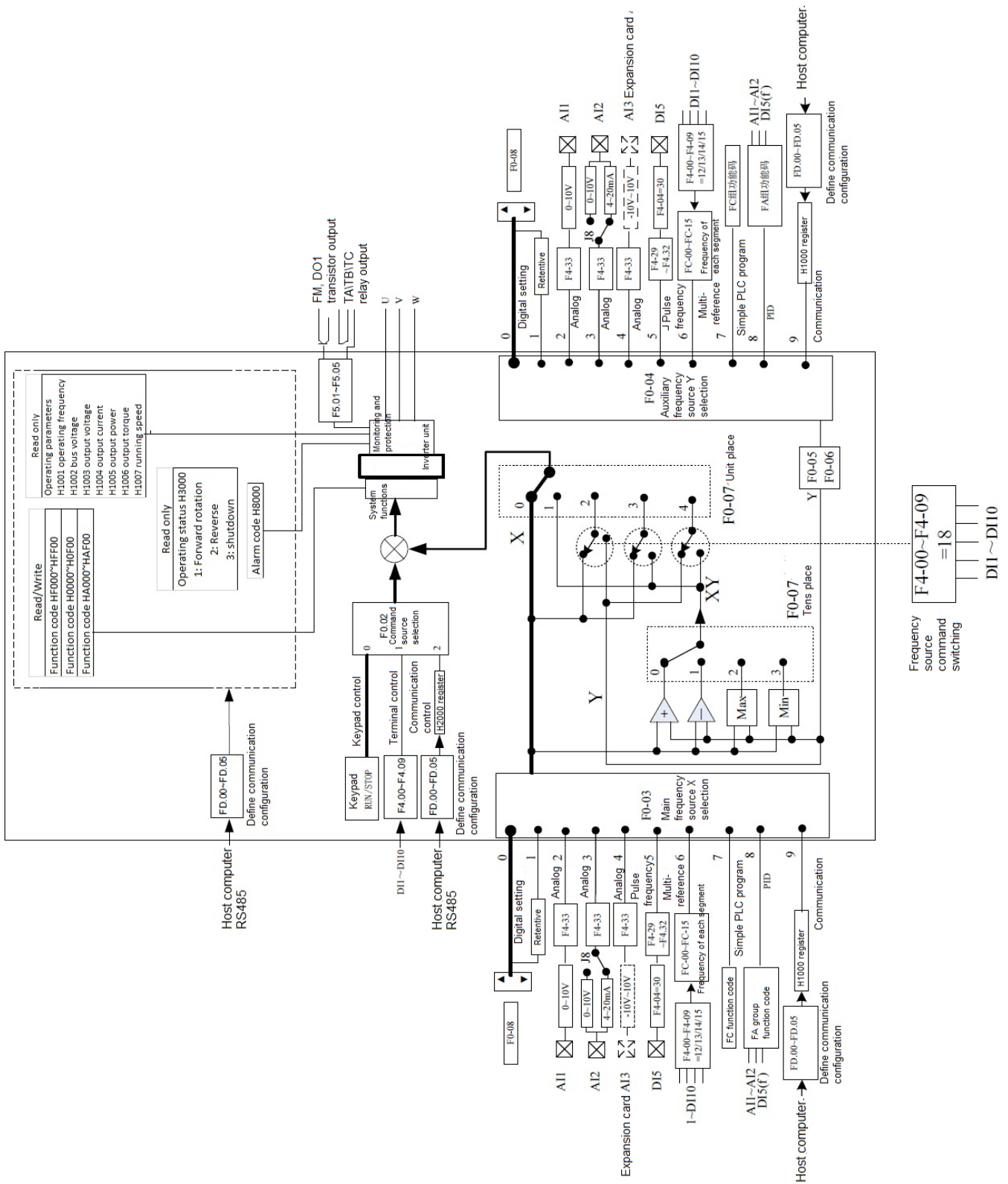
one parameter, and batch reading and writing of multiple variables with consecutive addresses is not supported, that is, the number of registers in the communication instruction is required to be  $n=K1$ .

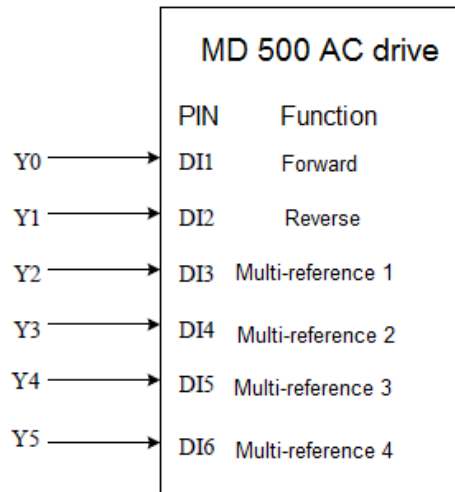
The function code of the inverter (address HF<sup>\*\*\*</sup>) can be read repeatedly, but do not rewrite repeatedly to prevent damage to the internal storage device; other parameters can be read or written repeatedly.

### 5.7.7 Function code setting of MD500 with PLC

The main function code setting related diagram of MD500 inverter operation control is as follows. According to the required working mode, just set the function code indicated in the figure; these function codes can be set manually through the keypad or through serial communication.

The function code settings of other MD series inverters may be different, but they all have basic function code settings such as "start-stop command source" and "frequency command source". The handling method can be used for reference. Please refer to the corresponding manual for terminal start and stop, multi-speed control. For the wiring as shown in the figure below, define the logic input DI pin function as required in the figure:

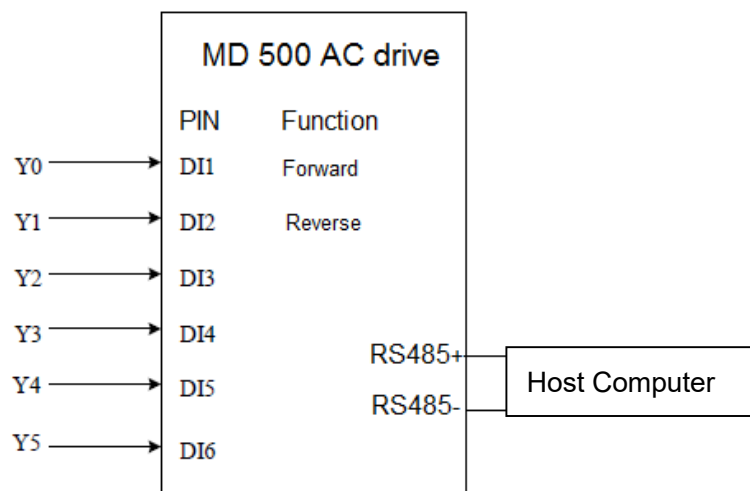




Item	Function Code Setting	Description
Select command source as terminal.	F0-02 = 1	The command source is selected as terminal.
Set terminal command mode to two-line mode.	F4-11 = 0	The FRW/REV terminals retain the default values.
Define the function of access terminal.	DI1:F4-00 = 1	Forward
	DI2:F4-01 = 2	Reversal
Select the frequency source as multi-section speed.	F0-03 = 6	The frequency source is selected as multi-section speed.
Define the function of multi-section speed access terminal.	DI3:F4-02 = 12	Multi-section speed terminal 1
	DI4:F4-03 = 13	Multi-section speed terminal 2
Define the multi-section speed frequency to be used. The default values need to be preset.	FC-00 = 10.00Hz FC-01 = 15.00Hz FC-02 = 20.00Hz FC-03 = 30.00Hz	DI4/DI3 = 0/0: Select multi-section speed 1 (FC-00) DI4/DI3 = 0/1: Select multi-section speed 2 (FC-01) DI4/DI3 = 1/0: Select multi-section speed 3 (FC-02) DI4/DI3 = 1/1: Select multi-section speed 4 (FC-03)
Others	F0-07=0, the default value after initialization, only the main frequency source X is used.	

### Terminal start and stop, communication frequency control

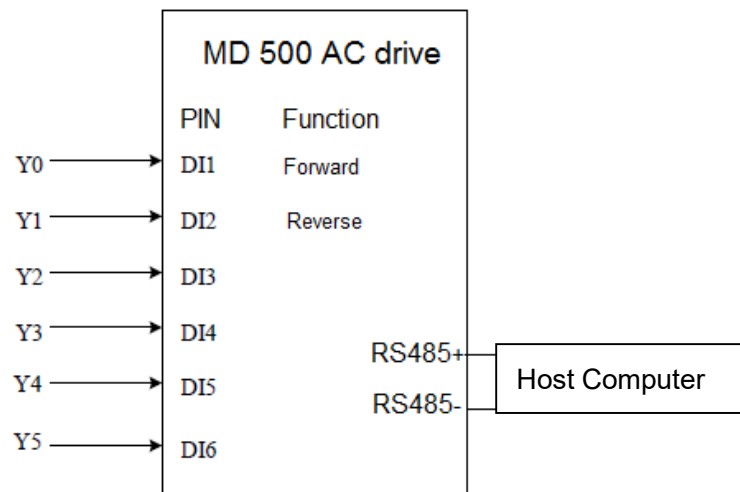
For the wiring as shown in the figure below, define the functions of two of the logic input DI pins as required in the figure, using the inverter communication port and its Modbus-RTU slave protocol:



Item	Function Code Setting	Description
Select command source as terminal.	F0-02 = 1	The command source is selected as terminal.
Set terminal command mode to two-line mode.	F4-11 = 0	The FRW/REV terminals retain the default values.
Define the function of access terminal.	DI1:F4-00 = 1	Forward
	DI2:F4-01 = 2	Reversal
Select the frequency source as serial port communication.	F0-03 = 9	The frequency source is selected as serial port communication.
Configure communication parameters the same as those on the host computer. Default values are recommended.	FD-00 = 5 FD-01 = 0 FD-02 = 1 FD-05 = 1	9600 bps Data format is "no checksum". Local host communication address (default 1) Select standard Modbus protocol.
Running frequency can be changed anytime.	H1000 unit	Write the running frequency (%) to H1000 unit through communication. (-10000~10000) corresponds to (-100.00%~100.00%) of the maximum frequency.
Others	FD-03/FD-04: communication response delay, timeout setting	

### Communication start and stop, communication frequency control

For the wiring as shown in the figure below, the logic input DI pin function is not required, and the communication port of the inverter is used:



Item	Function Code Setting	Description
Select the command source as serial port communication.	F0-02 = 2	The command source is selected as serial port communication.
The frequency source is selected as communication port.	F0-03 = 9	The frequency source is selected as communication port.
Configure communication parameters the same as those on the host computer. Default values are recommended.	FD-00 = 5 FD-01 = 0 FD-02 = 1 FD-05 = 1	9600 bps Data format is "no checksum". (Fixed as 8N2 by default) Local host communication address (default 1) Select standard Modbus protocol.
Start and stop the inverter	H2000 unit	Write the parameters: 1 = Forward 2 = Reversal 3 = Forward motor 4 = Reversal motor 5 = Coast to stop 6 = Slow-down stop 7 = Reset upon failure
Running frequency can be changed anytime.	H1000 unit	Write the running frequency (%) to H1000 unit through communication. (-10000~10000) corresponds to (-100.00%~100.00%) of the maximum frequency.
Others	FD-03/FD-04: communication response delay, timeout setting	

Communication reads the operating parameters of the inverter and modifies the function code.

As long as the host computer adopts the MODBUS protocol and adopts the same communication format configuration as the MD500 AC drive, it can communicate in real time to access the inverter, read the parameters of the inverter, and even modify the function code.

AC drive parameter address:

Parameter Address	Description	Parameter Address	Description
1000H	Communication setting value (decimal): -10000 to 10000	1010H	PID setting
1001H	Running frequency	1011H	PID feedback
1002H	Bus voltage	1012H	PLC process
1003H	Output voltage	1013H	Pulse input frequency, unit: 0.01kHz
1004H	Output current	1014H	Feedback speed, unit 0.1Hz
1005H	Output power	1015H	Remaining running time
1006H	Output torque	1016H	AI1 voltage before correction
1007H	Running speed	1017H	AI2 voltage before correction
1008H	DI input flag	1018H	AI3 voltage before correction
1009H	DO output flag	1019H	Linear speed
100AH	AI1 voltage	101AH	Current power-on time
100BH	AI2 voltage	101BH	Current running time
100CH	AI3 voltage	101CH	Pulse input frequency, unit 1Hz
100DH	Counting value input	101DH	Communication setting value
100EH	Length value input	101EH	Actual feedback speed
100FH	Load speed	101FH	Main frequency X display
		1020H	Auxiliary frequency Y display

The communication setting value is the percentage: 10,000 correspond to 100.00%, -10000 correspond to -100.00%.

1. Control commands input to AC drive (write-only):

Command Word Address	Command Description
2000H	0001: Forward running
	0002: Reverse running
	0003: Forward jog
	0004: Reverse jog
	0005: Coast to stop
	0006: Decelerate to stop
	0007: Fault reset

2. Read AC drive status (read-only):

Status Word Address	Status Word Description
3000	0001: Forward RUN
	0002: Reverse RUN
	0003: Stop

3. AC drive fault alarm code (read only):

AC drive fault address	AC drive fault information		
8000	0000: No fault 0001: Reserved 0002: Overcurrent during acceleration 0003: Overcurrent during deceleration 0004: Over-current at constant speed 0005: Overvoltage during acceleration 0006: Overvoltage during deceleration 0007: Overvoltage at constant speed 0008: Buffer resistance overloaded 0009: Undervoltage 000A: AC drive overloaded 000B: Motor overloaded 000C: Phase loss at input side 000D: Phase loss at output side 000E: Heatsink overheated 000F: External device fault 0010: Communication fault 0011: Contactor fault	0012: Current detection fault 0013: Motor auto-tuning fault 0014: Encoder/PG Fault 0015: Parameter reading and writing abnormality 0016: AC drive hardware abnormality 0017: Motor short circuited to ground 0018: Reserved 0019: Reserved 001A: Running time reached 001B: User defined fault 1 001C: User defined fault 2 001D: Power-on time reached 001E: Load becoming 0 001F: PID feedback loss during running 0028: Rapid current limit timeout 0029: Fault of motor switchover during running 002A: Too large speed deviation 002B: Motor over-speed 002D: Motor overheated 005A: Incorrect setting of encoder pulses per revolution 005B: Not connected with encoder 005C: Initial position error 005E: Speed feedback error	



MD500 function parameter data:

The function parameter data is an important setting parameters of the inverter. On the basis that MD380 only has F group function parameters, MD500 adds A0-AF group function parameters as follows:

MD500 function parameter data	Group F (read and write)	F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FB, FC, FD, FE, FF
	Group A (read and write)	A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, AA, AB, AC, AD, AE, AF

The function parameter data communication address is defined as follows:

1) When reading function codes for communication

For F0~FF, A0~AF group function code data, the higher sixteen bits of the communication address are directly the function group number, and the lower sixteen bits are directly the serial number of the function code in the function group, for example:

F0-16 function parameter, its communication address is F010H, where F0H represents the function parameter of group F0, and 10H represents the hexadecimal data format of the serial number 16 of the function code in the function group.

AC-08 function parameter, its communication address is AC08H, where ACH represents the AC group function parameter, 08H represents the function code in the hexadecimal data format of the serial number 8 in the function group .

2) When writing the function code data for communication, for F0 ~FF group function code data, the higher sixteen bits of the communication address, according to whether it is written into EEPROM, can be divided into 00~0F or F0~FF, the lower sixteen bits are directly the serial number of the function code in the function group, for example:

Write function parameter F0-16

When writing to EEPROM is not required, its communication address is 0010H

When writing to EEPROM, its communication address is F010H

For the function code data of A0~AF group, the higher sixteen bits of the communication address can be divided into 40~4F or A0~AF according to whether it needs to be written into the EEPROM. The lower sixteen bits are directly the serial number of the function code in the function group, for example, as follows :

Write function parameters AC-08

When there is no need to write EEPROM, its communication address is 4C08H

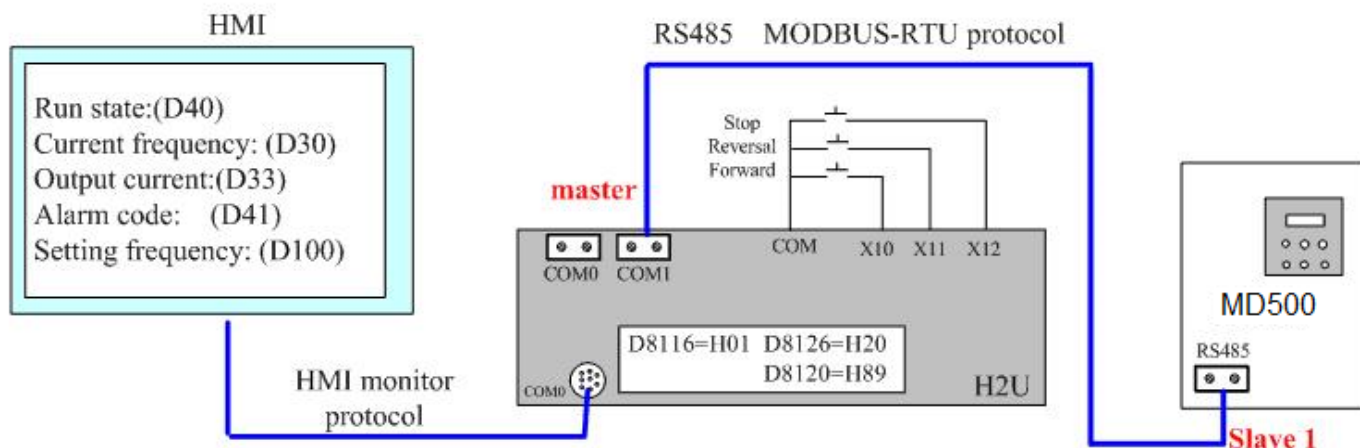
When writing to EEPROM, its communication address is AC08H

When reading or modifying the inverter function code, the "register address" refers to the "function code" number. If you want to read the F0-01 function code, the "register address" is HF001, expressed in Hex format, where the higher byte is function code group number. The lower byte is the index number in the function code group. Note that the index number must have HEX format. For example, to read the function code of FB-29, the "register address" is HFB1D, and so on.

The function code of the inverter has its modification attribute restrictions. Some function codes can be modified at any time, some can only be modified in the stop state, and the range of some modified values is restricted by other related function codes, so please pay attention.

### 5.7.8 PLC program to modify MD500 function code

H2U-MDI is required to be connected by communication, and H2U controls the start, stop, forward, reverse and operating frequency of the inverter, and read the operating parameters of the inverter in real time. For example, the following system:



#### 1) The functional requirements are as follows:

H2U will automatically set the command source of the inverter (F0-02=2) to the serial port every time it is powered on. After the modification is successful, no modification command will be sent; it has nothing to do with the power-on sequence of the H2U and MD500 AC drive.

Press X10 to make the inverter run forward; press X11 to make the inverter run in reverse; press X12 to stop the inverter. Use the frequency value of D100 as the operating frequency of the inverter and send it to the drive in a loop. Read the current operating status, operating frequency, output current, power and other parameters of the drive cyclically

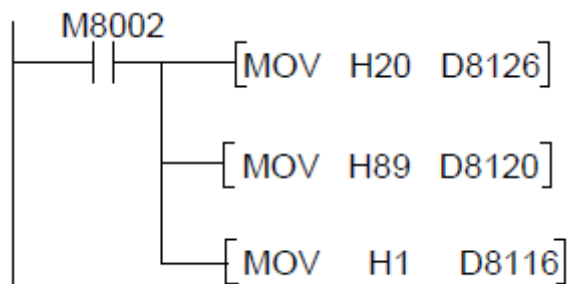
#### 2) Explanation of main points of PLC programming:

Set D8116=H01, set COM0 port as download and HMI monitoring protocol;

Set D8126=H20, D8120=H89 to set COM1 port to MODBUS-RTU protocol, 9600bps, 8N2, and then use MODBUS instruction or RS extended instruction to program the operation of COM1 port.

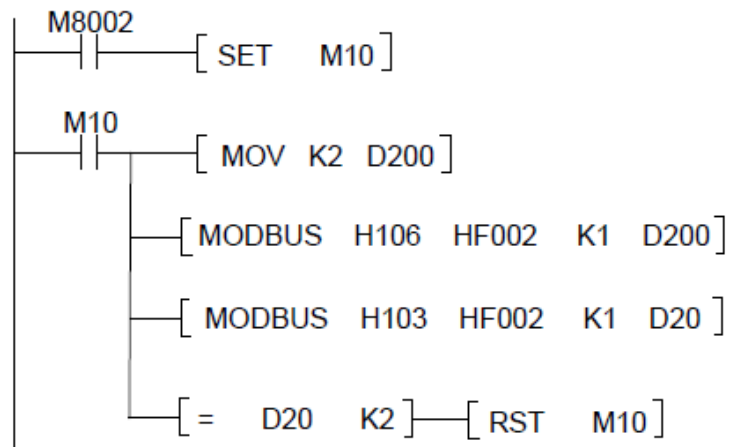
##### a) Examples for setting MODBUS protocol (preferably placed in the first line of the PLC program)

Initialize COM0 as download protocol; COM1 serial port as MODBUS-RTU master station protocol 9600bps, 8N2, RS command is MODBUS command mode.

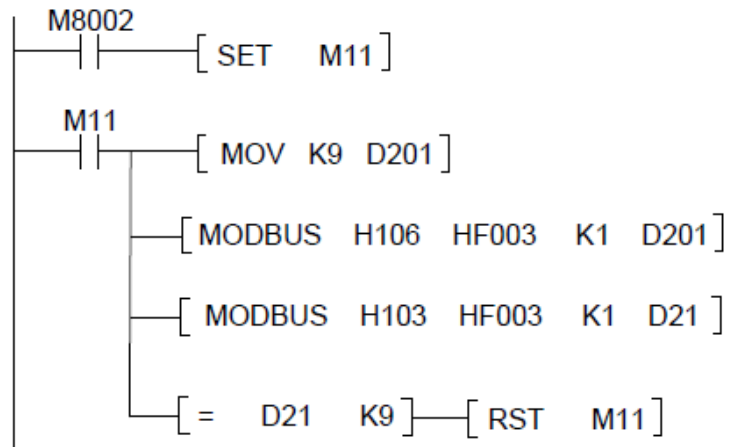


## b) Examples of read operation statements for function codes:

Modify the "command source" function code F0-02, change it to "communication start and stop", that is, make F0-02=2:

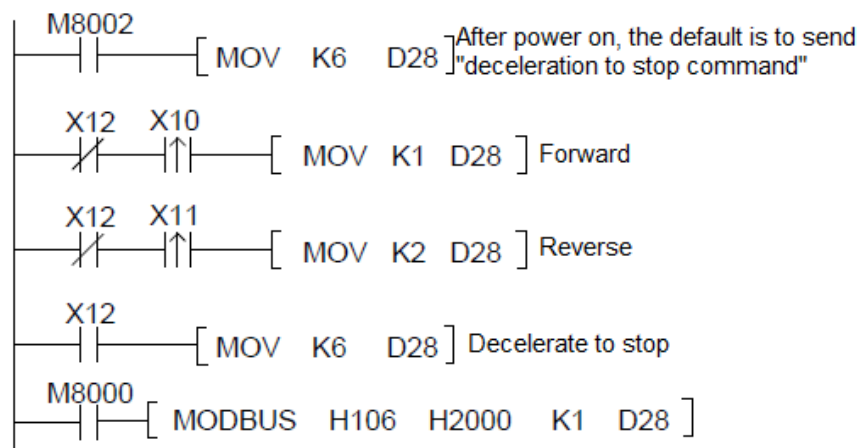


Modify the "frequency source" function code F0-03, change it to "communication setting", that is, set F0-03=9:



## c) In response to the key command, send the inverter forward running, reverse running or stop command, the stop key has priority:

Running commands: X10 - Forward, X11 - Reverses, X12 – Stop



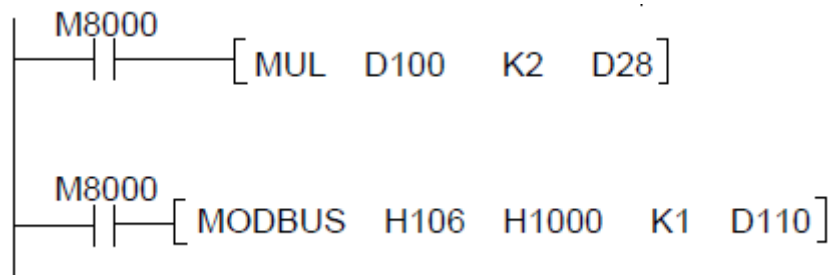
Here, the command words in response to the three operations are processed and sent by the same MODBUS or RS command.

d) Change the operating frequency of the inverter:

The frequency command issued to the inverter is not the data with the resolution of 0.01Hz, but the percentage value relative to the "maximum frequency". K10000 is the full scale and needs to be converted before sending. For example, the maximum frequency of the inverter is 50.00Hz, hope to run at 40.00Hz, the data to be sent is  $40.00 \times K10000 / 50.00 = K8000$ . In this example, K10000/K5000 are directly replaced by K2. If the maximum frequency is not 50.00Hz in actual programming, it is best to use instructions to calculate, and the instructions are sent in cycles:

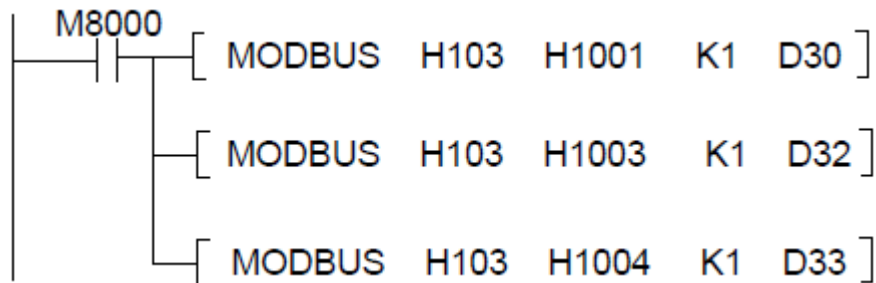
D100 is the given frequency ( $\times 0.01\text{Hz}$ ), pay attention to the relative value of the maximum frequency (F0-04=50.00Hz) before sending, the full value is K10000;

$$D110 = D100 \times K10000 / K5000$$



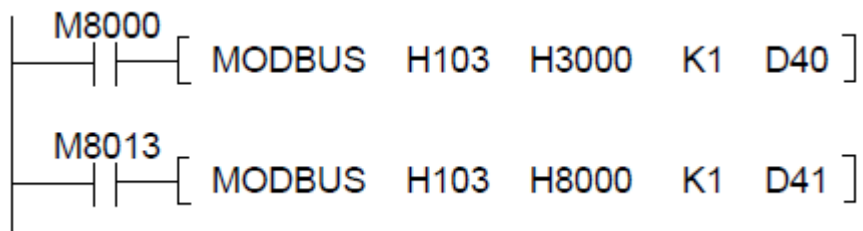
e) Examples of cyclic reading operation of operating parameters:

Read the inverter status: D30 is the operating frequency; D32 is the output voltage; D33 is the output current



Read inverter running status and alarm code: D40=(1=forward running; 2=reverse running; 3=stop)

D41=alarm code



## 5.9 CAN communication command description

### 5.9.1 Introduction to CANlink3.0 Application

H1U/H2U series PLC has CAN communication function, and CAN communication card is required on the hardware. On the PLC main module, remote module access commands FROM/TO is supported, and the CANlink network function is also supported.

CANlink is a network protocol based on CAN bus developed by Inovance. This protocol is an open protocol. Devices that support this protocol can be connected to the CANlink network.

In user programming, the H1U/H2U PLC main module adopts FROM/TO instructions or network configuration methods to access various H2U series remote expansion modules, IS series servo drives, MD series inverters and other intelligent equipment.

For H1U products, pay special attention. You must manually preset D8199 to "7", that is, set the configuration word of the connected expansion card type to "CAN expansion card", otherwise CAN may not communicate normally.

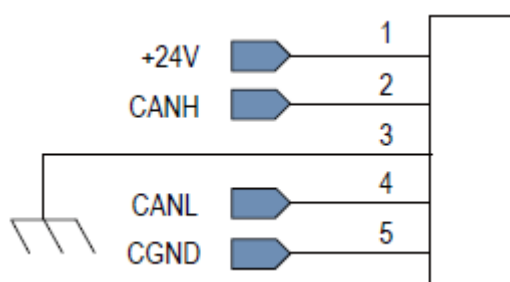
#### 1) Network topology



CANlink network topology diagram

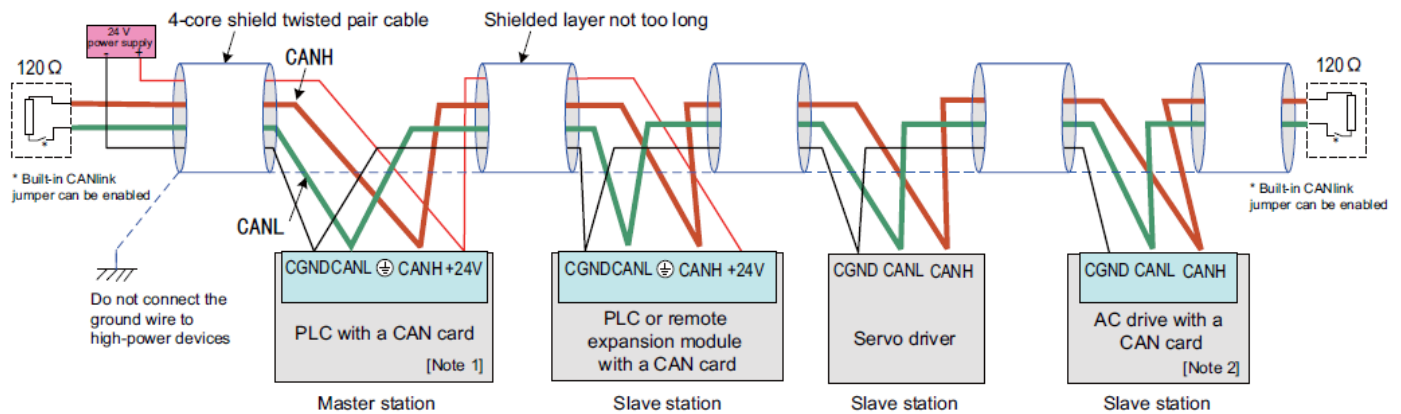
### 5.9.2 Master-Slave network

- H1U/H2U CAN expansion card interface definition



Pin no.	Signal	Description
1	+24 V	External power supply +24 Vdc
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 bus. An external +24Vdc power supply must be provided between pin 1 (+24Vdc) 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.

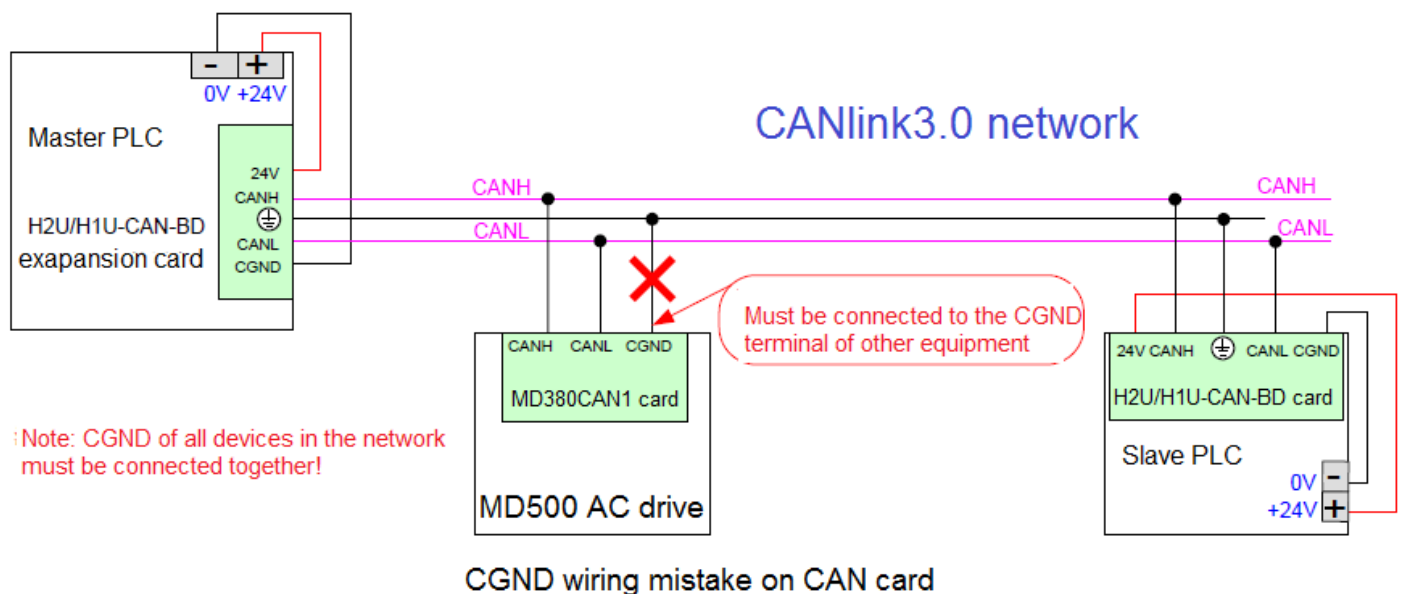
[Note 2] Select a CAN card based on the AC drive type.

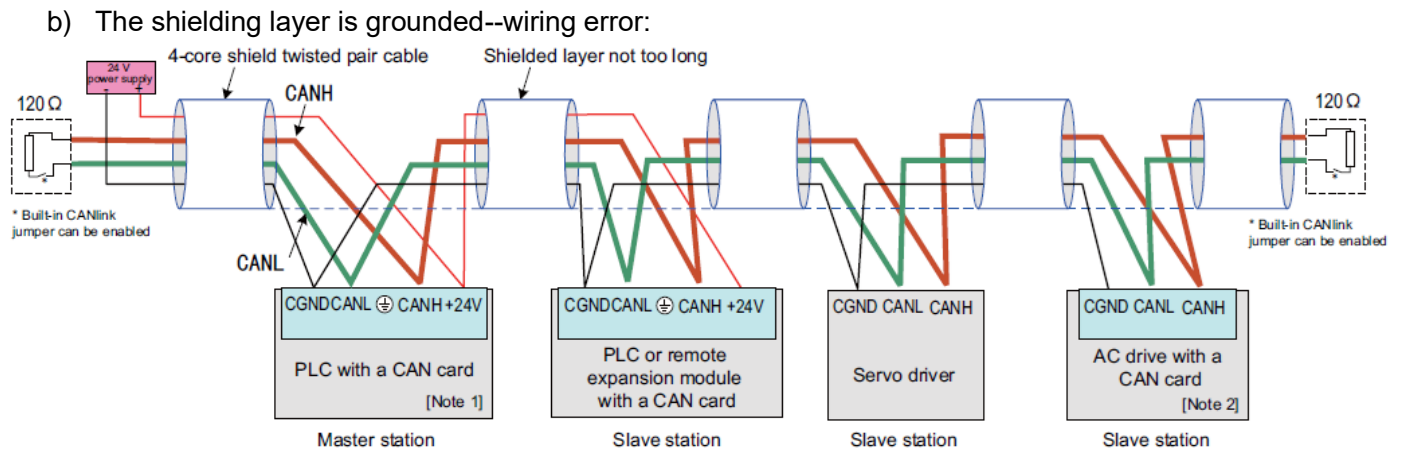
### Wiring diagram of a CAN network formed by multiple devices

Note: CAN communication cable requires shielded cable, recommended model: RVVP 2×2×0.5.

2) Please pay attention to incorrect signal connection: the following incorrect connection methods can easily cause damage to the communication interface or abnormal operation.

a) When the CAN card of each device adopts 24V self-power supply, the CGND wiring on the CAN card is wrong:





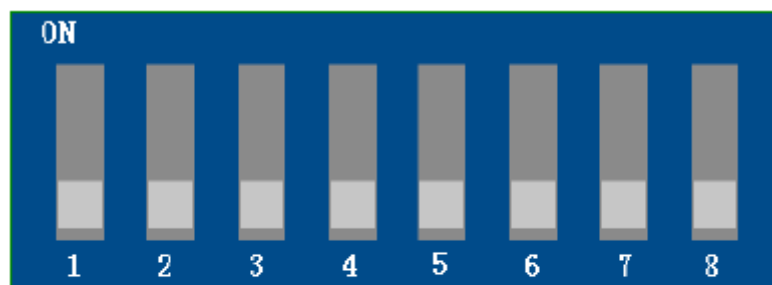
### Wiring error-shield grounding

#### 3) Baud rate, communication length and cable requirements

Estimate the total length of the communication cable and evaluate the cable quality. The maximum communication distance and the number of stations at each baud rate are shown in the figure below. The following data collection conditions are that the cable must be a twisted pair with a metal shield. The cross-sectional area meets the minimum wire diameter, all common ends are connected (cannot be connected with the power ground), and all stations are reliably grounded.

Baud Rate	Distance (m)	Minimum Cable Size (mm <sup>2</sup> )	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

DIP switch on the CANlink expansion card. The switch is defined as follows:



CANlink DIP switch

Table: CANlink DIP switch definition

DIP switch	Signal	Description
1	Address pin A1	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: A6A5A4A3A2A1. For example, if A1 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.
2	Address pin A2	
3	Address pin A3	
4	Address pin A4	
5	Address pin A5	
6	Address pin A6	
7	Baud rate	OFF: high speed mode, baud rate 500Kbps, ON: low speed mode, baud rate 100Kbps
8	Termination resistor	If the DIP switch is ON, it means the termination resistor of 120 ohm is connected, otherwise it is disconnected.

Changing the DIP switch in the PLC running state can make the setting of termination resistor take effect in time, but the baud rate and address cannot take effect immediately, and the system needs to restart to use the new setting parameters.

Note: When M8284 and M8285 are 0, the CAN address and CAN baud rate set by the DIP switch are valid respectively. If you need to set other baud rates by software, please refer to the setting instructions of M8285, D8285, and D8286.

### 5.9.3 Heartbeat attributes

The network heartbeat monitors the operation status of the master/slave station. All slave stations send heartbeats to the master station at this time interval. The master station monitors the status of each station in the network (offline, online) through the heartbeat mechanism, and the slave stations monitor the master station through the heartbeat mechanism of the master station.

Station status. (It is recommended to set the time greater than 200ms).

[Note] Short network heartbeat will increase the load status of the entire network; the network heartbeat setting range is 10~20000ms, and the network heartbeat is not allowed to occupy more than 10% of the network bandwidth.

### 5.9.4 Offline detection

PLC provides the function of judging the situation of station offline in the network by detecting the value of the register.

D7800 indicates whether the master station status detected by PLC slave station is normal. Only D7800 of slave station PLC has meaning.

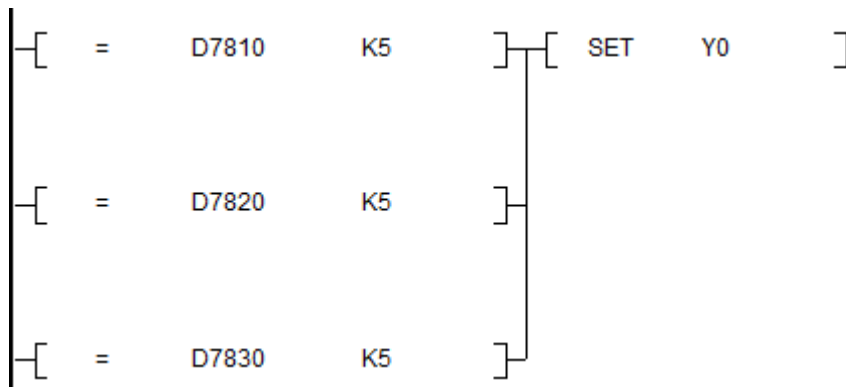
D7801~D7863 are the registers with which the master station monitors online slave station status. The register corresponding to the master station number is meaningless.



The register values for offline detection are defined as follows:

Register	Definition	Offline detection register value		Remarks									
D7800	Master station status			The slave station monitors the master station status.									
D7801	Status of station 1	<table border="1"> <thead> <tr> <th>D78xx Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>The station does not exist or the register is not defined.</td> </tr> <tr> <td>1</td> <td>Used by the master station only. The station is configured, but the master station cannot detect it.</td> </tr> <tr> <td>2</td> <td>The station works normally.</td> </tr> <tr> <td>5</td> <td>The station is disconnected.</td> </tr> </tbody> </table>	D78xx Value	Definition	0	The station does not exist or the register is not defined.	1	Used by the master station only. The station is configured, but the master station cannot detect it.	2	The station works normally.	5	The station is disconnected.	The master station monitors the status of all slave stations. The register corresponding to the master station number is meaningless. If the master station number is 7, D7807 is meaningless.
D78xx Value	Definition												
0	The station does not exist or the register is not defined.												
1	Used by the master station only. The station is configured, but the master station cannot detect it.												
2	The station works normally.												
5	The station is disconnected.												
D7802	Status of station 2												
D7803	Status of station 3												
...	...												
D7861	Status of station 61												
D7862	Status of station 62												
D7863	Status of station 63												

The master station can use the following procedure to detect whether there is a slave station down (in the example, when any one of the No. 10, No. 20, and No. 30 slave stations is down, Y0 is turned on).



CANlink3.0 has the feature of automatic reconnection after the station is dropped, that is, after the slave station is dropped, it will automatically detect the network, and automatically connect to the network after the communication cable is normal. The master station will automatically reconnect after detecting the dropped station. The corresponding register is restored. After the inverter and the servo are disconnected from the station and reconnect to the network, the error words on the panel must be manually cleared. Please refer to the inverter and servo user manual for other functions.

Note: PLC will correct the value of the corresponding register (D7800-D7863) according to the actual status, do not try to modify it manually.

### 5.9.5 Use of CANlink3.0

Special variable table of related information of CAN communication expansion card (VER 3.00)

M	M Element Definition	D Element	D Element Definition
M8280	Protocol flag *1 0: CANlink VER3.00 protocol 1: CANlink VER1.00 protocol	D8280	CANlink protocol version (300)
M8281	0: The expansion card is not recognized, and the CANlink function cannot be used at this time 1: Expansion card detected (including CANlink card)	D8281	Reserved
M8282	Reserved	D8282	Current heartbeat time (read only, unit in ms)

Table: Special variable table of CAN communication address and baud rate related settings

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	20: 20 Kbps 50: 50 Kbps 100: 100 Kbps 125: 125 Kbps 250: 250 Kbps 500: 500 Kbps 800: 800 Kbps 1000: 1 Mbps Other value, 500Kbps

Table: Special variable table of CAN communication status related information

M Element	M Element Definition	D Element	D Element Definition
M8287~M8289	Reserved	D8287~D8289	Reserved
M8290	CANlink start/stop element	D8290	Reserved
M8291	CANlink Synchronous Send trigger element	D8291	Reserved
M8292~M8295	Reserved	D8292~D8295	Reserved

M Element	M Element Definition	D Element	D Element Definition
M8296	The device address is wrong, and the CANlink address setting is wrong.	D8296	Reserved
M8297~M8298	Reserved	D8297~D8298	Reserved
M8299	CAN receiving timeout *2 refers to the return timeout of FROM and TO commands	D8299	CAN receiving timeout setting (ms) *3
M8300~M8306	Reserved	D8300~D8306	Reserved
M8307	Reserved	D8307	Command frame error register, the display format of the master station is: error slave station number *100+ error code Reserved
M83088	Reserved	D8308	

Table: Fault information (add D register value description)

Fault query register	Error code	Error description
D8066 to D8067	6380	Send timeout
	6381	Receive timeout
	6382	CAN send busy
	6383	CAN receive busy

\*1: The user needs to restart or STOP->RUN after setting to take effect. If there is one or more non-XP platform modules in the remote modules connected to the PLC, M8280 must be set to ON (switch to CANlink1.0 protocol), and restart again to make the module work normally. (The nameplate of the remote expansion module indicates whether it is an XP platform module)

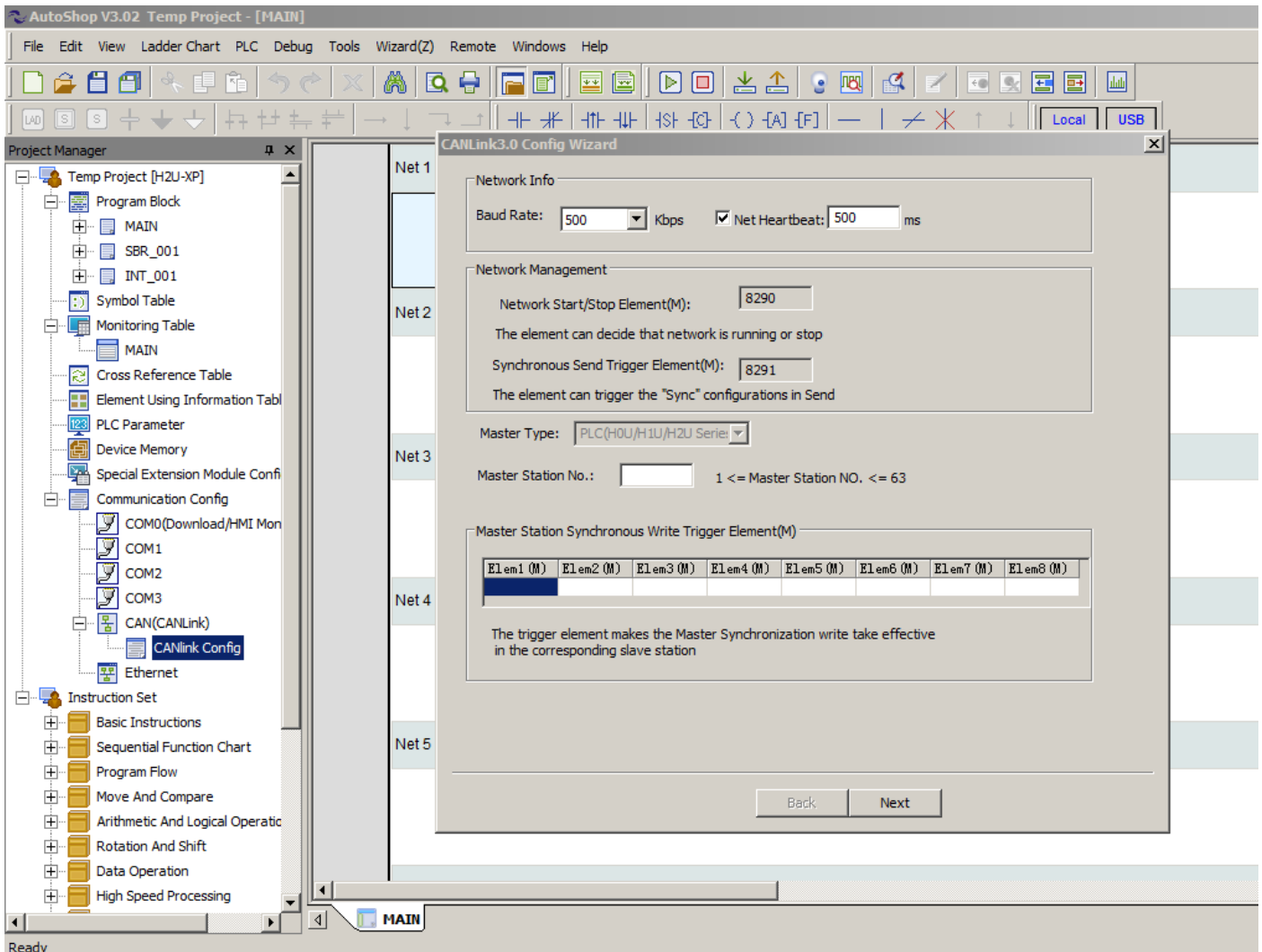
\*2: After setting, the user needs to reset it.

\*3: The receiving waiting time of FROM and TO, if the setting is less than 5, the default is 5ms, if the setting is greater than 1000, the default is 1000ms. It means to calculate the waiting time from sending the command. For FROM and TO commands, M8299 will be set if this time is exceeded. It is recommended to decrease the value when the baud rate is high and increase the value when the baud rate is small, otherwise the data will not be received due to the long transmission time. The baud rate becomes smaller, the setting time of D8299 becomes larger, for example;

50K is set to 10; 25K is set to 25, otherwise an abnormality will occur.

### 5.9.6 Software configuration CANlink3.0 master-slave communication configuration method

When programming the communication application of CANlink3.0, it is not the previous CAN communication command mode, but the "CANlink Config" mode, and the communication properties needs to be configured in advance:



When downloading the user program, download the "CANlink Config" to the PLC at the same time:

**Download** [X]

Option

PLC/Communication Parameter     Use PLC Identifier     Syn Clock

Program     Element comment     No Uploading

Program network comment

Config

Modbus Config     CANLink Config     Ethernet Config

Module Config     CANOpen Config     CAM Config

CAM Formula

Device Memory [ ]

D Element

Data register 1(0-999)    [ 0 ] - [ 999 ]

Data register 2(1000-7999)    [ 1000 ] - [ 7999 ]

M Element

Data Register(0-3071)    [ 0 ] - [ 3071 ]

R Element

Data register(0-32767)    [ 0 ] - [ 32767 ]

Note:If the PLC identifier was set in PLC,when executing the download function,make sure the PLC user program's PLC identifier matches to the PLC identifier in PLC,otherwise the PLC will not execute the user program.

[ Download ]    [ Close ]

**CANLink3.0 Config Wizard**

Network Info

Baud Rate: 500 Kbps  Net Heartbeat: 500 ms

Network Management

Network Start/Stop Element(M): 8290  
The element can decide that network is running or stop

Synchronous Send Trigger Element(M): 8291  
The element can trigger the "Sync" configurations in Send

Master Type: PLC(H0U/H1U/H2U Series)

Master Station No.: 1 1 <= Master Station NO. <= 63

Master Station Synchronous Write Trigger Element(M)

Elem1 (M)	Elem2 (M)	Elem3 (M)	Elem4 (M)	Elem5 (M)	Elem6 (M)	Elem7 (M)	Elem8 (M)
1	2						

The trigger element makes the Master Synchronization write take effective in the corresponding slave station

Back Next

To understand the principle of CANlink3.0 network configuration first, it is helpful to fill in the "CAN network configuration" form correctly:

- 1) In the CANlink3.0 network, there must be only one communication master station, this master station is H2U-XP or H1U-XP series PLC
- 2) In the CANlink3.0 network, there must be one or more communication slave stations, which can be remote expansion modules such as H2U series IO/AI/AO/AM/PT/TC, and MD series inverter with CANlink3.0 interface card, IS series servo driver with CANlink3.0 interface, it can be H2U-XP/H1U-XP series PLC, or it can be a slave device developed by users according to CANlink3.0 protocol
- 3) The master and slave devices in the CANlink3.0 network all use the method of actively sending "communication write" data instead of the inquiry response method. For example:
  - The master station wants to send data to the slave station. The realization method is that the master station "writes" the data of the designated register according to the CANlink communication configuration when the trigger conditions are met to the register of the designated slave
  - The master station reading data from the slave station is realized by the way that the slave station

automatically sends data to the host according to the CANlink communication configuration, and "writes" the data communication to the receiving unit of the master station

- Information exchange between slave stations is realized by the way that the slave station automatically sends data to the designated slave station according to the CANlink communication configuration, and "writes" the data communication to the receiving unit of the designated slave station
- The information that a station wants to send to multiple stations is based on the CANlink communication configuration, automatically sending "write operation" data to itself (equivalent to broadcasting), and other stations will selectively receive these data and automatically keep it until the scheduled time. Set the way in the receiving unit
- In order to improve the efficiency of data interaction in network communication, both the master station and the slave station can save the broadcast data sent by other station numbers that they "hear". The station number address of the receiving slave must be set in advance, and the broadcast data from stations other than the station number setting will be ignored

4) Because the CANlink3.0 slave station does not need to be configured, but the H2U-XP/H1U-XP master station PLC transfers the CANlink configuration to the slave station, so in the CANlink3.0 communication configuration item of the master station, there is a configuration for the master station, and there are separate configurations for each slave station. These configuration items for the slave station are forwarded by the CANlink master station through the configuration frame.

5) Every time the master station starts to run, it will send a configuration frame to the CANlink slave station and arrange the "communication task list" of each slave station. Once it is running, each slave station will actively send data according to the task list.

6) CANlink3.0 configuration items include: the register address of the data to be sent, the address of the target receiving slave station, the number of data, the receiving register address, the time interval of communication sending, the trigger condition, etc., these are necessary for general communication instructions.

- Different from general communication, these "communication write" operations do not require a response whether the operation is successful or not by default.
- For communication applications that require simultaneous action responses from multiple slaves, such as servo-driven multi-axis synchronous control and position-controlled high-speed motion applications, the master station configuration needs to be filled in the configuration options of "synchronous writing". In actual operation, the master station writes data to each slave station separately, and then sends a broadcast command frame that takes effect at the same time, so that each slave station operates at the same time.

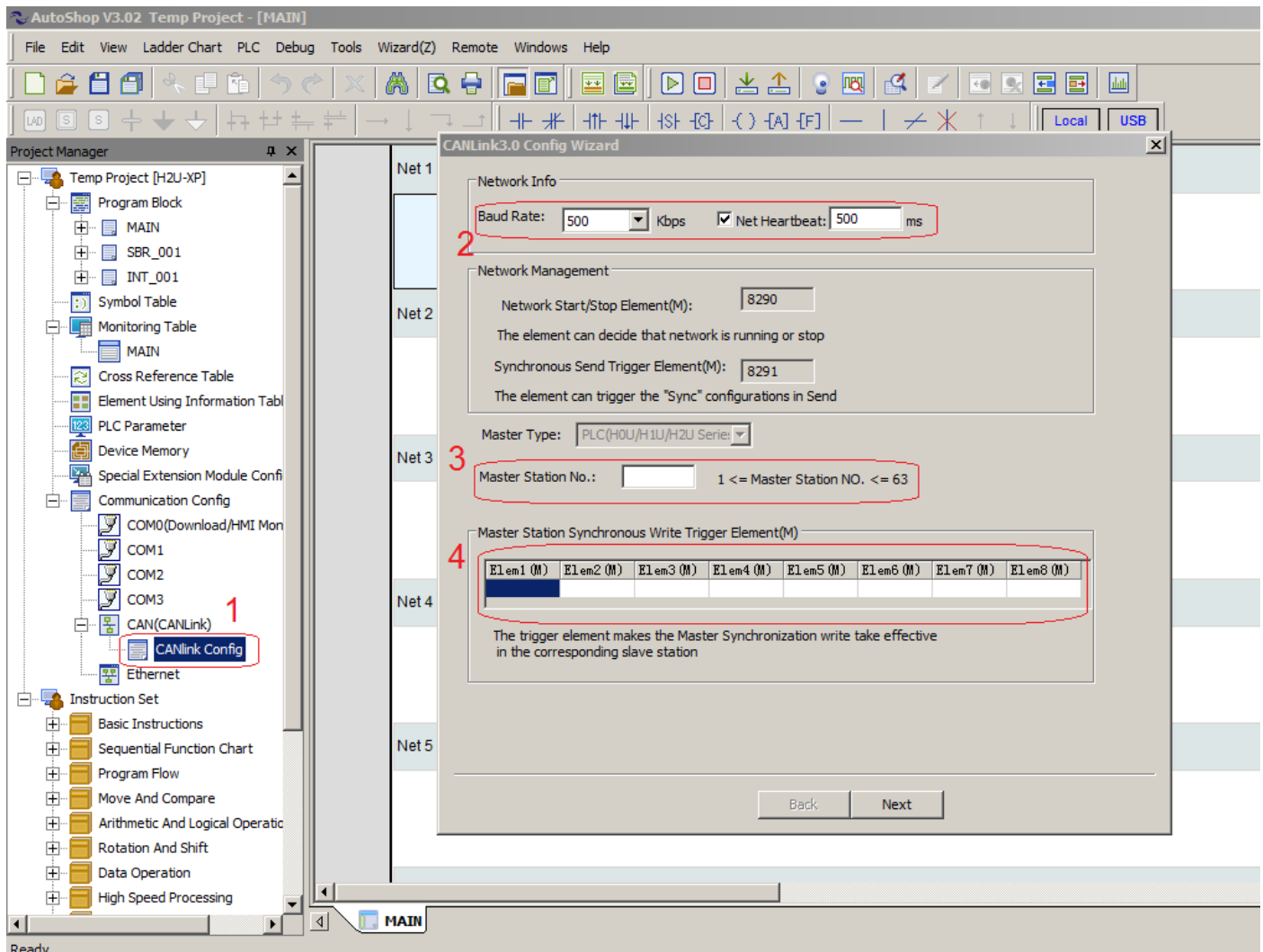
### 5.9.7 How to set CAN network configuration table

After the project is established, double-click "CAN Config" in "Project Management" to open the CANlink configuration table. When entering the configuration for the first time, directly enter the "CANlink3.0 configuration" window.

#### 1) Configuration Wizard:

The master station configuration wizard mainly completes the network configuration, master station

configuration, master station synchronization trigger component management, and slave station addition functions. The wizard page is shown in the figure:



- **Baud rate (required):**

There are eight baud rates: 20K, 50K, 100K, 125K, 250K, 500K, 800K, and 1M to meet the needs of different applications. You can select the required baud rate through the pull-down list, and download the configuration to the master station to take effect ( This parameter is only valid for the master station, and the slave station still needs to be modified manually), the user can select the appropriate baud rate according to the total load condition and the actual communication distance.

- **Network Heartbeat (optional):**

All stations send heartbeats to the master station at this time interval, the master station monitors the status of each station in the network (offline, online) through the heartbeat mechanism, and the slave station monitors the status of the master station through the master station heartbeat. (It is recommended to set the time greater than 200ms) If you remove the tick before the network heartbeat, the network heartbeat function will be cancelled and the network cannot be monitored.

[Note] Short network heartbeat will increase the load status of the entire network; the network heartbeat setting



range is 10~20000ms, and the network heartbeat is not allowed to occupy more than 10% of the network bandwidth.

- **Master Station Number (Required):**

The master station number is the station number of the master station in the entire network, and the station number of the PLC master station that downloads the configuration (here you need to fill in the station number of the PLC as the master station), it is only for configuration, and you cannot change the station here. If the filled station number is inconsistent with the actual station number, even if the configuration is downloaded to the PLC, the PLC will not execute it, but will treat it as an invalid configuration. For example, if the master station number in the background configuration is filled in as 7, then the configuration cannot be downloaded to other PLCs except station No. 7. It will only be effective if it is downloaded to station No. 7 PLC, and then No. 7 station will send the configuration to other stations. The entire CANlink network configuration is downloaded to the master station and distributed to the slave stations. The software can monitor and manage the entire network through the master station.

- Trigger element for master station configuration synchronous writing (optional)

The trigger element configured by the master station synchronously, if the trigger element (M) is set, the corresponding trigger configuration is valid, and it is automatically reset after sending.

- Add, modify or delete slaves

**CANLink3.0 Config Wizard** [X]

Station Info

Slave Station Type:

Slave Station NO.:  1 ≤ NO. ≤ 63

Status Code Register(D):  D0 ≤ Value ≤ D7000, the status information of the station

Start/Stop Element(M):  M0 ≤ Value ≤ M3071, decide the slave status(run or stop)

Slave	Station Type	Status Register(D)	Start/Stop Elem(M)
10	PLC(H0U/H1U/H2U Series)	3010	3010
15	TCM	3015	3015
20	IS(Servo)	3020	3020
30	MD(Inverter)	3030	3030
31	NTCM	3031	3031

- Delete

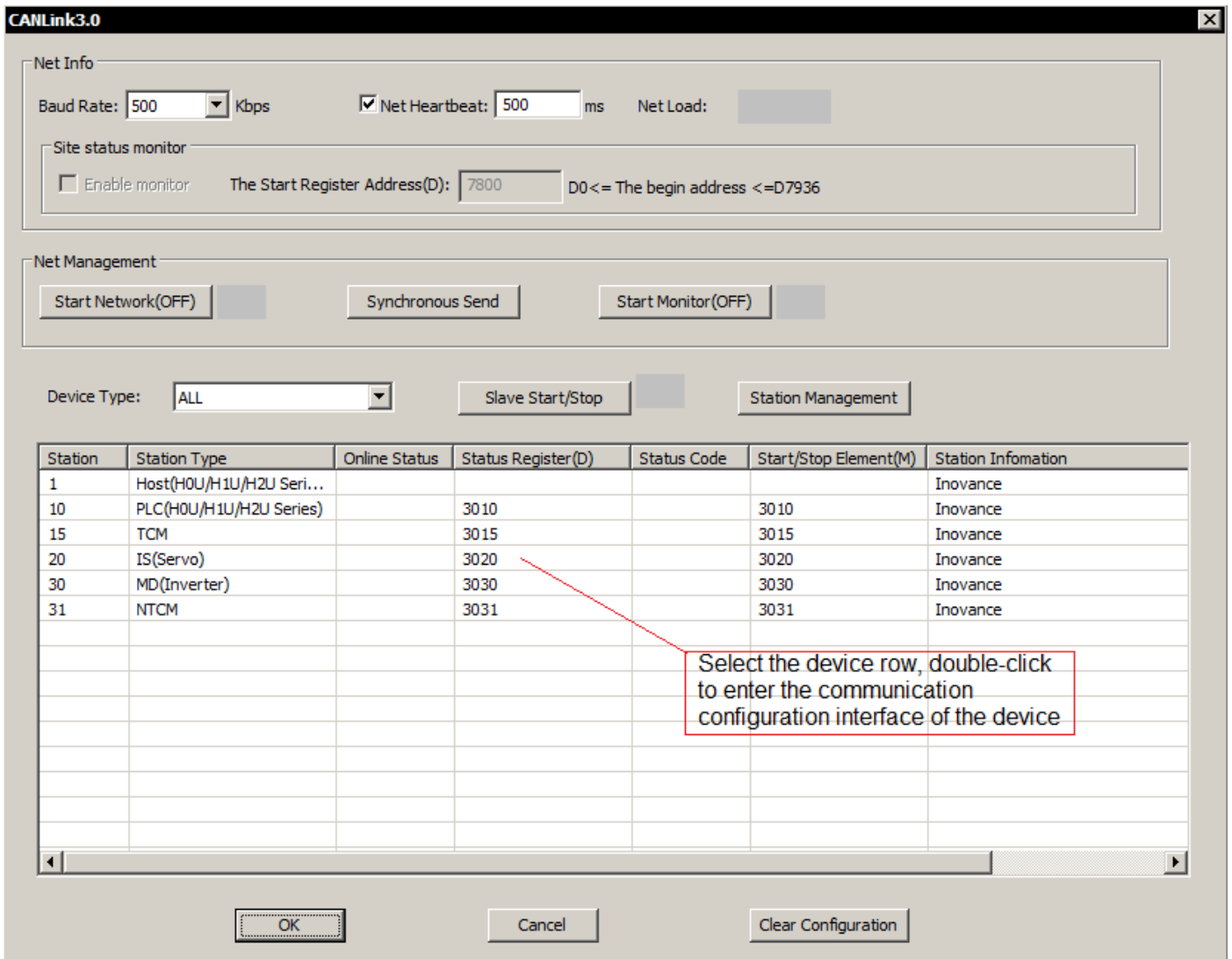
After selecting the station in the list, click the "Delete" button, and click "OK" at the prompt "Delete" to delete (you can select multiple stations at the same time)

- Modify

Select a single slave station in the list, modify the corresponding information in the "station information", and click the "modify" button (the station type cannot be modified)

- Finish

In the configuration wizard window, click the "Finish" button to save the modifications in the wizard and exit; click the "X" in the upper right corner of the window to cancel the modification operation and exit



## 2) Station configuration

Double-click any station in the "Main Interface" to open the station communication configuration window.

The communication configuration includes three parts:

Synchronous configuration (only master station), sending configuration, receiving configuration.

- Master station synchronization configuration

When the trigger condition M is set, the corresponding master station synchronous sending configuration takes effect. Select different "trigger conditions (M)" to view, add, modify, and delete the master station synchronization configuration information corresponding to the trigger component. Synchronous configuration is mainly suitable for occasions that need to start a certain operation at the same time.

In the figure, when the master station's M1=1, the above three configurations will be sent out in sequence. After receiving the configuration, the slave station will store the data in the buffer area. After the last one is sent successfully, the master station will automatically send an effective command, all the slave stations on the network will automatically write the data in the buffer to the corresponding component or function code after receiving the command. In the figure, the No. 10 PLC will write the value of the master station D10 received previously into D10, No. 20 servo will write the value to H200, and the No. 30 inverter writes the value of D30 of

the master station previously received into HF003, that is, all the above values are written at the same time after receiving the effective command.

After the effective command is successfully issued, the master station will automatically reset the trigger element M1. Refer to "Event Trigger" for the rules for using trigger components.

[Note] Trigger condition (M): Each trigger condition is associated with up to 16 configurations. This "trigger condition (M)" can determine whether the associated master station synchronization configuration is valid. The entire network allows up to 8 different trigger conditions (M); Click the trigger element (M) to pull down to switch between different trigger elements.

If you need to configure the servo's 32-bit register synchronously, please divide the data and address into the upper 16 bits and the lower 16 bits in the same trigger element, that is, write two under the same trigger element, one corresponding to the servo 32-bit function code For the low address bit of, one corresponds to the high address bit. If you only write one or divide the two under two different trigger components, the servo will report an error and cannot perform related operations.

**Host Station(1) Config**

Send Configuration | Receive Configuration | Synchronous Write

Trigger Condition(M):

NO.	Send Station		Send Register		Receive Station		Receive Register	
1	1	HOST	100	Dec	10	PLC (HOU/H1	1000	Dec
2	1	HOST	101	Dec	20	IS (Servo)	1112	Hex
3	1	HOST	102	Dec	30	MD (Inverte:	1113	Hex
4	1	HOST	103	Dec	30	MD (Inverte:	F003	Hex
5	1	HOST	104	Dec	30	MD (Inverte:	F004	Hex
6	1	HOST		Dec				
7	1	HOST		Dec				
8	1	HOST		Dec				
9	1	HOST		Dec				
10	1	HOST		Dec				
11	1	HOST		Dec				
12	1	HOST		Dec				
13	1	HOST		Dec				
14	1	HOST		Dec				
15	1	HOST		Dec				
16	1	HOST		Dec				

**Host Station(1) Config**

Send Configuration | Receive Configuration | Synchronous Write

Trigger Condition(M):

NO.	Send Station		Send Register		Receive Station		Receive Register	
1	1	HOST	200	Dec	10	PLC (HOU/HI)	1000	Dec
2	1	HOST	201	Dec	20	IS (Servo)	1112	Hex
3	1	HOST	202	Dec	20	IS (Servo)	1113	Hex
4	1	HOST	203	Dec	30	MD (Inverte:	F003	Hex
5	1	HOST	203	Dec	30	MD (Inverte:	F004	Hex
6	1	HOST		Dec				
7	1	HOST		Dec				
8	1	HOST		Dec				
9	1	HOST		Dec				
10	1	HOST		Dec				
11	1	HOST		Dec				
12	1	HOST		Dec				
13	1	HOST		Dec				
14	1	HOST		Dec				
15	1	HOST		Dec				
16	1	HOST		Dec				

- The master station writes the servo 32-bit register synchronization configuration
- In the above figure, H1112 is the 32-bit function code of the servo. If you want to set the function code, you need to set two, corresponding to its high and low address bits. When M2 is set, the master station writes the values of D101 and D102 into the 32-bit function code H1112 of the servo. When all five items under M2 are sent, the master station will send an effective command to enable each slave station to take effect at the same time. Then M2 is automatically reset; when M3 is set, the master station writes the values of D201 and D202 into the servo H1112, and then take effect, reset M3. Through the drop-down box of the trigger condition, you can pull down to switch between different trigger components.
- The setting of the trigger element is in the wizard, as shown in the figure:

**CANLink3.0 Config Wizard**

Network Info

Baud Rate:  Kbps  Net Heartbeat:  ms

Network Management

Network Start/Stop Element(M):   
The element can decide that network is running or stop

Synchronous Send Trigger Element(M):   
The element can trigger the "Sync" configurations in Send

Master Type:

Master Station No.:  1 <= Master Station NO. <= 63

Master Station Synchronous Write Trigger Element(M)

Elem1 (M)	Elem2 (M)	Elem3 (M)	Elem4 (M)	Elem5 (M)	Elem6 (M)	Elem7 (M)	Elem8 (M)
1	2						

The trigger element makes the Master Synchronization write take effective in the corresponding slave station

Back Next

Synchronous configuration trigger element settings

If only one address is operated under a single trigger element when operating the 32-bit function code, the servo will return an error and the synchronization cannot continue. This error is recorded in D8307 of the master station. For the error information, please refer to "Master station error code and handling".

**Host Station(1) Config**

Send Configuration | Receive Configuration | Synchronous Write

Trigger Condition(M): 2

NO.	Send Station	Send Register	Receive Station	Receive Register
1	1 HOST	100 Dec	10 PLC (HOU/HI)	1000 Dec
2	1 HOST	101 Dec	20 IS (Servo)	1112 Hex
3	1 HOST	102 Dec	30 MD (Inverte:	1113 Hex
4	1 HOST	103 Dec	30 MD (Inverte:	F003 Hex
5	1 HOST	104 Dec	30 MD (Inverte:	F004 Hex
6	1 HOST	Dec		
7	1 HOST	Dec		
8	1 HOST	Dec		
9	1 HOST	Dec		
10	1 HOST	Dec		
11	1 HOST	Dec		
12	1 HOST	Dec		
13	1 HOST	Dec		
14	1 HOST	Dec		
15	1 HOST	Dec		
16	1 HOST	Dec		

Wrong synchronization configuration

Incorrect synchronization configuration

- Synchronous configuration (only master station)

**Host Station(1) Config**

Send Configuration | Receive Configuration | Synchronous Write

Trigger Condition(M): 1

NO.	Send Station	Send Register	Receive Station	Receive Register
1	1 HOST	10 Dec	10 PLC (HOU/HI)	10 Dec
2	1 HOST	20 Dec	20 IS (Servo)	200 Hex
3	1 HOST	30 Dec	30 MD (Inverte:	F003 Hex
4	1 HOST	Dec		
5	1 HOST	Dec		
6	1 HOST	Dec		
7	1 HOST	Dec		
8	1 HOST	Dec		
9	1 HOST	Dec		
10	1 HOST	Dec		
11	1 HOST	Dec		
12	1 HOST	Dec		
13	1 HOST	Dec		
14	1 HOST	Dec		
15	1 HOST	Dec		
16	1 HOST	Dec		

### 3) Send configuration

The sending configuration is mainly used to configure the data sent by this site to other sites. The configuration page is shown in the figure:

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)	100	1	HOST	102	Dec	20	IS (Servo)	200	Hex	1
2	Time (ms)	100	1	HOST	110	Dec	1	HOST	514	Dec	1
3	Event (M)	8291	1	HOST	104	Dec	30	MD (Invert	P003	Hex	1
4			1	HOST		Dec					
5			1	HOST		Dec					
6			1	HOST		Dec					
7			1	HOST		Dec					
8			1	HOST		Dec					
9			1	HOST		Dec					
10			1	HOST		Dec					
11			1	HOST		Dec					
12			1	HOST		Dec					
13			1	HOST		Dec					
14			1	HOST		Dec					
15			1	HOST		Dec					
16			1	HOST		Dec					
Receive	---	---	---	---	---	---	---	---	---	---	---
1	Syn (M)	8291	10	PLC (HOU/H	1000	Dec	1	HOST	1	Dec	2
2	Time (ms)	1000	30	MD (Invert	8000	Hex	1	HOST	10	Dec	2

Send configuration

- Trigger

- Time (ms): Applicable to all devices, this station executes this communication configuration at a fixed interval (trigger condition), the setting range is 1~30000ms.
- Event (M): Applicable to Host and PLC, this communication configuration is executed when the trigger condition (M element) of this station is set, and the same M element is allowed to trigger, and it will automatically reset after sending.
- Synchronization (M): Applicable to all devices, this communication configuration is executed when the master station component M8291 is set.
- Event (ms): For IS, MD, and remote expansion modules (TCM/NTCM), the station detects that the value of the sending register has changed and the prohibition time (trigger condition) is met to execute this communication configuration.

\*1: Prohibited time: the minimum time between two consecutive transmissions of the same configuration.

\*2: The maximum number of single stations allowed in the sending configuration: 256 for the HOST master, 16 for a single slave, and 256 for the total number of slaves.

\*3: Select a configuration, press "Insert", an empty configuration line will be added after this configuration. Similarly, after selecting a configuration and pressing "Delete", this configuration will be deleted; in addition, "copy", "paste", "delete", "line insertion", "line deletion" can be realized through shortcut keys or right-click pop-up menu



- Register

The register value in Host and PLC corresponds to the D component; the register value in IS and MD corresponds to the function code; TCM/NTCM corresponds to the BFM area.

- Number of registers

Indicates the number of continuous D components or function codes sent and received.

- Point-to-multiple configuration

When the sending and receiving stations are the same, it is a point-to-multipoint configuration. This configuration does not specify the receiving station; any station that configures the station number of the sending station to the "receiving configuration" can receive the data sent by the configuration, and the receiving register corresponds to the receiving station D component or function code.

- This site receives

The grey part under the red dividing line is the data sent by other stations to this station, including point-to-point and point-to-multipoint data. Through this part, users can intuitively check which components or function codes of which stations will affect this station.

#### 4) Receive configuration

The receiving configuration is mainly used for this site to receive point-to-many data from other sites, and each site can receive point-to-many data from 8 other sites.

Example: the following configuration

The screenshot displays the 'Host Station(1) Config' window with three tabs: 'Send Configuration', 'Receive Configuration', and 'Synchronous Write'. The 'Receive Configuration' tab is active, showing a table with the following data:

No	Trigger Mode	Trigger	Send Station	Send Register	Receive Station	Receive Register	Length
1	Time (ms)	100	1 HOST	4096 Dec	1 HOST	8291 Dec	1
2	Time (ms)	100	1 HOST	102 Dec	20 IS (Servo)	200 Hex	1
3	Time (ms)	100	1 HOST	102 Dec	20 IS (Servo)	200 Hex	1
4	Event						1
5							1
6							1

Below the main table, three 'Slave Site' configuration windows are shown, each with a 'Receive Configuration' tab:

- Slave Site(10) Config:** Station1: 1, Station2: 20, Station3: 30
- Slave Site(20) Config:** Station1: 1, Station2: 10, Station3: 30
- Slave Site(30) Config:** Station1: 10, Station2: 20

In this example, the master station No. 1 sends the value of D1000 to D8192 of the receiving station in a point-to-multiple data frame format every 100ms. According to the receiving configuration of No. 20, No. 30, and No. 40, No. 20 and No. 30 station will receive the data frame and write it into D8192, but Station No. 40 is not configured to receive the point-to-many data of Station No. 1, so it will ignore the data frame directly.

[Note]: Point-to-multiple data can be effective at the same time, which is equivalent to the master station synchronization configuration, but it is not limited to the master station to send. Each station can receive point-to-many data from 8 different stations at most, but the point-to-many data sent by each station is not limited to the number of receiving stations, that is, all the network except the sending station itself can receive, as long as the receiving configuration has been Configured to receive the site.

### 5) Network management

The network management page is shown in the figure. After the user program and CAN network configuration are downloaded to the PLC, on this page, you can monitor and debug the network, monitor the communication status of each slave station, and analyse the load rate of the CAN network, etc.:

The screenshot shows the CANLink3.0 software interface. The 'Net Info' section includes a Baud Rate of 500 Kbps, a checked 'Net Heartbeat' of 500 ms, and a 'Net Load' indicator (1). The 'Site status monitor' section has an 'Enable monitor' checkbox and a 'The Start Register Address(D):' field set to 7800, with a note 'D0 <= The begin address <= D7936'. The 'Net Management' section contains three buttons: 'Start Network(OFF)' (2), 'Synchronous Send' (3), and 'Start Monitor(OFF)' (4). Below this is a 'Device Type:' dropdown menu set to 'ALL' (5), a 'Slave Start/Stop' button (6), and a 'Station Management' button (7). The main area is a table with the following data:

Station	Station Type	Online Status	Status Register(D)	Status Code	Start/Stop Element(M)	Station Infomation
1	Host(H0U/H1U/H2U Seri...					Inovance
10	PLC(H0U/H1U/H2U Series)		3010		3010	Inovance
15	TCM		3015		3015	Inovance
20	IS(Servo)		3020		3020	Inovance
30	MD(Inverter)		3030		3030	Inovance
31	NTCM		3031		3031	Inovance

At the bottom of the interface are 'OK', 'Cancel', and 'Clear Configuration' buttons.

#### ① Network status

In non-monitoring mode, it is obtained through calculation of network load bandwidth and slave station configuration; in monitoring mode, it is obtained from the network (via the master station) in real time; network load  $\leq 50$ : green (good),  $50 < \text{network load} \leq 75$ : yellow (warning),  $75 < \text{Network load} \leq 90$ : Red (serious warning), network load  $> 90$ : ERR Red bottom (error)

## ② Network management

After connecting to the master station PLC via COM0 or USB, click the "Start Network (OFF)" button.

After successful execution: the icon after the button changes to "green", and the button name changes to "Close Network (ON)"; click the "Stop Network (ON)" button after successful execution: the icon after the button turns to "grey", and the button name is changed to "Start Network (OFF)";

[Note] The state of the network management button will be updated in real time in the monitoring state. This button is associated with M8290 and can be implemented in the user program.

## ③ Synchronous sending trigger

Through this button, you can trigger the "synchronization (M)" type configuration in the "send configuration" of all stations; this operation is mainly realized by setting "M8291", and the user program can realize the "synchronization (M)" type configuration through this component;

## ④ Monitoring and management

After connecting to the host PLC via COM0 or USB, click "Start monitoring (OFF)" to turn on the monitoring function: the icon after the button turns green, and the button name changes to "Stop monitoring (ON)"; click the "Stop Monitoring (ON)" button to turn off the monitoring function: the icon after the button turns grey, and the button name is changed to "Start Monitoring (OFF)"

[Note] If the network connection fails, it will automatically turn to the state: the icon turns grey after the button, and the button name is changed to "Start Monitoring (OFF)".

## ⑤ Device type

"Device Type" can filter the site types displayed in the list;

## ⑥ Slave management

After connecting to the master station PLC via COM0 or USB, select the station in the list: click "Start Slave (OFF)" After successful execution: the icon after the button changes to "green", and the button name is changed to "Close Slave (ON)", the corresponding station "online status" in the list is "running"; click "turn off the slave station (OFF)" after successful execution, the icon after the button becomes "grey", the button name Change to "Start Slave (ON)", and the "Online Status" of the corresponding station in the list is "Stop";

[Note] The status of the slave station management button will be updated in real time in the monitoring state. This operation is invalid for offline stations; this button is the same as "Slave station start and stop element (M)" Related, you can also set the component in the user program for operation. This part of M components is not allowed to be reused.

## ⑦ Site management

Click the "Site Management" button to open the "CANlink" wizard window and enter the "Master Station Configuration" interface;

[Note] Network management, site additions, deletions, and modifications can only be performed in the wizard; in the monitoring state, the status of the slave management button will be updated in real time, and this operation is invalid for offline sites; this button is related to "Slave start and stop components" (M)" is associated, and this component can also be set in the user program for operation. This part of M components is not allowed to be reused.

### 6) Example of access from slave station (Servo, AC drive)

Products currently supporting CANlink3.0 include Inovance PLC (H2U-XP, H1U-XP), Inovance inverter MD380, MD500, MD310, Inovance Servo, IS620P, SV660P, XP series remote expansion modules and required AutoShop software is V1.40 or above

- Servo Drive Access
- Servo CANlink communication related function codes

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: 800 Kbps 7: 1 Mbps	1	5	Upon power-on again	Running setting	PST
H0C-13	Writing the updated parameter number in the EEPROM	0: Do not update EEPROM 1: Update EEPROM	1	1	Immediately	Running setting	PST
H0C-15	CAN communication protocol selection	0: CANlink protocol 1: Reserved (CANopen protocol)	1	0	Upon power-on again	Stop setting	PST

Note: H0C.08 of IS620P does not have 800K, and selecting 6 will also be set to 1Mbps.

#### Caution:

Servo function code group number (first 2 digits) remains unchanged, function code number (last 2 digits) needs to be converted from decimal to hexadecimal. For example: H08.22 (22 is decimal) is expressed as H0816 (16 is hexadecimal) in the PLC program. Please set the three parameters of the servo station address (H0C.00), baud rate (H0C.08) and CAN protocol selection (H0C.15) in advance. And some servo function codes take effect after power-on again.

## 7) Access to the AC drive

One H1U-XP, one MD380 AC drive and one H2U-4DAR form a CANlink network. The PLC is required to control the inverter to accelerate for 20 seconds and then freely stop for 20 seconds. In this way, the control voltage is provided by 4DAR. The station number of H1U-XP is 1, the station number of 4DAR is 2, and the station number of MD380 is 3. In view of the small number of stations, and the communication distance is only about 10m, the baud rate is 500Kbps, and the heartbeat has no special requirements, and the default value is 500ms. Create a new project "CANlink3.0 Example", double-click "CAN Network Configuration" of "Project Management" on the left, and configure as shown:

The screenshot shows the 'CANLink3.0 Config Wizard' window. The 'Network Info' section has 'Baud Rate' set to 500 Kbps and 'Net Heartbeat' checked at 500 ms. The 'Network Management' section has 'Network Start/Stop Element(M)' set to 8290 and 'Synchronous Send Trigger Element(M)' set to 8291. The 'Master Type' is 'PLC(H3U Series)' and 'Master Station No.' is 1. Below is a table for 'Master Station Synchronous Write Trigger Element(M)':

E1em1 (M)	E1em2 (M)	E1em3 (M)	E1em4 (M)	E1em5 (M)	E1em6 (M)	E1em7 (M)	E1em8 (M)
1	3						

Buttons 'Back' and 'Next' are at the bottom.

The screenshot shows the 'CANLink3.0 Config Wizard' window, 'Station Info' section. 'Slave Station Type' is 'MD(Inverter)', 'Slave Station NO.' is 4, 'Status Code Register(D)' is 202, and 'Start/Stop Element(M)' is 202. Buttons 'Add', 'Delete', and 'Modify' are at the bottom.

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

A12). The following figure shows how the master station controls the MD380 AC drive.

No	Trigger Mode	Trigger	Send Station	Send Register	Receive Station	Receive Register	Length
1	Time (ms)	100	1 HOST (CPU)	100 Dec	2 IS (Servo)	3E8 Hex	1
2			1 HOST (CPU)	Dec			
3			1 HOST (CPU)	Dec			
4			1 HOST (CPU)	Dec			
5			1 HOST (CPU)	Dec			
6			1 HOST (CPU)	Dec			
7			1 HOST (CPU)	Dec			
8			1 HOST (CPU)	Dec			
9			1 HOST (CPU)	Dec			
10			1 HOST (CPU)	Dec			
11			1 HOST (CPU)	Dec			
12			1 HOST (CPU)	Dec			
13			1 HOST (CPU)	Dec			
14			1 HOST (CPU)	Dec			
15			1 HOST (CPU)	Dec			
16			1 HOST (CPU)	Dec			
Receive							
1	Event (ms)	100	2 IS (Servo)	R07 Hex	1 HOST (CPU)	150 Dec	2

The AC drive returns to the current state, as shown in the following figure.

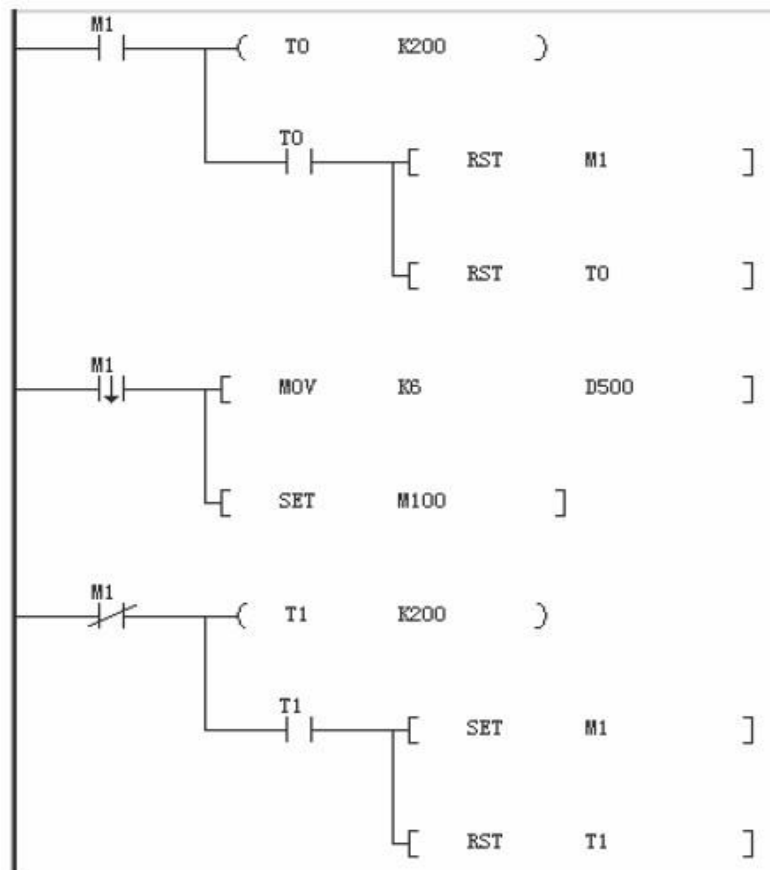
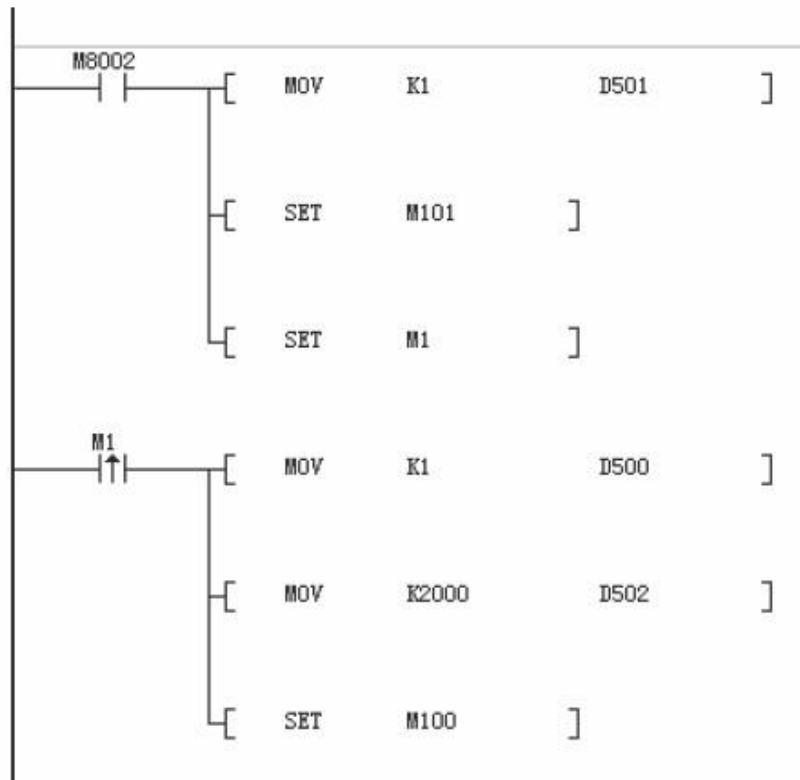
No	Trigger Mode	Trigger	Send Station	Send Register	Receive Station	Receive Register	Length
1	Time (ms)	100	1 HOST (CPU)	100 Dec	2 IS (Servo)	3E8 Hex	1
2			1 HOST (CPU)	Dec			
3			1 HOST (CPU)	Dec			
4			1 HOST (CPU)	Dec			
5			1 HOST (CPU)	Dec			
6			1 HOST (CPU)	Dec			
7			1 HOST (CPU)	Dec			
8			1 HOST (CPU)	Dec			
9			1 HOST (CPU)	Dec			
10			1 HOST (CPU)	Dec			
11			1 HOST (CPU)	Dec			
12			1 HOST (CPU)	Dec			
13			1 HOST (CPU)	Dec			
14			1 HOST (CPU)	Dec			
15			1 HOST (CPU)	Dec			
16			1 HOST (CPU)	Dec			
Receive							
1	Event (ms)	100	2 IS (Servo)	R07 Hex	1 HOST (CPU)	150 Dec	2

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 100ms.

Remote module configuration: When M101 is set, BFM#20 of the H2U-4DAR PLC is written (module reset register). Outputs are sent every 50ms.

No	Trigger Mode	Trigger	Send Station	Send Register	Receive Station	Receive Register	Length
1	Time (ms)	50	1 HOST (CPU)	502 Dec	2 IS (Servo)	1 Hex	1
2	Event (M)	100	1 HOST (CPU)	500 Dec	3 MD (Invert)	2000 Hex	1
3	Event (M)	101	1 HOST (CPU)	501 Dec	2 IS (Servo)	20 Hex	1
4			1 HOST (CPU)	Dec			

The following figure shows the application program.



### 5.9.8 CANlink common faults and troubleshooting

#### 1) Check whether the device supports CANlink 3.0

Device	How to check
PLC	Check the D8280 value. If D8280 = 300, CANlink3.0 is supported; otherwise, CANlink3.0 is not supported.
AC drive/Servo	Please check its software version: <ul style="list-style-type: none"> <li>• MD380 requires F7-11 to be 4.71.06 or above</li> <li>• MD310 requires F7-11 to be u37.18 or above</li> <li>• IS500 servo requires the value of H00-02 to be 214.xx</li> <li>• IS620P servo requires the value of H01-00 to be 6.0 or above</li> </ul>
PLC remote expansion module	Please check that the software version BFM#27 is 51210/52210 (hexadecimal 0xC80A/0xCBf2) or higher.

#### 2) Check termination resistor

Power off all equipment, use a multi-meter to measure the resistance between CANH and CANL at either end of the network. It should be about 60Ω. If it is too small, it means that the network is not only connected to matching resistors at both ends, but there are errors in other locations. Connect, disconnect the wrongly connected termination resistor. If only one equipped resistor is connected, it will be about 120Ω, and the network will have poor communication quality. The network cannot communicate without connecting to the equipped resistor. Please connect the termination resistor of the two sites at the beginning and end of the network.

#### 3) Possible reasons that the device cannot communicate at all or can occasionally communicate

- The baud rate setting is abnormal. Check whether the baud rate is normal. The baud rate of the device can only take effect after it is powered on or off or stopped and run again.
- The power connection is abnormal. The PLC CAN communication card and PLC expansion module need an external 24V power supply. It is recommended that all devices use the same power supply. If the inverter or servo is supplying power for itself, please connect all the CGND terminals of the CANlink power supply together to keep all devices sharing the CAN communication power CGND terminal.
- Check whether there is a short circuit between the communication line, shielding line, and power supply.
- If there is no problem with the above, it may be that there is a lot of interference on site. If there is no way to troubleshoot, please try to lower the communication baud rate.
- The D7000~D7999 registers cannot be modified in the user program of the PLC master station. If modified, unpredictable errors will occur. D7800~D7863 are only used to monitor the status of slave stations.

4) The FROM/TO instruction of the PLC master station cannot be used simultaneously with the CANlink3.0 configuration function.





## Chapter 6 Expansion Modules and Expansion Card Usage

### Content

6.1 Overview of extension modules .....	- 367 -
6.1.1 Naming rules.....	- 367 -
6.1.2 Expansion model .....	- 368 -
6.1.3 H2U-CAN-BD (or H1U-CAN-BD) expansion card.....	- 369 -
6.1.4 Expansion module address number .....	- 370 -
6.1.5 Module installation dimensions and user interface table .....	- 372 -
6.2 Digital input/output expansion module .....	- 373 -
6.2.1 Digital input expansion module.....	- 373 -
6.2.2 Digital output expansion module .....	- 375 -
6.2.3 Digital input and output expansion module .....	- 377 -
6.3 Analog input module .....	- 380 -
6.4 Analog output module.....	- 387 -
6.5 Analog mixed (input/output) module.....	- 393 -
6.5.1 H2U-4AM 2-channel analog current input/2-channel analog voltage and current output hybrid expansion module .....	- 393 -
6.5.2 H2U-6AM 4-channel analog current input/2-channel analog voltage and current output hybrid expansion module .....	- 402 -
6.5.3 H2U-6CM 4-channel analog voltage input/2-channel analog voltage and current output hybrid expansion module .....	- 410 -
	- 365 -

---

6.6 Temperature control module.....	- 419 -
6.7 Expansion Card Instructions .....	- 437 -
6.7.1 Instructions for H2U Series 6A/6B Expansion Card.....	- 437 -
6.7.2 Instructions for H2U Series Ethernet Communication Expansion Card.....	- 441 -
6.7.3 Instructions for H1U/ H2U Series Communication Expansion Card .....	- 447 -
6.7.4 Instructions for H1U Series 4A Expansion Card .....	- 449 -
6.8 Instructions for use of differential to collector output module.....	- 453 -

## Chapter 6 Expansion Modules and Expansion Card Usage

### 6.1 Overview of extension modules

#### 6.1.1 Naming rules

1) Digital expansion module:

**H<sub>2</sub>U-0016ERNR**

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

- ①: Inovance Controller
- ②: Model number
- ③: Input points
- ④: Output points
- ⑤: Product type. E- means expansion module
- ⑥: Output type. N - no output; R - relay output; T - transistor output
- ⑦: Indicates the type of input power. N - no power input; D - 24VDC input
- ⑧: Represents remote or local extension. R- remote expansion module, default - local expansion module

2) Analog expansion module:

**H<sub>2</sub>U-4ADR**

① ② ③ ④ ⑤

- ①: Inovance Controller
- ②: Serial number
- ③: Number of channels
- ④: Special function module type. AD - analog input; DA - analog output; PT - thermal resistor input; TC - thermocouple input; AM - mixed mode; Analog quantity module; CM - mixed analog quantity module.
- ⑤: Represents remote or local expansion. R - remote expansion module, default - local expansion module

## 6.1.2 Expansion model

Table: Expansion model

Model number	Description	Applicable model	
		H1U-XP	H2U-XP
H2U-0016ERN-INT	16-point relay output module (local)	-	√
H2U-0016ETN-INT	16-point transistor output module (local)	-	√
H2U-1600ENN-INT	16-point digital input module (local)	-	√
H2U-0800ENN	8-point digital input module (local)	-	√
H2U-0008ERN	8-point relay output module (local)	-	√
H2U-0008ETN	8-point transistor output module (local)	-	√
H2U-0404ERN	4-point digital input&4-point relay output module (local)	-	√
H2U-0404ETN	4-point digital input&4-point transistor output module (local)	-	√
H2U-0808ERN	8-point digital input&8-point relay output module (local)	-	√
H2U-0808ETN	8-point digital input&8-point transistor output module (local)	-	√
H2U-2AD	2-channel voltage/current input module (local)	-	√
H2U-2DA	2-channel voltage/current output module (local)	-	√
H2U-4AD-INT	4-channel voltage/current input module (local)	-	√
H2U-4DA-INT	4-channel voltage/current output module (local)	-	√
H2U-4HC	4-channel high-speed counter local module	-	
H2U-4PT-XP	4-channel thermistor input module (local)	-	√
H2U-4TC-XP	4-channel thermocouple input module (local)	-	√
H2U-4AM	2-channel voltage/current input&2-channel voltage/current output module (local)	-	√
H2U-6AM	4-channel current input&2-channel voltage/current output module (local)	-	√
H2U-6CM	4-channel voltage input&2-channel voltage/current output module (local)	-	√
H2U-2DC	Differential-to-OC (2 groups) output conversion module	√	√
H2U-0016ERDR	16-point relay output module (remote)	√	√
H2U-0016ETDR	16-point transistor output module (remote)	√	√
H2U-1600ENDR	16-point input module (remote)	√	√
H2U-0808ERDR	8-point digital input&8-point relay output module (remote)	√	√
H2U-2ADR	2-channel voltage/current input module (remote)	√	√

Model number	Description	Applicable model	
		H1U-XP	H2U-XP
H2U-2DAR	2-channel voltage/current output module (remote)	√	√
H2U-4ADR	4-channel voltage/current input module (remote)	√	√
H2U-4DAR	4-channel voltage/current output module (remote)	√	√
H2U-4PTR-XP	4-channel thermistor input module (remote)	√	√
H2U-4TCR-XP	4-channel thermocouple input module (remote)	√	√
H2U-4AMR	2-channel voltage/current input & 2-channel voltage/current output module (remote)	√	√
H2U-6AMR	4-channel current input & 2-channel voltage/current output module (remote)	√	√
H2U-6CMR	4-channel voltage input & 2-channel voltage/current output module (remote)	√	√

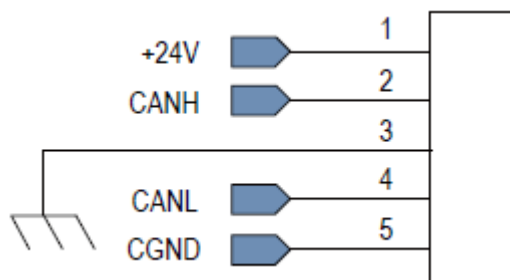
Remarks: H1U can only be connected to remote expansion modules and needs to be equipped with H1U-CAN-BD card; H2U is divided into two types, general models can only be connected to local modules, N series models with CAN interface can be connected to local and remote modules, and remote modules require H2U-CAN-BD card.

### 6.1.3 H2U-CAN-BD (or H1U-CAN-BD) expansion card

The main module CAN BD card communication interface definition:

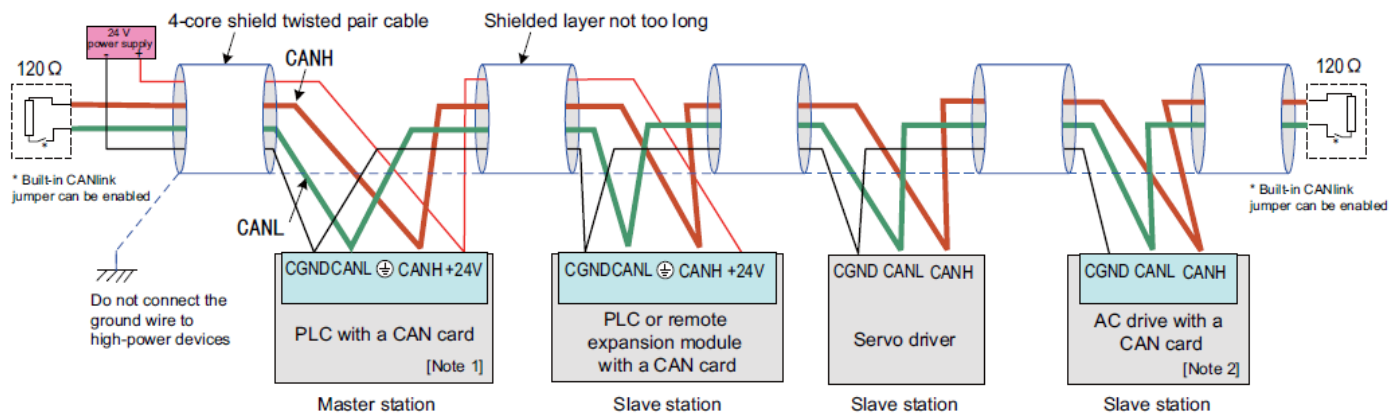
All remote expansion modules are connected to the H2U-CAN-BD (or H1U-CAN-BD) board through the CAN interface to exchange data with the PLC. CAN BD card is defined as follows:

CAN port definition:



Pin no.	Signal	Description
1	+24 V	External power supply +24 Vdc
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 0Vdc

The five wires of each device must be interconnected to form a CAN. An external +24Vdc power supply must be provided between pin 1 (+24Vdc) 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: CAN communication cable adopts shielded cable, recommended model: RVVP 2×2×0.5.

### 6.1.4 Expansion module address number

1) The address number of the local special expansion module:

All kinds of expansion modules (such as 4AD/4DA/4AM/4PT/CC-Link modules) other than IO expansion are collectively called special modules. Each time the PLC main module is powered on, it will automatically check all the expansion modules that have been connected, and "number" the special modules and IO expansion ports respectively. The user cannot intervene or change the numbering results unless the connection sequence of the modules is changed.

The address numbering method of the main module to the special module is to start from close to the main PLC module, followed by #0, #1, ... #7 and so forth. If the inserted IO expansion module does not participate in the numbering, as shown in the figure below, module address number of the 4AD expansion module is #0, the 4DA module address number is #1, and the subsequent 4PT and 4AM module addresses are #2 and #3, and so on, up to 8 special modules can be connected:

PLC Main module	#0 Special module	#1 Special module	IO	IO	#2 Special module	#3 Special module
H <sub>2U</sub> -1616MR X <sub>0</sub> ~X <sub>17</sub> Y <sub>0</sub> ~Y <sub>17</sub>	H <sub>2U</sub> -4AD	H <sub>2U</sub> -4DA	H <sub>2U</sub> -1600ENN X <sub>20</sub> ~X <sub>37</sub>	H <sub>2U</sub> -0016ERN Y <sub>20</sub> ~Y <sub>37</sub>	H <sub>2U</sub> -4PT	H <sub>2U</sub> -4AM

Even if there are several IO expansion modules inserted in the middle, the numbering sequence will not be affected. Knowing the above-mentioned addressing principles, users can accurately access the specified modules during programming.

Before powering on, the external 24V/COM power terminal of the module must be properly connected to the 24V/COM of the PLC main module, so that when the main module is powered on, the expansion module can also be powered on at the same time. If the external power supply of the expansion module lags behind, the PLC main module may not be able to correctly identify the module type; the PLC main module only checks all the expansion modules connected to the system once when the power is turned on. The expansion modules inserted during operation will not be checked by the PLC main module and cannot be accessed normally; plugging or unplugging the expansion module during operation may damage the device, and more serious, may cause unpredictable control output status, leading to user equipment failure.

2) The address number of the remote expansion module:

In the remote expansion module, the remote module address is: module communication station number +100, a maximum of 62 remote expansion modules are allowed (the CANlink3.0 version allows a maximum of 30 remote expansion modules), and each remote module has an eight-digit DIP switch, the user can set the station number of the module, select the baud rate, whether to connect the termination resistor (standard 120Ω). As shown in the figure below, "ON" means logic "1".

Table 1: DIP switch definition of CANlink3.0 version										
Terminal resistor	Description	Baud rate		Description	Address settings					Description
		7	6		5	4	3	2	1	
8										
0	Terminal matching resistance connected	0	0	500kbps	0	0	0	0	0	Retain
					0	0	0	0	1	1
		0	1	250kbps	0	0	0	1	0	2
					0	0	0	1	1	3
1	Terminal matching resistance is disconnected	1	0	100kbps	.....					
					1	1	1	0	1	29
		1	1	50kbps	1	1	1	1	0	30
					1	1	1	1	1	31

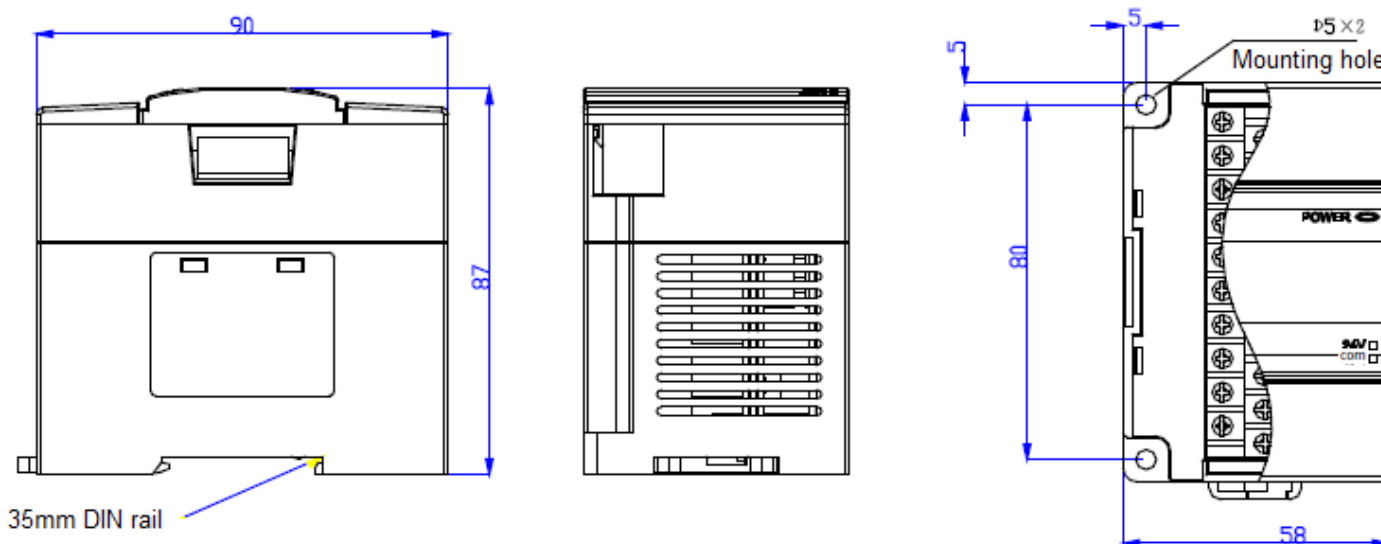
Table 1: DIP switch definition of CANlink3.0 version											
Terminal resistor	Description	Baud rate	Description	Address settings					Description		
				6	5	4	3	2		1	
8											
0	Terminal matching resistance connected	0	500kbps	0	0	0	0	0	0	Retain	
				0	0	0	0	1	0	2	
		0	0	0	0	0	0	0	1	1	3
					0	0	0	0	1	1	
1	Terminal matching resistance is disconnected	1	100kbps	.....							
				0	1	1	1	0	1	29	
		1	1	100kbps	0	1	1	1	1	0	30
					1	1	1	1	1	1	63

If you change the DIP switch, except for the matching resistance, the baud rate and address cannot take effect immediately. You need to power on the system again to use the new setting parameters. Pay attention to the uniqueness of the CAN address, and there can be no duplicate addresses.

Note: The versions above 51210-0000 and 52210-0000 (including these two versions) support CANlink3.0.

### 6.1.5 Module installation dimensions and user interface table

1) The installation size of the module:



2) Module user interface description

Item	Description	
Terminal functions	V+: positive input/output terminal of the channel; I+: current positive input/output terminal of the channel; VI-: input/output public end of the channel; 24+/24-: external 24 V input terminal; GND: protection ground.	
Indicator	Local extension module	PWR: It is on when the module is connected to the main module and the main module is powered on. COM: It blinks fast when the module works normally and slowly when a failure occurs. 24 V: It is on when the external 24 V power is connected.
	Remote extension module	PWR: The module's 24V power supply is normal COM: When the FROM/TO instruction is used to access, it lights up with access, and goes off without access. COM and ERR: When using CANlink3.0 to configure access, please refer to the relevant instructions of CANlink3.0 application manual.
Extended port	Local extension module	The extension input end is a 26-pin extension module cable connector. A flat cable is connected to the module. The extension output end is a 26-pin extension module cable connector.
	Remote extension module	Extension input end is a 5-pinconnector. It is recommended to use a 4-core shielded twisted pair line or network cable as the transmission line.



## 6.2 Digital input/output expansion module

### 6.2.1 Digital input expansion module

H2U-1600ENN-INT, H2U-1600ENDR-INT, H2U-0800ENN-INT

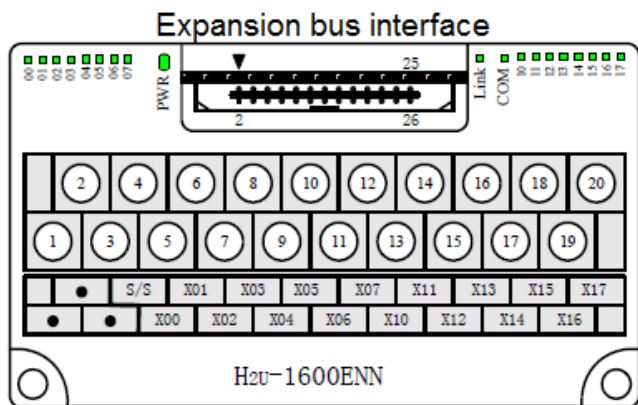
1) Electrical specifications:

Item		Index
Signal input method		Sink type/source type method. When the S/S terminal is shorted to 24V, it is a sink input; when the S/S terminal is shorted to COM, it is a source input.
Electrical parameters	Detection voltage	DC24V
	Input resistance	4.3K
	Input is ON	3.5mA min
Filter function	Input is OFF	1.4mA max
	Digital filtering	None
	Hardware filtering	Filter time is about 10ms
Common terminal		There is only one common end, which is S/S

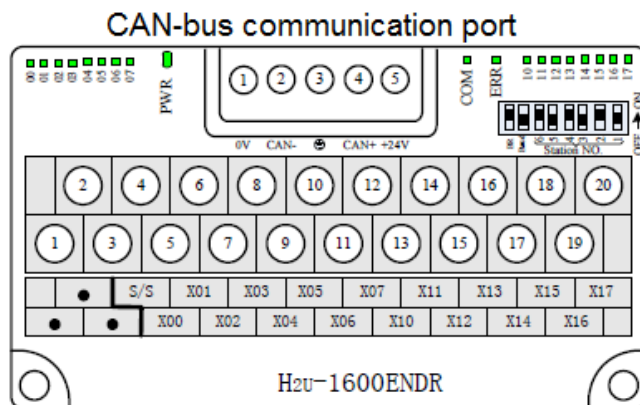
2) User connection interface table

Item	Comments	
Terminal function	X00~X17: input point S/S: input point common terminal, sink input is connected to +24V, source input is connected to 0V	
Indicator	Local extended module	PWR: The power supply of the module's digital circuit is normal X00~ X17: input point indicator LINK: Lights up when the main module accesses the expansion module RUN: Lights up when the module is working normally.
	Remote expansion module	PWR: The module's 24V power supply is normal X00~ X17: input point indicator COM: When the FROM/TO instruction is used to access, it lights up with access, and goes off without access. COM and ERR: When using CANlink3.0 configuration access, please refer to the relevant instructions in the CANlink3.0 application manual
Expansion port	Local expansion module	The extension input end is a 26-pin extension module cable connector. A flat cable is connected to the module. The extension output end is a 26-pin extension module cable connector.
	Remote expansion module	Extension input end is a 5-pinconnector. It is recommended to use a 4-core shielded twisted pair line or network cable as the transmission line

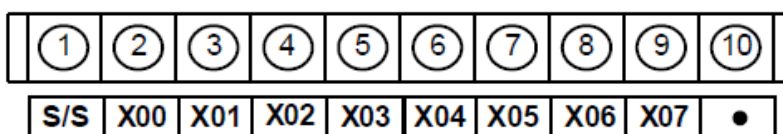
3) Port description



H2U-1600ENN



H2U-1600ENDR



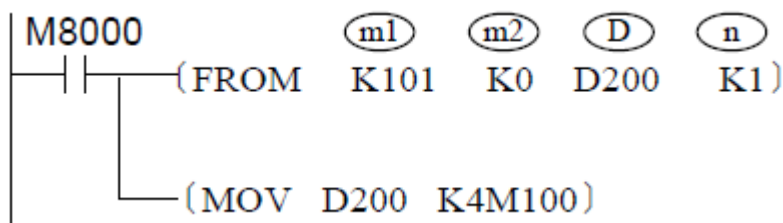
H2U-0800ENN

4) Local expansion and remote expansion programming methods:

When the local expansion module is later connected to the main module, the number of the X port on the expansion module is next to the number of the X port on the main module, and is numbered backwards. For example, when the main module is H2U-1616MR, now it is necessary to connect the H2U-1600ENN extension Module, because the last X port number of the main module is X17, the access number of X of the expansion module during programming is X20~X37, that is, the X0 point on the expansion module corresponds to X20 in the program, and so on.

Note that the number of the expansion module always starts from 0 in the octal unit. For example, when the main module is H2U-3624MR, the last X port number is X43, and the access number of the X of the expansion module during programming is X50~ X67, the vacant port numbers of X44~X47 on the main module are discarded. The Y port on the expansion module also takes the same approach.

The remote digital expansion module cannot directly access the X and Y points. Must be accessed by FROM/TO instruction, the programming method is as follows:



This example introduces the access method of the remote digital expansion module H2U-1600ENDR, because the 16-point input module uses a BFM area to represent all input points, that is, each bit of K0 (a register) in the BFM area corresponds to each input Point, so you can read back the status of the module and use MOV D200 K4M100 to respectively take the position. At this time, M100~M107 are X0~X7 of the remote

module, and M108~M115 are X10~X17 of the remote module. We can read M100~M115 can read the status of X0~X17. Note: In order to ensure that the latest status of the input points of the digital expansion module can be read, please use M8000 to drive the FROM instruction so that it can be updated every scan cycle.

### 6.2.2 Digital output expansion module

H2U-0016ERN-INT, H2U-0016ERDR-INT, H2U-0016ETN-INT, H2U-0016ETDR-INT, H2U-0008ERN-INT, H2U-0008ETN-INT

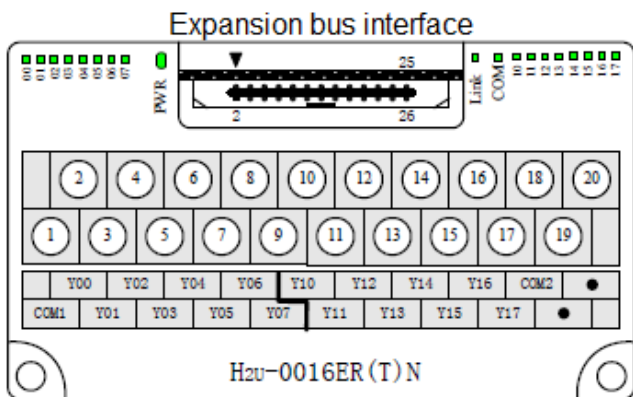
#### 1) Electrical specifications:

Item		Relay output port	Transistor output port
Power supply		AC250V, below DC30V	DC5~24V
Isolation		Relay mechanical insulation	Optocoupler insulation
LED indicator		When the relay output contact closes, the LED indicator becomes ON.	When the opto-coupling is driven, the LED indicator becomes ON.
Leakage current during open circuit		-	Less than 0.1 mA/30 VDC
Min. load		2 mA/5 VDC	5 mA (5~24 VDC)
Max. output current	Resistive load	2A/1 point 8A/4 points common port 8A/8 points common port	0.5A/point 0.8A/4 points 1.6A/8 points
	Inductive load	220 VAC, 80 VA	12 W/24 VDC
	Lamp load	220 VAC, 100 W	12 W/24 VDC
ON response delay		20ms (max.)	High speed output: 10μs Others: 0.5ms
OFF response delay		20ms (max.)	
Output common port		Each group shares a common port and the groups are isolated	
Fuse protection		Without fuse protection	

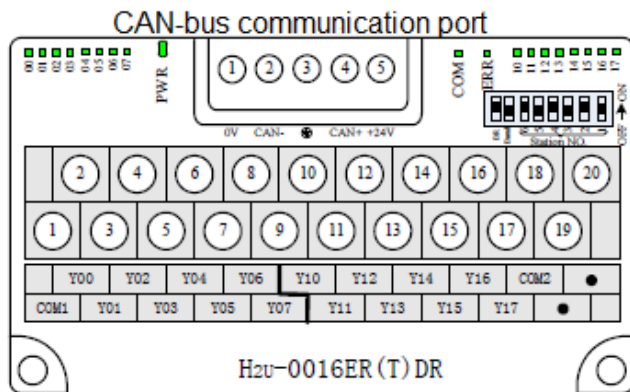
#### 2) User connection interface table

Item	Comments	
Terminal function	Y00~Y17: input point COM1~2: common terminal of output point	
Indicator	Local extended module	PWR: The power supply of the module's digital circuit is normal Y00~ Y17: input point indicator LINK: Lights up when the main module accesses the expansion module RUN: Lights up when the module is working normally.
	Remote expansion module	PWR: The module's 24V power supply is normal Y00~ Y17: input point indicator COM: When the FROM/TO instruction is used to access, it lights up with access, and goes off without access. COM and ERR: When using CANlink3.0 configuration access, please refer to the relevant instructions in the CANlink3.0 application manual
Expansion port	Local expansion module	The extension input end is a 26-pin extension module cable connector. A flat cable is connected to the module. The extension output end is a 26-pin extension module cable connector.
	Remote expansion module	Extension input end is a 5-pinconnector. It is recommended to use a 4-core shielded twisted pair line or network cable as the transmission line

3) Port description



H2U-0016ER(T)N



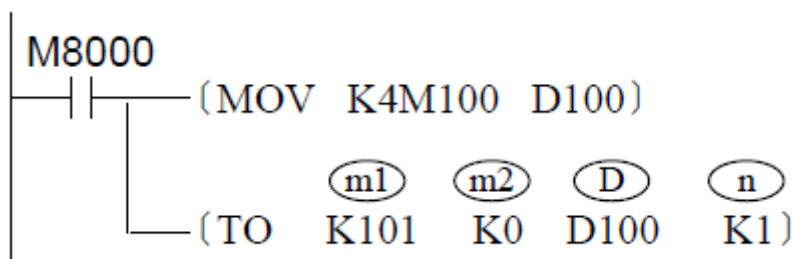
H2U-0016ER(T)DR

4) Local expansion and remote expansion programming methods

After the local expansion module is connected to the main module, the number of the Y port on the expansion module is numbered backwards according to the number of the Y port on the main module. For example, when the main module is H2U-1616MR, it is now connected to H2U-0016ERN Expansion module, because the last Y port number of the main module is Y17, the access number of Y of the expansion module during programming is Y20~Y37, that is, the Y0 point on the expansion module corresponds to Y20 in the program, and so on.

Note: The port number of the expansion module always starts from 0 in the octal unit. The relay output expansion module can be connected to the relay or transistor main module; similarly, the transistor output expansion module can also be connected to the transistor or relay main module.

The remote digital expansion module cannot directly access the X and Y points. Must be accessed by FROM/TO instruction, the programming method is as follows:



This example introduces the access method of the remote digital expansion module H2U-0016ERDR, because the 16-point input module uses a BFM area to represent all output points, that is, each bit of K0 (a register) in the BFM area corresponds to each output. Therefore, you can put the status of all output points in one word and then transfer it to the module. In the example, M100~M107 are Y0~Y7 of the remote module, and M108~M115 are Y10~Y17 of the remote module. M100~M115 can change the status of Y0~Y17 in the remote module. In the remote digital output module, the TO command can be driven after each output point is updated, or it can be driven by M8000.

The remote output expansion module newly added the function of BFM#5: it is the communication

timeout judgment time, the time unit is 1ms, when the communication between the main module and the remote module is lost for more than the set value, all Y output ports of the remote module All will be closed (the default waiting time for communication is 500ms); if BFM#5=0, this function is invalid and the module keeps the original output state.

### 6.2.3 Digital input and output expansion module

H2U-0016ERN, H2U-0016ERDR, H2U-0016ETN, H2U-0016ETDR, H2U-0808ERN, H2U-0808ETN

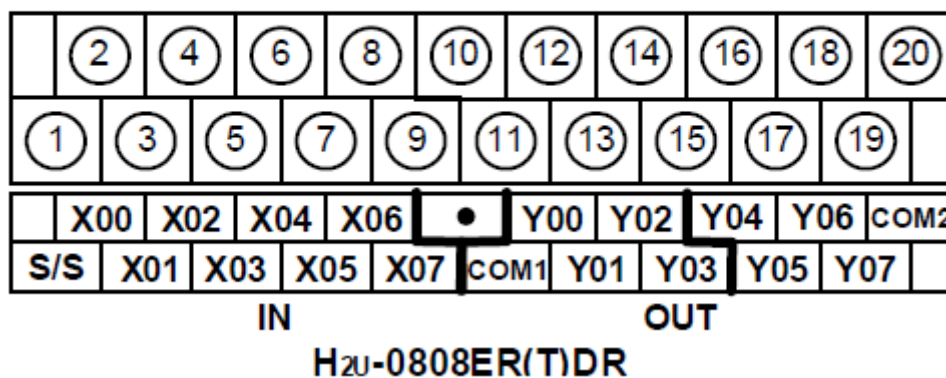
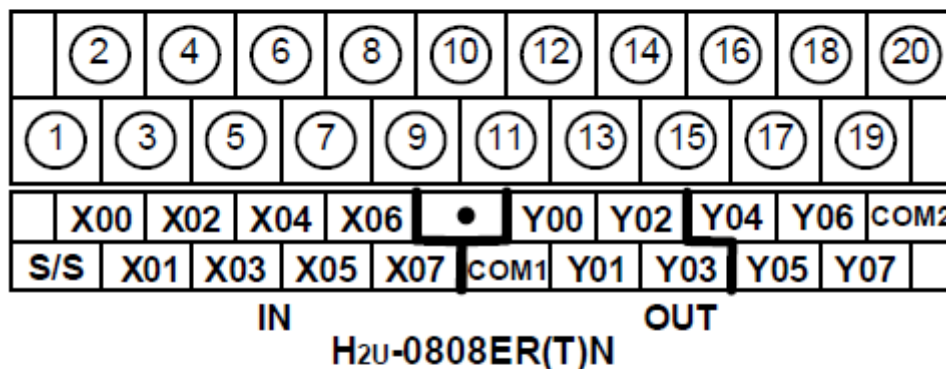
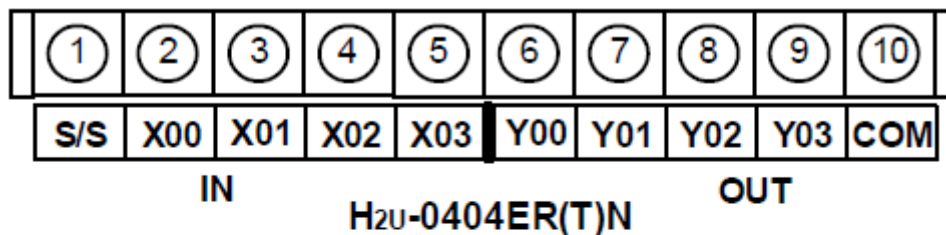
#### 1) Electrical specifications:

Item		Index
Signal input method		Sink type/source type method. When the S/S terminal is shorted to 24V, it is a sink input; when the S/S terminal is shorted to COM, it is a source input.
Electrical parameters	Detection voltage	DC24V
	Input resistance	4.3K
	Input is ON	3.5mA min
Filter function	Input is OFF	1.4mA max
	Digital filtering	None
	Hardware filtering	Filter time is about 10ms
Common terminal		There is only one common terminal, which is S/S

#### 2) User connection interface table

Item		Relay output port	Transistor output port
Power supply		AC250V, below DC30V	DC5~24V
Isolation		Relay mechanical insulation	Optocoupler insulation
LED indicator		When the relay output contact closes, the LED indicator becomes ON.	When the opto-coupling is driven, the LED indicator becomes ON.
Leakage current during open circuit		-	Less than 0.1 mA/30 VDC
Min. load		2 mA/5 VDC	5 mA (5~24 VDC)
Max. output current	Resistive load	2A/1 point 8A/4 points common port 8A/8 points common port	0.5A/point 0.8A/4 points 1.6A/8 points
	Inductive load	220 VAC, 80 VA	12 W/24 VDC
	Lamp load	220 VAC, 100 W	12 W/24 VDC
ON response delay		20ms (max.)	High speed output: 10μs Others: 0.5ms
OFF response delay		20ms (max.)	
Output common port		Each group shares a common port and the groups are isolated	
Fuse protection		Without fuse protection	

## 3) Port description



## 4) Local expansion and remote expansion programming methods

When the local expansion module is later connected to the main module, the number of the X and Y ports on the expansion module is next to the number of the X and Y ports on the main module, and they are numbered backwards. For example, when the main module is H2U-1616MR, it is now connected H2U-0808ENN extension module, because the last X port number of the main module is X17, the access number of the X of the extension module during programming is X20~X27, the last Y port number of the main module is Y17, and the Y of the extension module is during programming The access numbers of Y20~Y27, and so on.

Note that the number of the expansion module always starts from 0 in the octal unit. For example, when the main module is H2U-3624MR, the last X port number is X43, and the access number of the X of the expansion module during programming is X50~ X57, that is, the vacant port numbers of X44~X47 on the main module are discarded. The Y port on the expansion module also takes the same approach.

The remote digital expansion module cannot directly access the X and Y points. Must be accessed through FROM/TO instructions.

5) Access the BFM area of the digital remote expansion module:

Model	Description	Remark
H2U-1600ENNR	BFM #0=The input port status of the expansion module (read X input)	As per bit
H2U-0016ERNR	<ul style="list-style-type: none"> <li>▪ BFM#0=Extension module output buffer (set Y output)</li> <li>▪ BFM#5=CAN communication timeout automatic shutdown output time (unit of ms, 1~32767, factory default is 500ms; set to 0 means that this function is disabled)</li> </ul>	As per bit
H2U-0808ERNR	<ul style="list-style-type: none"> <li>▪ BFM#0=Expansion module input buffer (read X input)</li> <li>▪ BFM#1=Extension module output buffer (set Y output)</li> <li>▪ BFM#5=CAN communication timeout automatic shutdown output time (unit of ms, 1~32767, factory default is 500ms; set to 0 means that this function is disabled)</li> </ul>	Lower 8 bit

## 6.3 Analog input module

H2U-4AD(R) / H2U-2AD 4-channel/2-channel analog input expansion module

### 1) Introduction

The 4AD(R) / 2AD expansion module can work with the H2U series main module to realize the signal detection of 4/2 analog input channels and convert the signal of -10V~10V or -20mA~20mA into 12bit digital signal so that the PLC main module can read. The main module accesses the BFM unit of the register in the expansion module through the FROM/TO instruction.

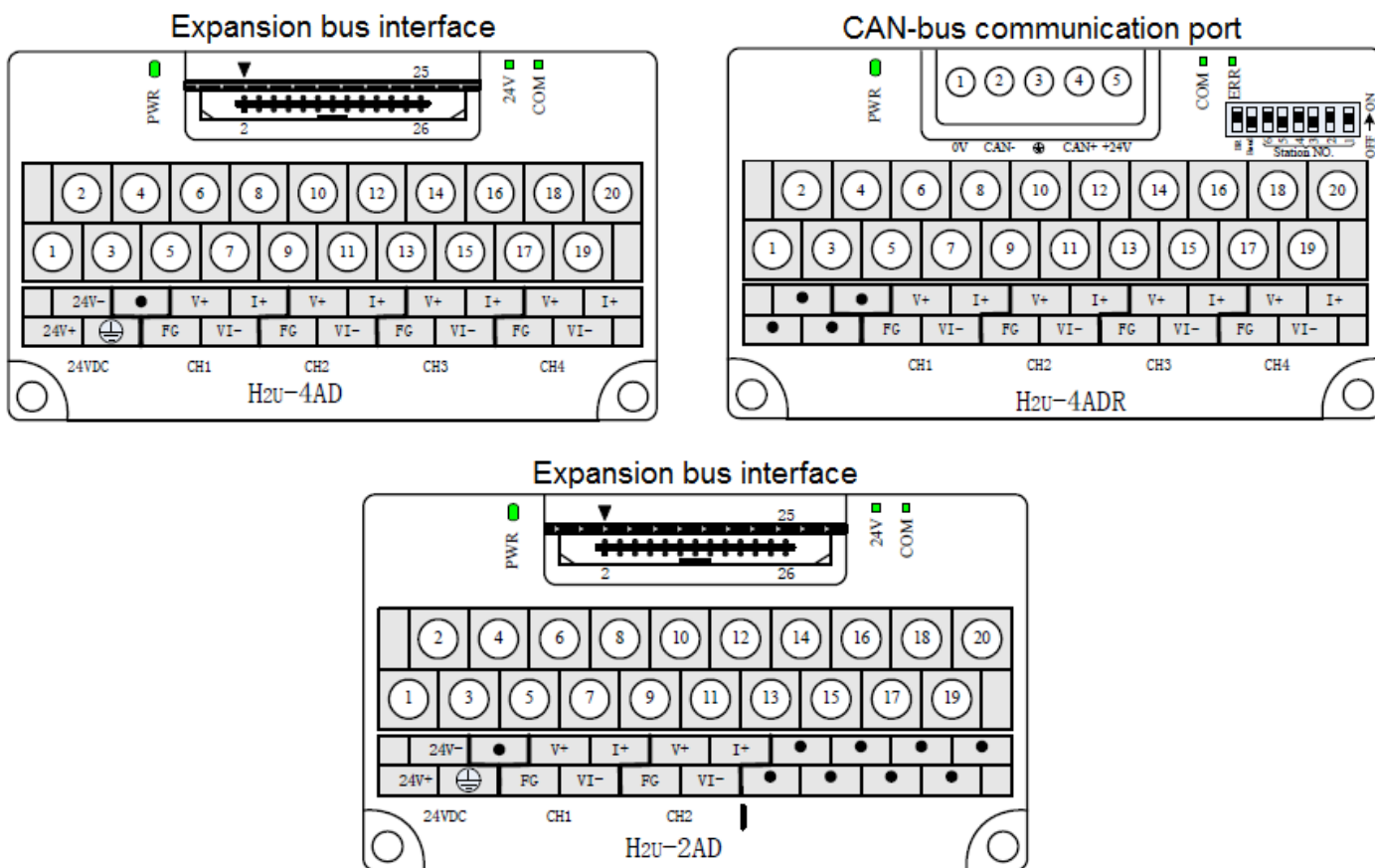
Remarks: The difference between 4AD(R) and 2AD: 4AD(R) has 2 more input channels in hardware than 2AD, BFM, usage, programming, etc. are the same.

### 2) Electrical specifications

Item		Description	
Power supply	Analog	24V DC -15%/+20%, maximum allowable ripple voltage 5%, current consumption 80mA (taken from 24V/COM of the main module, or other 24VDC power supply)	
	Digital	5V DC 50mA (through the module extension cable, taken from the main module power supply internal power supply)	
	Remarks	The remote expansion module does not require a digital circuit power supply.	
Number of input channels		4DA (R): 4 channels; 2DA: 2 channels	
Analog input range	Voltage input	-10~10V DC, (input impedance is 200K $\Omega$ )	The user needs to set the corresponding BFM area according to the type of signal to be connected.
	Current input	-20~20mA (input impedance is 250 $\Omega$ )	
Digital output		-2048 ~ +2047	
Resolution	Voltage input	5mV	
	Current input	20 $\mu$ A	
Conversion speed		15ms/channel (normal speed), 6ms/channel (fast)	
Precision		$\pm$ 1% full range	
Occupied I/O points		Does not occupy the main module I/O points	
Isolation		The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.	

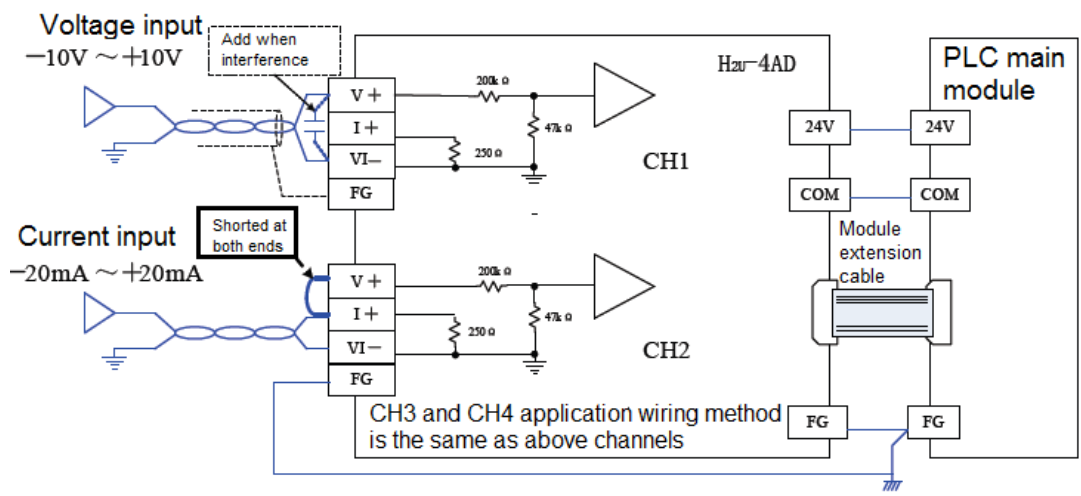


3) Port description



4) Wiring of input signal and extension cable:

The analog input signal is connected to the input port of the expansion module through a twisted pair cable. When wiring, do not close to the AC power line or the line that interferes with the signal; if the interference of the analog signal is serious, use a shielded cable, and connect a 0.1μF/25V high-frequency capacitor in parallel at the input port, the shell of the signal source and its shielding wire is connected to the signal ground terminal FG of H2U-4AD, and is grounded together.



Note: If the input signal in a channel exceeds the specified range, it will affect the accuracy of other channels.

## 5) BFM area of the 4AD(R)/2AD module:

The PLC main module reads the digital AD conversion result by reading the register buffer unit (BFM area) of the 4AD(R)/2AD module and sets the module status by rewriting the specific BFM area. The PLC main module accesses these BFM units through read and write instructions FROM/TO.

The expansion module is equipped with EEPROM storage unit, which is used to save some BFM setting values, such as the signal type, offset value, gain value of each analog input channel, etc. The saving action of these units is determined by the setting status of the corresponding BFM unit and done automatically.

The width of each register in the BFM area is 16-bit (i.e. 1Word), which is defined in the following table according to the BFM area of the 4AD(R)/2AD module: (2AD channels 3 and 4 are reserved)

BFM	R/W	Content	
#0 (E)	WR	Channel signal mode selection, each HEX bit represents 1 input channel, 4AD (R): the highest bit is ch4, the lowest bit is ch1: (default value = H0000), 2AD: take the high HEX bit in the lower 8 bits as ch2, and the low HEX bit as ch1: (default value = H00) 0=-10V~10V; corresponding digital output: -2000~2000 1=4mA~20mA; corresponding digital output: 0~1000 2=-20mA~20mA; corresponding digital output: -1000~1000 3=This channel is closed; 4=-10V~10V; corresponding digital output: -10000~10000 5=4mA~20mA; corresponding digital output: 0~10000 6=-20mA~20mA; corresponding digital output: -10000~10000	<div style="border: 1px dashed black; padding: 5px; text-align: center;"> <p><b>Example:</b></p> <p>ch4 3 2 1</p> </div>
#1	WR	Channel 1	Average filter constant, namely, the number of sampled values for average value calculation. Range: 1-4096; default value: 8. The value can be set to 1 if high-speed collection is needed. When BFM#15 is changed, the value is restored to the default.
#2	WR	Channel 2	
#3	WR	Channel 3	
#4	WR	Channel 4	
#5	R	Channel 1	Mean filtering value of the sampled value of the input channel
#6	R	Channel 2	
#7	R	Channel 3	
#8	R	Channel 4	
#9	R	Channel 1	Currently sampled data of the input channel, namely, burst value without filtering
#10	R	Channel 2	
#11	R	Channel 3	
#12	R	Channel 4	
#13-14	-		Reserved
#15	WR	ADC speed selection	0 = normal speed, 15 ms/channel (default value); 1 = fast conversion, 6 ms/channel; 1000-30000 = high-speed sampling, corresponding to 1 ms-30 ms/channel
(#16-19)	-	Reserved	
#20 (E)	WR	1 = Reset the value to the default (factory setting) Default = 0	

BFM	R/W	Content											
#21 (E)	WR	2= Adjusting offset/gain values is not allowed (default); 1=Adjusting offset/gain values is allowed											
#22 (E)	WR	Operations for the four channels corresponding to the lower 8 bits				G4	O4	G3	O3	G2	O2	G1	O1
		Offset/gain value adjusting is enabled. When the value is non-0, the module writes the BFM 23/24 value into the control register of the internal channel. Initial value = H00.											
#23 (E)	WR	Offset value. When the digital output is 0, the initial value of the analog input value (mode 0, 1, or 2) is 0.											
#24 (E)	WR	Gain value. When the digital output is +1000, the initial value of the analog input value (mode 0, 1, or 2) is 5000.											
#25- 26	-	Reserved											
#27	R	4AD module software version											
#28	-	Reserved											
#29	R	Error status											
#30	R	ID of the extension module. The ID of H2U-4AD is K2010.											
#31	-	Reserved, not accessible											

Note: After the module is powered off, the curve characteristics of the output conversion are saved, including the offset gain characteristics of the curve. The offset gain displayed in the current BFM area is the value to be effective, not the offset gain currently effective.

The meaning of the status information word BFM #29 is explained as follows:

BFM# 29 bit	ON	OFF
b0:	An error exists. When any value in b0-b3 is not 0, A/D conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal.
b10:	Digital output exceeds the range of -2048-2047.	Digital output is normal.
b11:	The sampling filter constant exceeds the range of 1-4096.	The sampling filter constant is normal.
b12:	The value of BFM#21 cannot be set to K2.	The value of BFM#21 can be set to K2.
Bits 4-7 and bits 13-15 of BFM#29 are not defined.		

In the table, the BFM unit with E is the project stored in EEPROM. It will not be deleted upon power off.

#### Register Modification

The modification of the BFM units carrying (E), such as BFM#0, #23, and #24 causes the write operation on EEPROM. In the write operation, each Word requires about 300ms. When the write operation is performed on multiple BFM units carrying (E), the PLC programming requires a delay after each BFM unit is written. The next write operation can be performed only when the previous instruction is complete.

Note: The external 24 V power supply for the local extension module is lost. The system flag M8067 of the PLC main module is reset. The error code is D8067=K6708. This symptom can be detected if you often check the flag in editing.

#### 6) BFM area explanation

##### BFM#0 channel selection

The channels are initialized. By default, all the four channels are -10 V to 10 V, which is controlled by the hex string HXXXX of BFM #0. The lowest bit controls channel 1, the second lowest bit controls channel 2, the third lowest bit controls channel 3, and the highest bit controls channel 4. The character control is as follows:

X=0, pre-defined range: -10 V to 10 V (-2000 to 2000)

X=1, pre-defined range: 4 mA to 20 mA (0 to 1000)

X=2, pre-defined range: -20 mA to 20 mA (-1000 to 1000)

X=3, the channel is closed.

X=4, pre-defined range: -10 V to 10 V (-10000 to 10000)

X=5, pre-defined range: 4 mA to 20 mA (0 to 10000)

X=6, pre-defined range: -20 mA to 20 mA (-10000 to 10000)

For example, if BFM#0 is H1230, channel 1 is -10 V-10 V, channel 2 is closed, channel 3 is -20 mA-20 mA, and channel 4 is 4 mA-20 mA.

The unused channel can be closed or not. The closed channel does not occupy conversion time (BFM#15). In the example, channel 2 is closed, so the total conversion time is the sum of conversion time of the other three channels, namely, (3xBFM#15).

BFM #1~#4 average sampling number. Sampling value of each channel (BFM#9 to #12) plus the number of samples (BFM #1 to #4) divided the number of samples (BFM #1 to #4) are stored in BFM #5 to #8).

BFM #5 to #8 stores the average sampling values.

BFM #9 to #12 stores the instant sampling values.

##### BFM #15 ADC rate

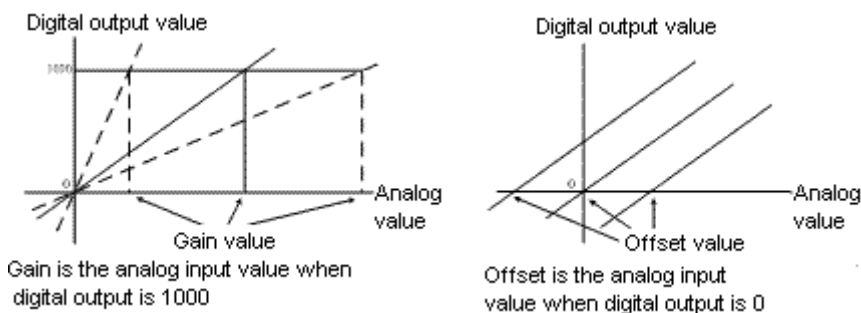
Time of one conversion of each channel. The time of each data update is BFM#15 x number of opened channels.

For example, when BFM #0 is H3310, BFM #1 is K7, BFM #2 is K6, and BFM #15 is K10, the time of one data update of BFM #9 and BFM #10 is  $\text{BFM}\#0 \times \text{BFM}\#15 = 2 \times 10 = 20\text{ms}$ , the time of one data update of BFM #5 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#1 = 2 \times 10 \times 7 = 140\text{ms}$ , and the time of one data update of BFM #6 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#2 = 2 \times 10 \times 6 = 120\text{ms}$ . The FROM/TO instruction is time consuming, so the BFM#5 data collected by this module can be read using the LDP M8012 FROM K0 K5 D10 K1 instruction, which has the same effect as using LD M8000 FROM K0 K5 D10 K1. However, the data is read in each scanning period of the M8000 drive.

BFM #20 factory restore setting:

To restore the default setting, set #20 to 1.

BFM #21 to #24 Definition and setting method of offset and gain values:



The offset and gain values can be set separately or together. The normal gain value ranges from 1 V to 15 V or 4 mA to 32 mA. The normal offset value ranges from -5 V to 5 V or -20 mA to 20 mA.

Before setting the offset/gain, set BFM#21 to 1, modify BFM#23/24, and then allow offset/gain BFM#22 for each channel. After the modification, set BFM#21 to 2 again, to avoid change.

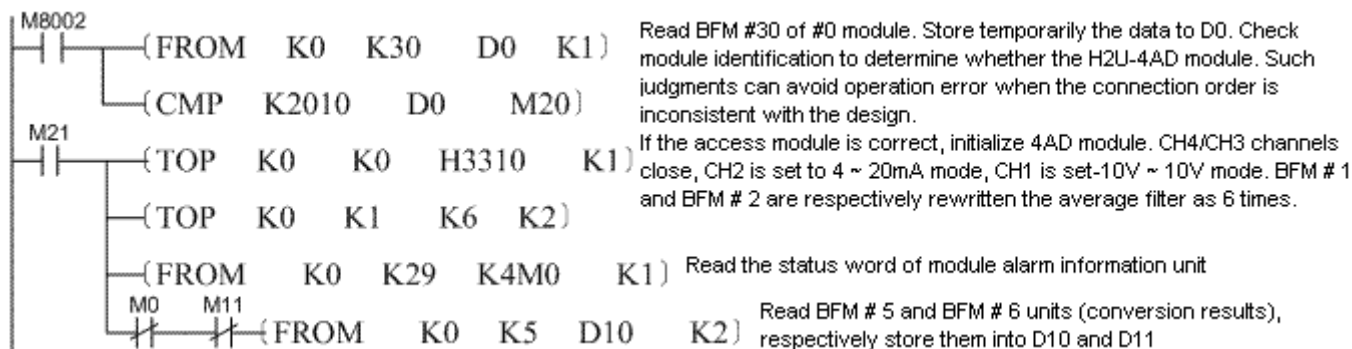
Note that the channels requiring offset/gain must be the same. It is not allowed that the offset of one channel is 1000 but the offset of another channel is 1200.

For example, when BFM#0 is in mode 0, set the offset and gain of channel 1 and channel 2 to 0.5 V and 6 V, respectively. The operation is as follows:

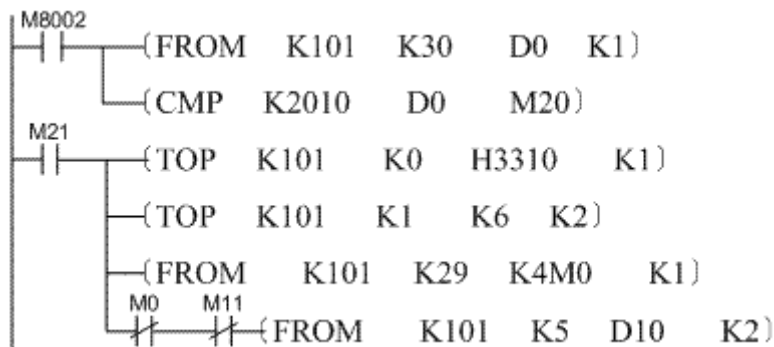
Change BFM#21 to 1. After 300ms, send K500 and K6000 to BFM#23 and BFM#24 respectively. After 300ms, allow gain BFM#22. In this example, BFM#22 should be binary form 00001111. That is, change BFM#22 to H.000F. Then change BFM#21 to 2, to prevent the data from being modified again.

Programming example 1:

An H2U-4AD module is inserted behind the PLC main module and is numbered #0. The CH1 port needs to collect the voltage signal of -10 V-10 V, CH2 needs to collect the current signal of 4-20 mA, and CH3/CH4 are not in use. It is required to change the filtering times to 6. The data collected from the two channels is stored in D10 and D11. The user program is as follows:

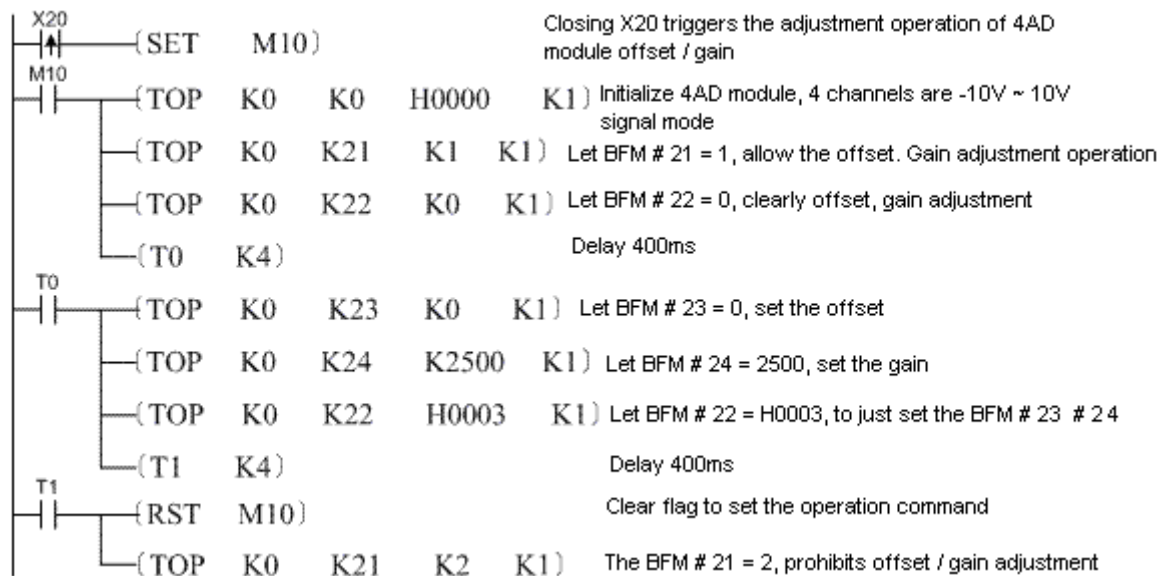


In this example, if the module is changed to H2U-4ADR, and the CAN station address is 1, then the sub-program is as follows:



Programming example 2:

For the 4AD module with number #0, it is required that the offset/gain setting be performed for CH1 by using an X20 button. The programming is as follows:



Precautions:

Online removal or installation is forbidden. The extension module can be connected or removed only when the main module and application systems are powered off; otherwise, there may be a harm to human body or the components may be damaged.

## 6.4 Analog output module

H2U-4DA(R)/H2U-2DA: 4 channel/2 channel analog output expansion module

### 1) Introduction

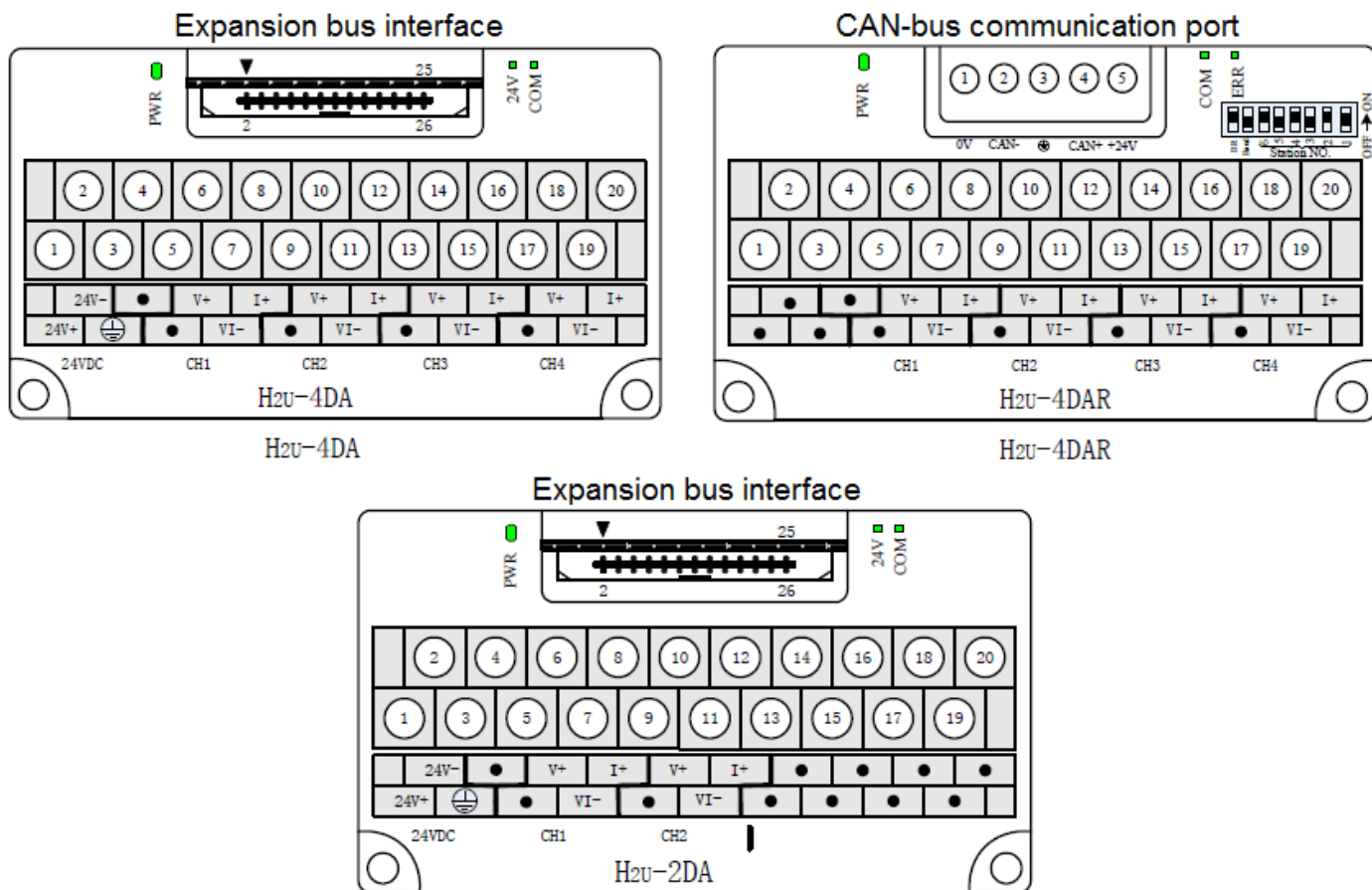
The 4DA(R)/2DA expansion module can work with the H2U series main module to realize the output of 4/2 analog signal channels, each channel has voltage signal and current signal output ports, and the signal amplitude can be -10~10V or 0-20mA. The main module accesses the BFM unit of the register in the expansion module through the FROM/TO instruction to control the analog output signal.

Remarks: The difference between 4DA(R) and 2DA: 4DA(R) has 2 extra output channels in hardware than 2DA, BFM, usage, programming, etc. are the same.

### 2) Electrical specifications:

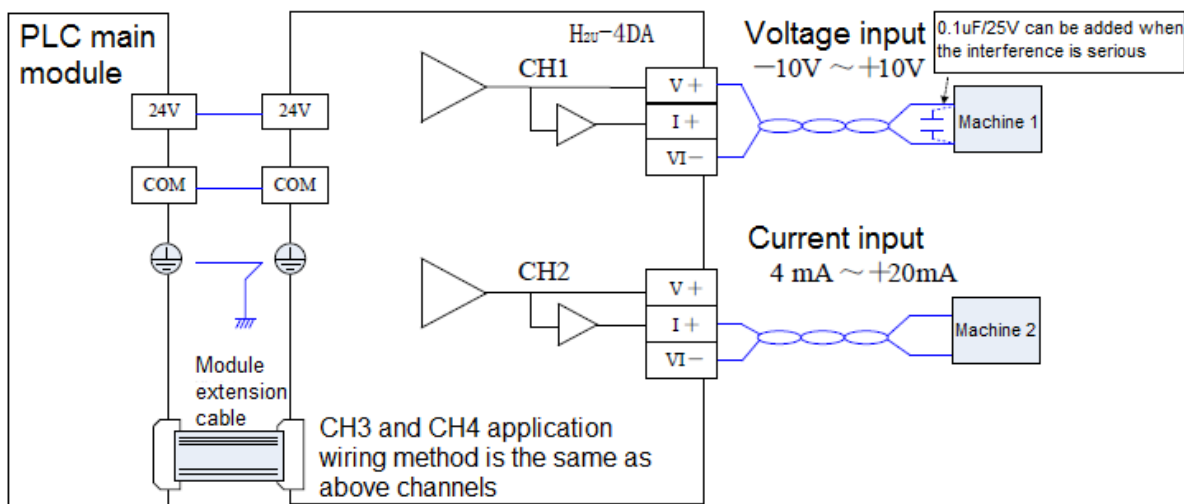
Item		Description	
Power supply	Analog	24V DC -15%/+20%, maximum allowable ripple voltage 5%, current consumption 80mA (taken from 24V/COM of the main module, or other 24VDC power supply)	
	Digital	5V DC 50mA (through the module extension cable, taken from the main module power supply internal power supply)	
	Remarks	The remote expansion module does not require a digital circuit power supply.	
Number of input channels		4AD (R): 4 channels; 2AD: 2 channels	
Analog input range	Voltage input	-10~10V DC, (input impedance is 2K $\Omega$ )	The user needs to set the corresponding BFM area according to the type of signal to be connected.
	Current input	4DA(R): 4~20mA (the external load impedance is 100 $\Omega$ ~500 $\Omega$ ) 2DA: 0~20mA (The external load impedance is 500 $\Omega$ or less)	
Digital output		Default setting: -2000 ~ 2000	Allowable range: -10000 ~10000
Resolution	Voltage input	5mV	Corresponds to 10V/2000
	Current input	20 $\mu$ A	Corresponds to 20mA/1000
Conversion speed		4 channels: 4ms/channel 2 channels: 2.1ms/channel	
Precision		$\pm$ 1% full range	
Occupied I/O points		Does not occupy the main module I/O points	
Isolation		The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.	

3) Port description



4) Wiring of output signal:

The analog input signal is connected to the input port of the expansion module through a twisted pair cable. When wiring, do not approach the AC power line or a line with serious interference; if the interference on the site is serious, use a shielded wire for signal connection, and connect a 0.1uF/25V high-frequency capacitor at the voltage input port of the controlled device; the shell of the signal source and its shielding wire is connected to the FG terminal of the PLC and the signal ground terminal of the controlled device, and is grounded together.





## 5) The BFM area of the 4DA(R) / 2DA module:

The PLC main module is to write digital values to the register buffer unit (BFM area) of the 4DA(R) / 2DA module, which is converted from 4DA(R) / 2DA to analog output, and the module is set up by rewriting the specific BFM area status. The PLC main module accesses these BFM units through read and write instructions FROM/TO.

The expansion module is equipped with EEPROM storage unit, which is used to save some BFM setting values, such as the signal type, offset value, gain value of each analog input channel, etc. The saving action of these units is determined by the setting status of the corresponding BFM unit and done automatically.

The width of each register in the BFM area is 16-bit (i.e. 1Word), according to the definition of the BFM area of the 4DA(R) / 2DA module as follows: (2DA channels 3 and 4 are reserved)

BFM	R/W attribute	Content			
#0 (E)	WR	Output mode selection, each HEX bit represents 1 input channel, 4DA(R): the highest bit is ch4, the lowest bit is ch1: (default value = H0000); 2DA: the high HEX bit in the lower 8 bits is ch2, The low HEX bit is ch1: (default value = H00) 0=-10V~10V; corresponding digital output: -2000~2000 1=4mA~20mA; corresponding digital output: 0~1000 2=0mA~20mA; corresponding digital output: 0~1000 3=This channel is closed; 4=-10V~10V; corresponding digital output: -10000~10000 5=4mA~20mA; corresponding digital output: 0~10000 6=0mA~20mA; corresponding digital output: 0~10000			
#1	WR	Channel 1	Output value of the channel, initial value 0		
#2	WR	Channel 2			
#3	WR	Channel 3			
#4	WR	Channel 4			
#5 (E)	WR	Value retention mode when PLC is stopped, each HEX bit represents 1 output channel (4DA(R): Hxxxx, the highest bit is CH4, and the lowest bit is CH1; 2DA: Take the high HEX bit in the low 8 bits as ch2, and the low HEX bit as ch1,), when x=0, keep the output before stopping; x=1, reset the output to the offset setting value			
#6-7	-	Reserved			
#8 (E)	WR	G2	O2	G1	O1
		CH2/CH1 offset/gain configuration command. Configured by hex (four binary bits) bits, initial value = H0000 0= Modification not allowed; 1=EEPROM allowed to be modified.			
#9 (E)	WR	G4	O4	G3	O3
		CH3/CH4 offset/gain configuration command. Configured by hex (four binary bits) bits, initial value = H0000 0= Modification not allowed; 1=BFM allowed to be modified.			
#10	WR	Offset data ch1	Unit: mV or uA, initial offset: 0, initial gain: +5000, corresponding mode: 0		
#11	WR	Gain data ch1			
#12	WR	Offset data ch2			
#13	WR	Gain data ch2			
#14	WR	Offset data ch3			
#15	WR	Gain data ch3			
#16	WR	Offset data ch4			
#17	WR	Gain data ch4			

BFM	R/W attribute	Content
#18-19	-	Reserved
#20 (E)	WR	Initial value = 0; when 1 is written, all BFM units restore the initial values to default values.
#21 (E)	WR	1 = Adjusting output (initial value) is allowed; 2 = Adjusting output is not allowed.
#22-26	-	Reserved
#27	R	Extension module software version
#28	-	Reserved
#29	R	Error status word
#30	R	Module ID. The ID of the 4DA module is K3020.
#31	-	Reserved

Note: After the module is powered off, the curve characteristics of the output conversion are saved, including the offset gain characteristics of the curve. The offset gain displayed in the current BFM area is the value to be effective, not the offset gain currently effective.

The meaning of the status information field BFM #29 is as follows:

BFM# 29 bit	ON	OFF
b0:	An error exists. When any value in b1-b3 is not 0, D/A conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal.
b10:	Written value is beyond the acceptable range. (-2350-2350)	Digital output is normal.
b12:	The value of BFM#21 cannot be set to K1.	BFM#21=K1
Bits 4-7, bit 11, and bits 13-15 of BFM#29 are not defined		

#### Register Modification

E in the table indicates that the data is stored in EEPROM. The BFM units carrying E, such as #8, #9, and #20 cannot be frequently written, to prevent the EEPROM memory unit from being damaged.

The modification of the BFM units carrying (E), such as BFM#0 causes the write operation on EEPROM. The write operation requires 3s. In PLC programming, before BFM#10-#17 are modified, a 3s delay is required. The next write operation can be performed only when the previous instruction is complete. There should be a 300 ms interval between write operations.

Note: The external 24 V power supply for the local extension module is lost. The system flag M8067 of the PLC main module is reset. The error code is D8067 = K6708. This symptom can be detected if you often check the flag in editing.

Explanation of part of BFM area:

#### BFM#0 channel selection

The channels are initialized. By default, all the four channels are -10 V to 10 V, which is controlled by the hex string HXXXX of BFM #0. The lowest bit controls channel 1, the second lowest bit controls channel 2, the third lowest bit controls channel 3, and the highest bit controls channel 4. The character control is as follows:

- X=0, pre-defined range: -10 V-10 V (-2000 to 2000)
- X=1, pre-defined range: 4 mA to 20 mA (0 to 1000)
- X=2, pre-defined range: 0 mA to 20 mA (0 to 1000)
- X=4, pre-defined range: -10 V to 10 V (-10000 to 10000)
- X=5, pre-defined range: 4 mA to 20 mA (0 to 10000)
- X=6, pre-defined range: 0 mA to 20 mA (0 to 10000)

For example, if BFM#0 is H1220, channel 1 is -10 V-10 V, channel 2 and channel 3 are 0 mA-20 mA, and channel 4 is 4 mA-20 mA.

#### BFM #1 to #4 Channel output value

To control analog output, use the TO instruction to write data to BFM #1 to #4. The initial values are 0.

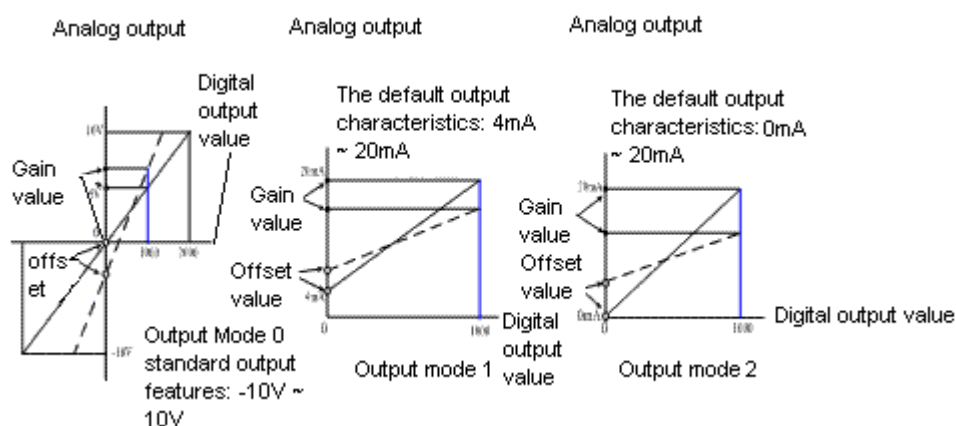
#### BFM #5 Data retain mode

When the main module status changes from RUN to STOP, the last mode of RUN status is retained (X=0) or the offset (X=1) output.

For example, when BFM#0 is H0000 and BFM#5 is H0101, the offsets of the 4 channels are 0.1V. When the status changes from RUN to STOP, all values of BFM#1 to #4 are 1500 (7.5 V). When the status changes to STOP, the output voltages of channels 1 and 3 are 0.1 V, and the output voltages of channels 2 and 4 are 7.5 V.

#### BFM #8 to #17, BFM #21 Definition and setting method of offset and gain values:

H2U-4DA(R) / 2DA has three working modes. The curve is as follows:



The gain is the analog output of digital 1000; the offset is the analog output of digital 0. The offset and gain values can be set separately or together. The units are mV (mode 0) and  $\mu$ A (modes 1 and 2).

BFM #8 to #9 are the offset/gain configuration commands. In the hex format, each HEX bit (4 binary bits) controls prohibit and permit. Note that it is controlled by each bit in the AD module. The offset and gain configuration commands for the DA and AD modules are different. BFM #10 to #17 are the offset and gain values. BFM #21 is the curve setting command.

Before setting the gain/offset, set BFM#21 to 1, and modify the values of BFM#10-#17. Write the

permission word to BFM#8 and #9. After the modification, set BFM#21 to 2 again, to avoid change.

For example, when BFM#0 is in mode H0000, set the offset and gain of channel 1 to 0.2 V and 5.5 V, respectively, and set the offset and gain of channel 3 to 0.5 V and 6 V, respectively. The operation is as follows:

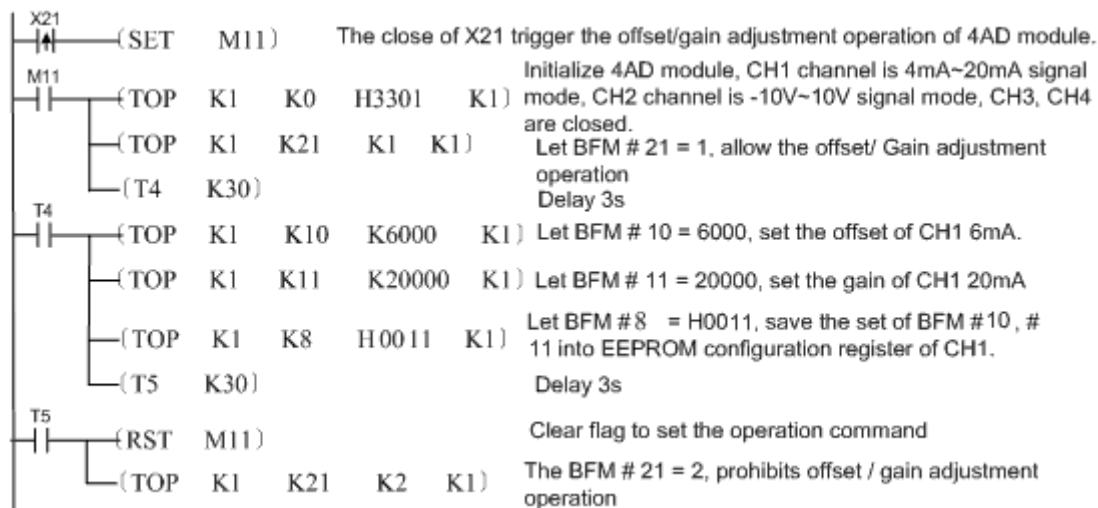
Change BFM#21 to 1. After 300 ms, send K200, K5500, K500, and K6000 to BFM#10, #11, #14, and #15, respectively. Allow offset and gain in BFM#8 to #9, that is, change BFM#8 and BFM#9 to H0011. The modification is complete. Then change BFM#21 to 2, to prevent the data from being modified again.

BFM #20 restores the factory setting.

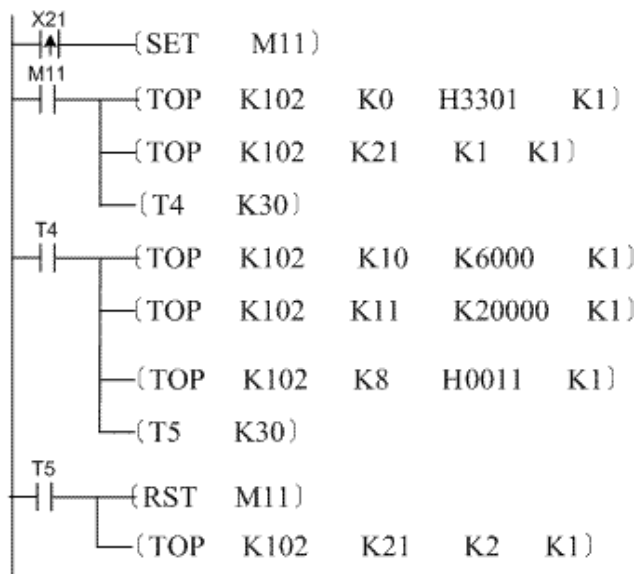
To restore the default setting, set BFM#20 to 1.

Programming example 1:

After an H2U-4AD module is inserted behind the PLC main module, an H2U-4DA module is inserted behind them. According to the numbering rule, the number of H2U-4DA is #1. It is required that CH1 outputs 6 mA-20 mA current signal, CH2 outputs -10 V-10 V voltage signal, and CH3/CH4 are not in use. It is required that the X21 port close trigger the initialization of the 4DA module. The user program is as follows:

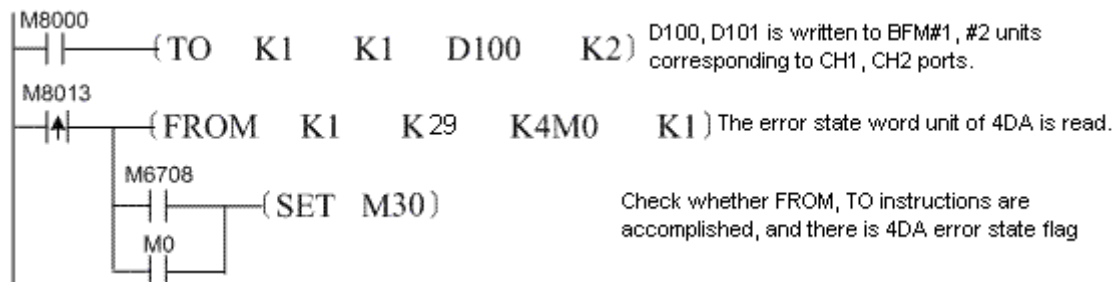


In this example, if the module is changed to H2U-4DAR, and the CAN station number is 2, then the program in this example is as follows:



## Programming example 2:

In the previous example, it is required to output D100 and D101 through the CH1 and CH2 ports. The working status of the 4DA module is checked every second. The programming is as follows:



If the external 24 V power supply for the 4DA module is lost, the system flag M8067 is reset. The error code is D8067 = K6708.

## Precautions:

Online removal or installation is forbidden. The extension module can be connected or removed only when the main module and application systems are powered off; otherwise, there may be a harm to human body or the components may be damaged.

## 6.5 Analog hybrid (input/output) module

### 6.5.1 H2U-4AM 2-channel analog current input/2-channel analog voltage and current output hybrid expansion module

#### 1) Introduction:

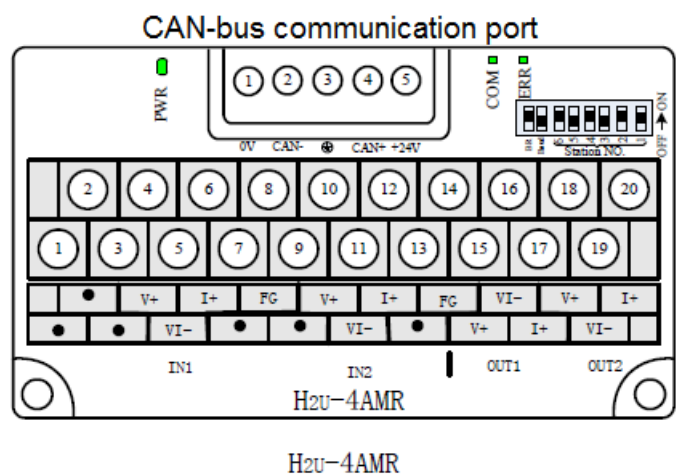
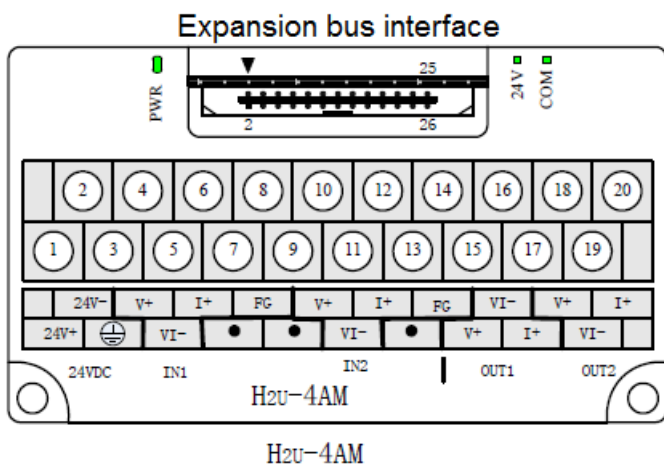
The H2U local expansion module can work with the H2U series main module to realize the signal detection of 2 analog input channels and the output of 2 analog signal channels. Each input channel converts 0V~10V or 0mA~20mA signal into 12-bit digital quantity for PLC main module to read; each output channel has voltage signal and current signal output ports; the signal amplitude can be -10~10 or 0~20mA. The main module accesses the BFM unit of the register in the expansion module through the FROM/TO instruction to control the analog output signal.

#### 2) Electrical specifications:

Item		Description	
Power supply	Analog	24V DC -15%/+20%, maximum allowable ripple voltage 5%, current consumption 100mA (taken from 24V/COM of the main module, or other 24VDC power supply)	
	Digital	5V DC 50mA (through the module extension cable, taken from the main module power supply internal power supply)	
Item		Index for AD	
Occupied I/O points		Does not occupy the main module I/O points	
Conversion speed		15ms/channel (normal speed), 6ms/channel (fast), 1ms/channel (fastest)	
Analog input range	Voltage input	-10~10V DC, (input impedance is 200KΩ)	The user needs to set the corresponding BFM area according to the type of signal to be connected.
	Current input	0~20mA (The external load impedance is 250Ω or less)	

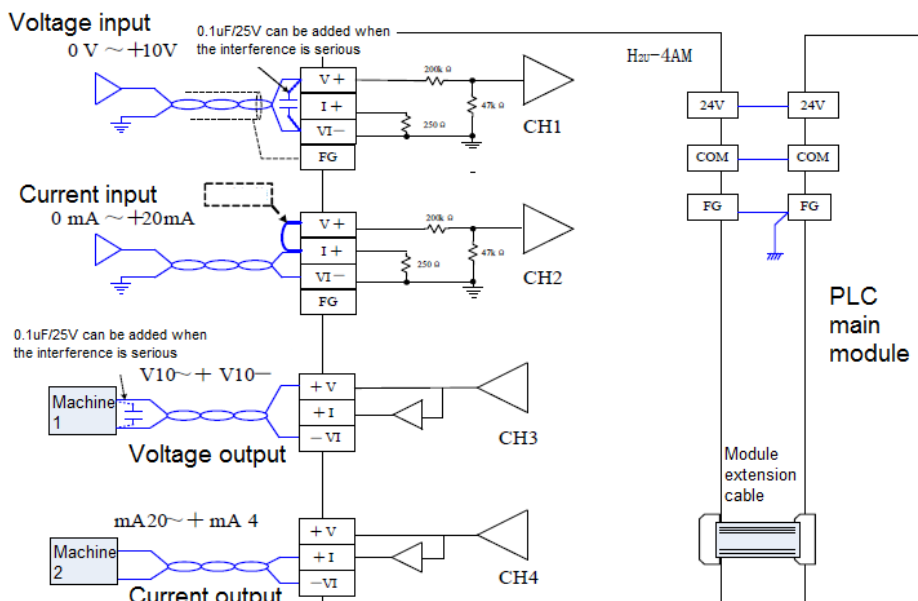
Item		Index for AD
Digital output		Default setting: 0 ~ +2000
Resolution	Voltage input	5mV
	Current input	20 $\mu$ A
Precision		$\pm 1\%$ full range
Isolation		The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.
Item		Index for DA
Occupied I/O points		Does not occupy the main module I/O points
Conversion speed		2 channels 2.1ms
Analog input range	Voltage input	-10~10V DC, (input impedance is 2K $\Omega$ ~ 1M $\Omega$ )
	Current input	0~20mA (The external load impedance is 500 $\Omega$ or less)
Digital output		Default setting: -2000 ~ 2000, allowable range: -10000 ~ 10000
Resolution	Voltage input	5mV (10V\2000)
	Current input	20 $\mu$ A (20mA\1000)
Precision		$\pm 1\%$ (for the full range of 10V) $\pm 1\%$ (for full range of 20mA)
Isolation		The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.

## 3) Port description:



## External wiring:

The analog signal is connected to the input/output port of the expansion module through a twisted pair cable. When wiring, do not close to the AC power line or the line of interference signals; if the interference of the analog signal is serious, use a shielded and connect a 0.1 $\mu$ f/25V high-frequency capacitor at the voltage input port of the controlled device; the shell of the signal source/controlled component and its shielding wire is connected to the signal ground terminal FG of H2U-4AM, and is grounded together.



4) Access the BFM area of 4AM (R):

The PLC main module accesses the BFM units using the FROM/TO instruction.

The extension module has the built-in EEPROM memory unit, for saving the BFM values, for example, the signal type, offset value, and gain value. The unit saving action is implemented by the setting of the corresponding BFM units.

The width of each register in the BFM area is 16 bits, that is, 1 Word. The BFM area definitions according to the 4AM (R) module is as follows:

BFM	R/W attribute	Input channel content	
#0 (E)	WR	Channel signal mode selection, each HEX bit represents 1 input channel, 4AM module takes the high HEX bit in the low 8 bits as ch2, and the low HEX bit as ch1: (default = H00) 0=0V~10V; corresponding digital output: 0~2000 1=4mA~20mA; corresponding digital output: 0~1000 2=0mA~20mA; corresponding digital output: 0~1000 3=This channel is closed; 4=0V~10V; corresponding digital output: 0~10000 5=4mA~20mA; corresponding digital output: 0~10000 6=0mA~20mA; corresponding digital output: 0~10000	
#1	WR	Channel 1	Average filter constant, namely, the number of sampled values for average value calculation. Range: 1-4096; default: 8. The value can be set to 1 if high-speed collection is needed. When BFM#15 is changed, the value is restored to the default.
#2	WR	Channel 2	
#3-4	-	Reserved	
#5	R	Channel 1	Mean filtering value of the sampled value of the input channel
#6	R	Channel 2	
#7-8	-	Reserved	
#9	R	Channel 1	Currently sampled data of the input channel, namely, burst value without filtering
#10	R	Channel 2	

BFM	R/W attribute	Input channel content								
#11-14	-	Reserved								
#15	WR	ADC speed selection	0 = normal speed, 15ms/channel (default value); 1 = fast conversion, 6ms/channel; 1000-30000 = high-speed sampling, corresponding to 1 ms-30ms/channel							
#16-19	-	Reserved								
#20 (E)	WR	1 = reset; Restore the parameters of all channels (BFM#0 to BFM#32) to the default (factory setting). Default = 0								
#21 (E)	WR	2 = Adjusting offset/gain values is not allowed (default); 1= Adjusting offset/gain values is allowed (default)								
#22 (E)	WR	Operations for the two channels corresponding to the lower 4 bits	-	-	-	-	G2	O2	G1	O1
		Offset/gain value adjusting is enabled. When the value is non-0, the module writes the BFM 23/24 value into the control register of the internal channel.								
#23 (E)	WR	Offset value. Analog input value (mode 0, 1, or 2) when the digital output is 0.								
#24 (E)	WR	Gain value. Digital input value (mode +1000, 1, or 2) when the digital output is +1000.								
#25-26	-	Reserved								
#27	R	4AM module software version								
#28	-	Reserved								
#29	R	Input channel error status								
#30	R	ID of the extension module. The ID of H2U-4AM is K2010.								
#31	-	Reserved, not accessible								
#32 (E)	WR	Output mode selection. Each hex bit indicates one output channel. The highest hex bit in the lowest 8 bits of the 4AM(R)0 = -10 V to 10 V; corresponding digital output: -2000-2000 1 = 4 mA-20 mA; corresponding digital output: 0-1000 2 = 0 mA-20 mA; corresponding digital output: 0-1000 4 = -10 V-10 V; corresponding digital output: -10000-10000 5 = 4 mA-20 mA; corresponding digital output: 0-10000 6 = 0 mA-20 mA; corresponding digital output: 0-10000								
#33	WR	Channel 3				Output value of the channel, initial value 0				
#34	WR	Channel 4								
#35-36	-	Reserved								
#37 (E)	WR	Value retain mode when PLC is stopped. Each hex bit indicates a channel. The highest hex bit of the lower 8 bits of 4AM(R) is ch4, and the lowest hex bit is ch3. When x = 0, retain the output before stop; x=1, reset output to the offset value								
#38-39	-	Reserved								
#40 (E)	WR	G2		O2		G1		O1		
		CH4/CH3 offset/gain configuration command. Configured by hex bits, initial value = H0000 0= Modification not allowed; 1 = EEPROM allowed to be modified.								
#41	-	Reserved								
#42	WR	Offset data ch3							Unit: mV or uA, initial offset: 0, initial gain:	
#43	WR	Gain data ch3								



BFM	R/W attribute	Input channel content									
#44	WR	Offset data ch4								+5000, corresponding mode: 0 (mode 0, 1, 2)	
#45	WR	Gain data ch4									
#46-51	-	Reserved									
#52 (E)	-	Reserved									
#53 (E)	WR	1 = Adjusting output (initial value) is allowed; 2 = Adjusting output is not allowed.									
#54-60	-	Reserved									
#61	R	Output channel error status									

The meaning of the status information field BFM #29 is as follows:

BFM# 29 bit	ON	OFF
b0:	An error exists. When any value in b0-b3 is not 0, A/D conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal:
b10:	Digital output exceeds the range of -2048-2047.	Digital output is normal.
b11:	The sampling filter constant exceeds the range of 1-4096.	The sampling filter constant is normal.
b12:	The value of BFM#21 cannot be set to K2.	The value of BFM#21 can be set to K2.
Bits 4-7 and bits 13-15 of BFM#29 are not defined.		

The meaning of the status information field BFM #61 is as follows:

BFM# 61 bit	ON	OFF
b0:	An error exists. When any value in b1-b3 is not 0, D/A conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal.
b10:	Written value is beyond the acceptable range.	Digital output is normal.
b12:	The value of BFM#21 cannot be set to K1.	BFM#21=K1
Bits 4-7, bit 11, and bits 13-15 of BFM#61 are not defined.		

In the table, the BFM unit with E is the project stored in EEPROM. It will not be deleted upon power off.

#### Register Modification

The modification of the BFM units carrying (E), such as BFM#0 and #32, and #24 causes the write operation on EEPROM. In the write operation, each Word requires about 300ms. When the write operation is performed on multiple BFM units carrying (E), the PLC programming requires a delay after each BFM unit is written. The next write operation can be performed only when the previous instruction is complete.

Note: The external 24 V power supply for the local extension module is lost. The system flag M8067 of the PLC main module is reset. The error code is D8067 = K6708. This symptom can be detected if you often check the flag in editing.

#### 5) Explanation of input channel of BFM area

##### BFM#0 input channel selection

The channels are initialized. By default, all the two channels are 0 to 10 V, which is controlled by the hex string HXX of BFM #0. The lowest bit controls channel 1, and the highest bit controls channel 2. The character control is as follows:

- X=0, pre-defined range: 0 V to 10 V (0 to 2000)
- X=1, pre-defined range: 4 mA to 20 mA (0 to 10000)
- X=2, pre-defined range: 0 mA to 20 mA (0 to 10000)
- X=3, the channel is closed.
- X=4, pre-defined range: 0 V to 10 V (0 to 10000)
- X=5, pre-defined range: 4 mA-20 mA (0 to 10000)
- X=6, pre-defined range: 0 mA-20 mA (0 to 10000)

For example, if BFM#0 is H30, channel 1 is 0 V-10 V and channel 2 is closed.

The unused channel can be closed or not. The closed channel does not occupy conversion time (BFM#15). In the example, channel 2 is closed, so the total conversion time is the conversion time of channel 1 x BFM#15. BFM #1 to #2 average sampling value.

The sampling value corresponding to (BFM#9 to #10) of each channel plus the number of samples (BFM #1 to #2) divided the number of samples (BFM #1 to #2) are stored in (BFM #5 to #6).

BFM #5 to #6 stores the average sampling values.

BFM #9 to #10 stores the instant sampling values.

##### BFM #15 ADC rate

Time of one conversion of each channel. The time of each data update is BFM#15 x number of opened channels.

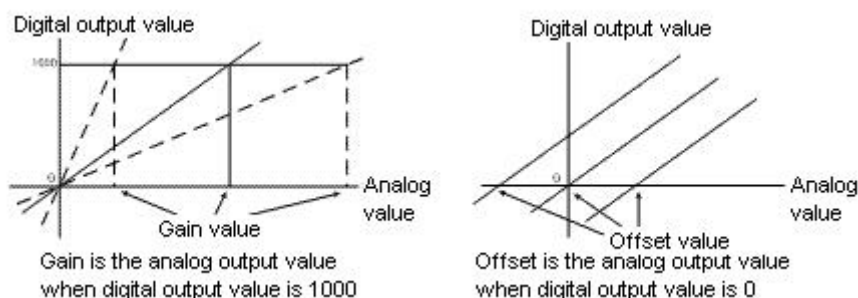
For example, when BFM #0 is H10, BFM #1 is K7, BFM #2 is K6, BFM #15 is K10, the time of one data update for BFM #9 and BFM #10 is  $\text{BFM}\#0 \times \text{BFM}\#15 = 2 \times 10 = 20\text{ms}$ , the time of one data update for BFM #5 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#1 = 2 \times 10 \times 7 = 140\text{ms}$ , and the time of one data update for BFM #6 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#2 = 2 \times 10 \times 6 = 120\text{ms}$ . The FROM/TO instruction is time consuming, so the BFM#5 data collected by this module can be read using the [LDP M8012] [FROM K0 K5 D10 K1] instruction, which has the same effect as using [LD M8000] [FROM K0 K5 D10 K1]. However, the data is read in each scanning period of the M8000 drive.

BFM #20 restores the factory setting of the output channel.

To restore the default setting, set #20 to 1.

BFM #21 to #24 definition and setting method of offset and gain values:

H2U-4AM(R) has three working modes. The curve is as follows:



The offset and gain values can be set separately or together. The normal gain value ranges from 1 V to 15 V or 4 mA to 32 mA. The normal offset value ranges from 0 V to 5 V or -0 mA to 20 mA.

Before setting the offset/gain, set BFM#21 to 1, modify BFM#23/24, and then allow offset/gain BFM#22 for each channel. After the modification, set BFM#21 to 2 again, to avoid change.

Note that the channels requiring offset/gain must be the same. It is not allowed that the offset of one channel is 1000 but the offset of another channel is 1200.

For example, when BFM#0 is in mode 0, set the offset and gain of channel 1 and channel 2 to 0.5V and 6V, respectively. The operation is as follows:

Change BFM#21 to 1. After 300ms, send K500 and K6000 to BFM#23 and BFM#24 respectively. After 300ms, allow gain BFM#22. In this example, BFM#22 should be binary form 1111. That is, change BFM#22 to HF. The modification is complete. Then change BFM#21 to 2, to prevent from the data being modified again.

#### 6) Explanation of input channel of BFM area

##### BFM#32 output channel selection

The channels are initialized. By default, both the two channels are -10 V to 10 V, which is controlled by the hex string HXXXX of BFM 32#. The lowest bit controls channel 3, and the second bit controls channel 4. The character control is as follows:

- X=0, pre-defined range: -10 V to 10 V (-2000 to 2000)
- X=1, pre-defined range: 4 mA to 20 mA (0 to 10000)
- X=2, pre-defined range: 0 mA to 20 mA (0 to 10000)
- X=4, pre-defined range: -10 V to 10 V (-10000 to 10000)
- X=5, pre-defined range: 4 mA-20 mA (0 to 10000)
- X=6, pre-defined range: 0 mA-20 mA (0 to 10000)

For example, when BFM#32 is H10, channel 3 is -10 V to 10 V and channel 4 is 4 mA to 20 mA.

##### BFM #33 to #34 channel output value

To control analog output, use the TO instruction to write data to BFM #33 to #34. The initial values are 0.

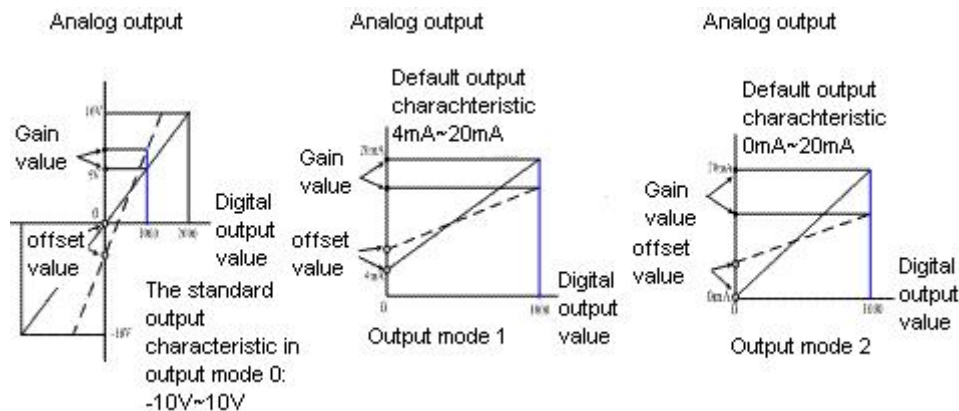
##### BFM #37 data retain mode of the output channel

When the main module status changes from RUN to STOP, the last mode of RUN status is retained (X=0) or the offset (X=1) output.

For example, when BFM#32 is H00 and BFM#37 is H01, the offset values of 2 channels are 0.1 V. When the status changes from RUN to STOP, the values of BFM#33~#34 are 1500 (7.5 V). When the status changes to STOP, the output voltage of 3 is changed to 0.1 V and the output voltage of channel 4 retains 7.5 V.

BFM #40, BFM #42 to #45, BFM #53 Definition and setting method of offset and gain values:

H2U-4AM(R) has three working modes. The curve is as follows:



The gain is the analog output of digital 1000; the offset is the analog output of digital 0. The offset and gain values can be set separately or together. The units are mV (mode 0) and  $\mu$ A (modes 1 and 2).

BFM #40 are the offset/gain configuration commands. In the hex format, each HEX bit (4 bits) controls prohibit and permit. Note that it is controlled by each bit in the AD module. The offset and gain configuration commands for the DA and AD modules are different. BFM #10 to #13 are the offset and gain values. BFM #53 is the curve setting command.

Before setting the gain/offset, set BFM#53 to 1, and modify the values of BFM#42-#45. Write the permission word to BFM#40. After the modification, set BFM#53 to 2 again, to avoid change. For example, when BFM#40 is in mode H00, set the offset and gain of channel 3 to 0.2V and 5.5V, respectively. The operation is as follows:

Change BFM#53 to 1. After 300ms, send K200 and K5500 to BFM#42 and #43, respectively. Allow offset and gain in BFM#40, that is, change BFM#40 to H0011. The modification is complete. Then change BFM#53 to 2, to prevent the data from being modified again.

BFM #52 factory restore setting of the output channel.

To restore the default setting, set BFM#52 to 1.

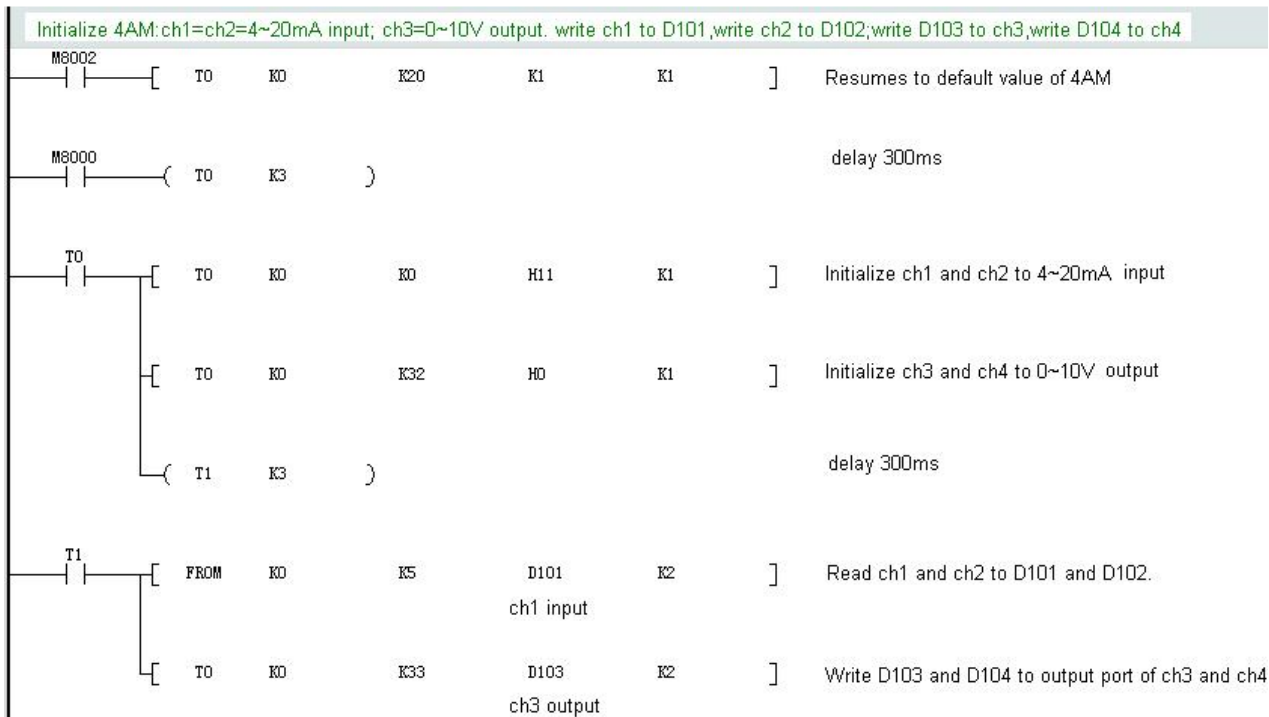
Programming example 1:

Connect 2 input channels (ch1 and ch2) of the 4AM module to 4-20 mA signal. The collected data range is 0-1000. Two output channels are configured to output 0-10 V signals, corresponding to digital 0-2000.

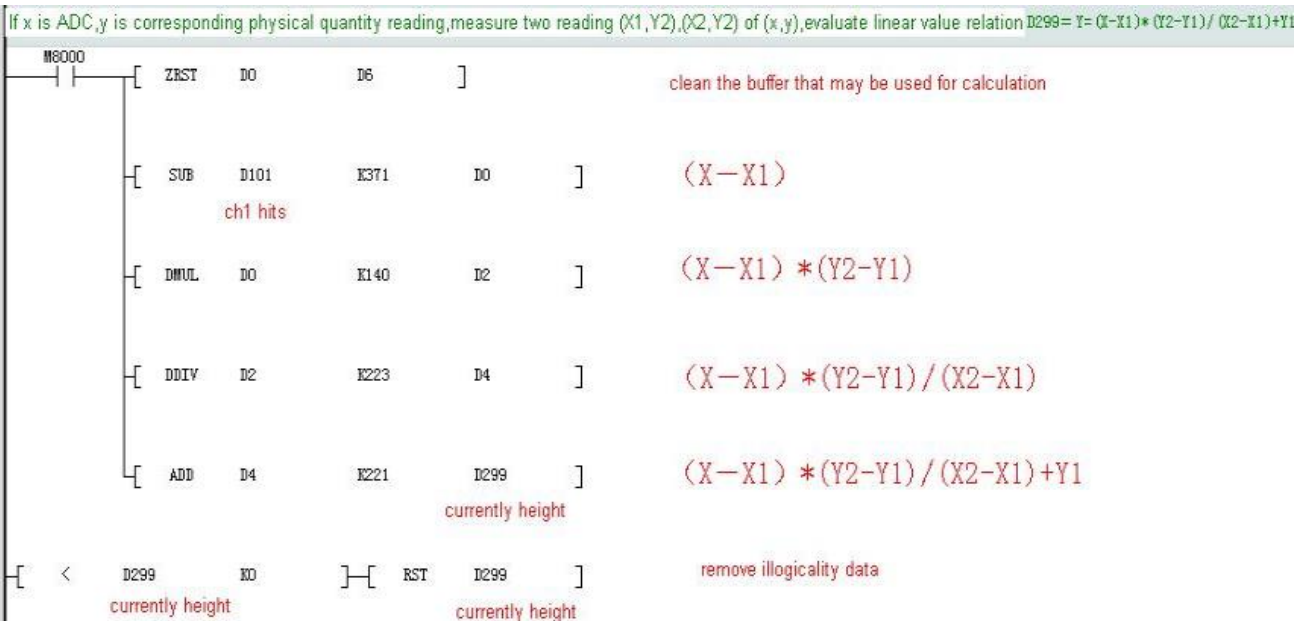
According to the BFM#0 definition of ch1 and ch2, it should be configured as BFM#0=H0011.

According to the BFM#32 definition of ch3 and ch4, it should be configured as BFM#32=H0000.

The programming is as follows:



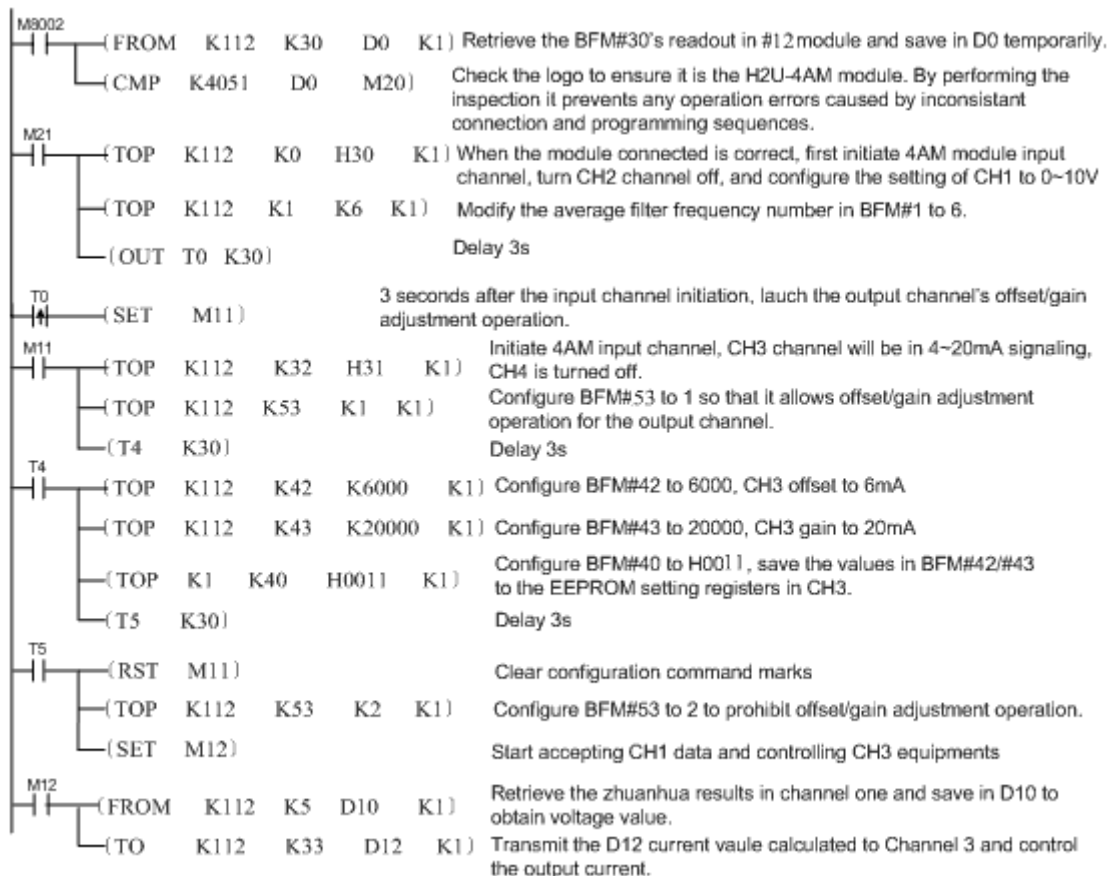
The linear calculation needs to be performed for the ADC, to get the value consistent with the actual material quantity. Generally, the values are calculated at the two ends of the material quantity, and the linear interpolation is used to correct the values. Use the DMUL and DDIV clauses to avoid calculation overflowed.



The D299 value is get by the ch1 value (D101) of 4AM.

Programming example 2:

There is a remote extension module H2U-4AMR. The CAN station address is 12. The CH1 port needs to collect the 0 V-10 V voltage signal. It is required to change the filtering times to 6, and store the collected data to D10. The CH3 outputs 6 mA-20 mA current signal through D12. The user program is as follows:



Precautions:

Online removal or installation is forbidden. The extension module can be connected or removed only when the main module and application systems are powered off; otherwise, there may be a harm to human body or the components may be damaged.

6.5.2 H2U-6AM 4-channel analog current input/2-channel analog voltage and current output hybrid expansion module

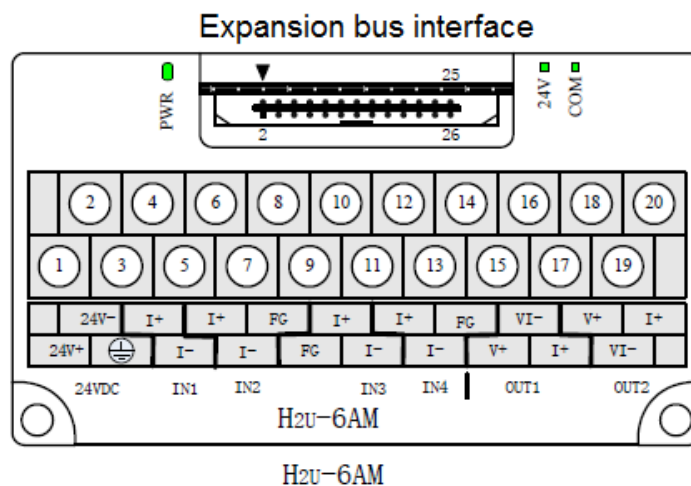
1) Introduction

The H2U local expansion module can work with the H2U series main module to realize the signal detection of 4 analog input channels and the output of 2 analog signal channels. Each input channel converts 0mA~20mA signal into 12bit digital quantity for PLC main module to read; each output channel has voltage signal and current signal output port, the signal amplitude can be -10V~10V or 0~20mA respectively. The main module accesses the BFM unit of the register in the expansion module through the FROM/TO instruction to control the analog output signal.

## 2) Electrical specifications:

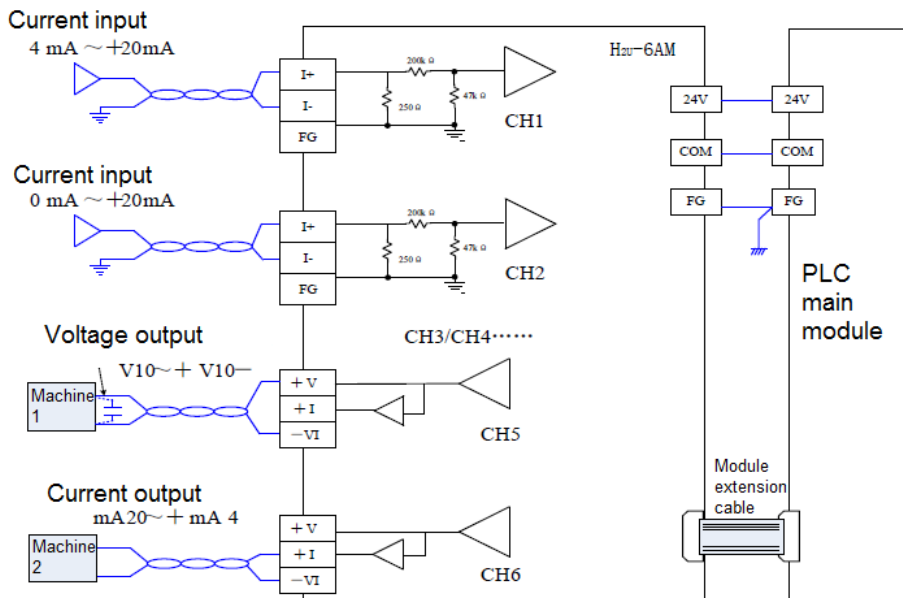
Item		Description
Power supply	Analog	24V DC -15%/+20%, maximum allowable ripple voltage 5%, current consumption 100mA (taken from 24V/COM of the main module, or other 24VDC power supply)
	Digital	5V DC 50mA (through the module extension cable, taken from the main module power supply internal power supply)
Item		Index for AD
Occupied I/O points		Does not occupy the main module I/O points
Conversion speed		15ms/channel (normal speed), 6ms/channel (fast), 1ms/channel (fastest)
Analog input range	Current input	0~20mA (The external load impedance is 250Ω or less)
Digital output		Default setting: 0 ~ +1000
Resolution	Current input	20μA
Precision		±1% full range
Isolation The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.		
Item		Index for DA
Occupied I/O points		Does not occupy the main module I/O points
Conversion speed		2 channels 2.1ms
Analog input range	Voltage input	-10~10V DC, (input impedance is 2KΩ ~ 1MΩ)
	Current input	0~20mA (The external load impedance is 500Ω or less)
Digital output		Default setting: -2000 ~ 2000, allowable range: -10000 ~ 10000
Resolution	Voltage input	5mV (10V\2000)
	Current input	20μA (20mA\1000)
Precision		±1% (for the full range of 10V) ±1% (for full range of 20mA)
Isolation The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.		

## 3) Port description:



4) External wiring:

The analog signal is connected to the input/output port of the expansion module through a twisted pair cable. When wiring, do not close to the AC power line or interference signal line; if the interference of the analog signal is serious, a shielded and connect a 0.1uf/25V high-frequency capacitor at the voltage input port of the controlled device; the shell of the signal source/controlled component and its shielding wire is connected to the signal ground terminal FG of H2U-6AM, and is grounded together.



5) Access the BFM area of 6AM (R):

The PLC main module accesses the BFM units using the FROM/TO instruction.

The extension module has the built-in EEPROM memory unit, for saving the BFM values, for example, the signal type, offset value, and gain value. The unit saving action is implemented by the setting of the corresponding BFM units.

The width of each register in the BFM area is 16 bits, that is, 1 Word. The BFM area definitions according to the 6AM (R) module is as follows:

BFM	R/W attribute	Input channel content	
#0 (E)	WR	Signal mode selection of channel. Each hex bit indicates one input channel. The highest hex bit of the lower 8 bits in the 4AM(R) module is ch2, and the lowest hex bit is ch1 (default is H1111). 1 = 4 mA-20 mA; corresponding digital output: 0-1000 2 = 0 mA-20 mA; corresponding digital output: 0-1000 3 = local channel is closed; 5 = 4 mA-20 mA; corresponding digital output: 0-1000 6 = 0 mA-20 mA; corresponding digital output: 0-1000	
#1	WR	Channel 1	Average filter constant, namely, the number of sampled values for average value calculation. Range: 1-4096; default 8. The value can be set to 1 if high-speed collection is needed. When BFM#15 is changed, the value is restored to the default.
#2	WR	Channel 2	
#3	WR	Channel 3	
#4	WR	Channel 4	
#5	R	Channel 1	Mean filtering value of the sampled value of the input channel
#6	R	Channel 2	



BFM	R/W attribute	Input channel content								
#7	R	Channel 3	Currently sampled data of the input channel, namely, burst value without filtering							
#8	R	Channel 4								
#9	R	Channel 1								
#10	R	Channel 2								
#11	R	Channel 3								
#12	R	Channel 4								
#13-14	-		Reserved							
#15	WR	ADC speed selection	0 = normal speed, 15 ms/channel (default value); 1 = fast conversion, 6ms/channel; 1000-30000 = high-speed sampling, corresponding to 1ms-30ms/channel							
#16-19	-	Reserved								
#20 (E)	WR	1 = reset; Restore the parameters of all channels (BFM#0~BFM#32) to the default (factory setting). Default = 0								
#21 (E)	WR	2 = Adjusting offset/gain values is not allowed; 1 = Adjusting offset/gain values is allowed (Default value)								
#22 (E)	WR	Operations for the four channels corresponding to the lower 8 bits	G4	O4	G3	O3	G2	O2	G1	O1
		Offset/gain value adjusting is enabled. When the value is non-0, the module writes the BFM 23/24 value into the control register of the internal channel.								
#23 (E)	WR	Offset value. When the digital output is 0, the initial value of the analog input value (mode 0, 1, or 2) is 400.								
#24 (E)	WR	Gain value. When the digital output is +1000, the initial value of the analog input value (mode 0, 1, or 2) is 20000.								
#25-26	-	Reserved								
#27	R	6AM module software version								
#28	-	Reserved								
#29	R	Input channel error status								
#30	R	ID of the extension module. The ID of H2U-6AM is K2010.								
#31	-	Reserved, not accessible								

#32 (E)	WR	Output mode selection. Each hex bit indicates one output channel. The highest hex bit in the lowest 8 bits of the 6AM(R) is ch6, and the lowest hex bit is ch5: (default = H00) 0 = -10 V to 10 V; corresponding digital output: -2000-20001 = 4 mA-20 mA; corresponding digital output: 0-10002 = 0 mA-20 mA; corresponding digital output: 0-10004 = -10 V-10 V; corresponding digital output: -10000-10000 5 = 4 mA-20 mA; corresponding digital output: 0-100006 = 0 mA-20 mA; corresponding digital output: 0-10000								
#33	WR	Channel 5	Output value of the channel, initial value 0							
#34	WR	Channel 6								
#35-36	-	Reserved								
#37 (E)	WR	Value retain mode when PLC is stopped. Each hex bit indicates an output channel. The highest hex bit of the lower 8 bits of 6AM(R) is ch6, and the lowest hex bit is ch5. When x=0, retain the output before stop; x=1, reset output to the offset value								
#38-39	-	Reserved								
#40 (E)	WR	Operations for the two channels corresponding to the lower 4 bits	G2	O2	G1	O1				

BFM	R/W attribute	Input channel content	
		CH6/CH5 offset/gain configuration command. Configured by hex bits, initial value = H0 0 = Modification not allowed; 1 = EEPROM allowed to be modified.	
#41	-	Reserved	
#42	WR	Offset data ch5	Unit: mV or uA, initial offset: 0, initial gain: +5000, corresponding mode: 0 (mode 0, 1, 2)
#43	WR	Gain data ch5	
#44	WR	Offset data ch6	
#45	WR	Gain data ch6	
#46-51	-	Reserved	
#52 (E)	WR	Initial value = 0; when 1 is written, all BFM units (BFM#32 to BFM#64) restore the initial values to default values.	
#53 (E)	WR	1 = Adjusting output (initial value) is allowed; 2 = Adjusting output is not allowed.	
#54-60	-	Reserved	
#61	R	Output channel error status	

The meaning of the status information field BFM #29 is as follows:

BFM# 29 bit	ON	OFF
b0:	An error exists. When any value in b0-b3 is not 0, A/D conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal:
b10:	Digital output exceeds the range of -2048-2047.	Digital output is normal.
b11:	The sampling filter constant exceeds the range of 1-4096.	The sampling filter constant is normal.
b12:	The value of BFM#21 cannot be set to K2.	The value of BFM#21 can be set to K2.
Bits 4-7 and bits 13-15 of BFM#29 are not defined.		

The meaning of the status information field BFM #61 is as follows:

BFM# 61 bit	ON	OFF
b0:	An error exists. When any value in b1-b3 is not 0, D/A conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal.
b10:	Written value is beyond the acceptable range.	Digital output is normal.
b12:	The value of BFM#21 cannot be set to K1.	BFM#21 = K1
Bits 4-7, bit 11, and bits 13-15 of BFM#61 are not defined.		

In the table, the BFM unit with E is the project stored in EEPROM. It will not be deleted upon power off.

#### Register Modification

The modification of the BFM units carrying (E), such as BFM#0 and #32, and #24 causes the write operation on EEPROM. In the write operation, each Word requires about 300ms. When the write operation is performed on multiple BFM units carrying (E), the PLC programming requires a delay after each BFM unit is written. The next write operation can be performed only when the previous instruction is complete.

#### 6) Explanation of input channel of BFM area:

##### BFM#0 input channel selection

The channels are initialized. By default, all the four channels are 4 to 20 mA, which is controlled by the hex string HXXXX of BFM #0. The lowest bit controls channel 1, and the highest bit controls channel 4. The character control is as follows:

X=1, pre-defined range: 4mA to 20mA (0 to 1000)

X=2, pre-defined range: 0-20mA (0 to 1000)

X=3, the channel is closed.

X=5, pre-defined range: 4mA to 20mA (0 to 10000)

X=6, pre-defined range: 0 to 20mA (0 to 10000)

For example, if BFM#0 is H2235, channel 1 is 4 mA-20 mA, channel 2 is closed, and channel 3 and channel 4 are 4 mA-20 mA.

The unused channel can be closed or not. The closed channel does not occupy conversion time (BFM#15). In the example, channel 2 is closed, so the total conversion time is the sum of conversion time of the other three channels, namely, (3 X BFM#15).

##### BFM #1 to #4 average sampling value of the input channel.

The sampling value corresponding to (BFM#9 to #12) of each channel plus the number of samples (BFM #1 to #4) divided the number of samples (BFM #1 to #4) are stored in (BFM #5 to #8).

BFM #5 to #8 stores the average sampling values of the input channel.

BFM #9 to #12 stores the instant sampling values of the input channel.

##### BFM #15 ADC rate of the input channel

Time of one conversion of each channel. The time of each data update is BFM#15 x number of opened channels.

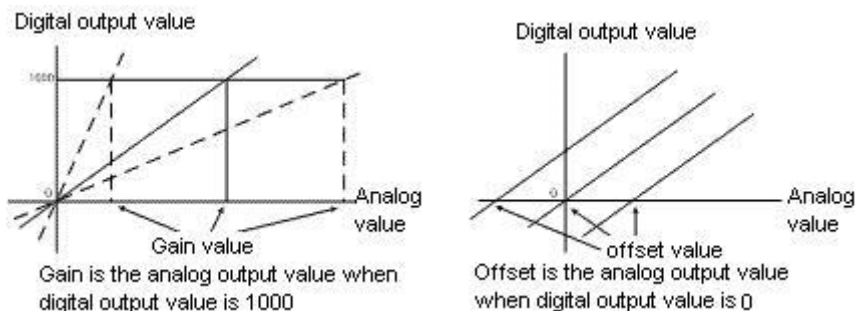
For example, when BFM #0 is H3311, BFM #1 is K7, BFM #2 is K6, and BFM #15 is K10, the time of one data update of BFM #9 and BFM #10 is  $\text{BFM}\#0 \times \text{BFM}\#15 = 2 \times 10 = 20\text{ms}$ , the time of one data update of BFM #5 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#1 = 2 \times 10 \times 7 = 140\text{ms}$ , and the time of one date update of BFM #6 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#2 = 2 \times 10 \times 6 = 120\text{ms}$ . The FROM/TO instruction is time consuming, so the BFM#5 data collected by this module can be read using the [LDP M8012] [FROM K0 K5 D10 K1] instruction, which has the same effect as using [LD M8000] [FROM K0 K5 D10 K1]. However, the data is read in each scanning period of the M8000 drive.

##### BFM #20 factory restore setting of the output channel.

To restore the default setting, set #20 to 1.

##### BFM #21 to #24 definition and setting method of offset and gain values:

H2U-6AM(R) has three working modes. The curve is as follows:



The offset and gain values can be set separately or together. The normal gain value ranges from 4 mA to 32 mA. The normal offset value ranges from -20 mA to 20 mA.

Before setting the offset/gain, set BFM#21 to 1, modify BFM#23/24, and then allow offset/gain BFM#22 for each channel. After the modification, set BFM#21 to 2 again, to avoid change.

Note that the channels requiring offset/gain must be the same. It is not allowed that the offset of one channel is 1000 but the offset of another channel is 1200.

For example, when BFM#0 is in mode 1111, set the offset and gain of channel 1 and channel 2 to 5 mA and 18 mA, respectively. The operation is as follows:

Change BFM#21 to 1. After 300ms, send K5000 and K18000 to BFM#23 and BFM#24 respectively. After 300ms, allow gain BFM#22. In this example, BFM#22 should be binary form 00001111. That is, change BFM#22 to H.000F. The modification is complete. Then change BFM#21 to 2, to prevent the data from being modified again.

#### 7) Explanation of output channel of BFM area:

##### BFM#32 output channel selection

The channels are initialized. By default, all the two channels are -10 V to 10 V, which is controlled by the hex string HXXXX of BFM 32#. The lowest bit controls channel 5, and the second bit controls channel 6. The character control is as follows:

- X=0, pre-defined range: -10 V-10 V (-2000 to 2000)
- X=1, pre-defined range: 4 mA to 20 mA (0 to 1000)
- X=2, pre-defined range: 0 mA to 20 mA (0 to 1000)
- X=4, pre-defined range: -10 V to 10 V (-10000 to 10000)
- X=5, pre-defined range: 4 mA to 20 mA (0 to 10000)
- X=6, pre-defined range: 0 mA to 20 mA (0 to 10000)

For example, if BFM#32 is H10, channel 5 is -10 V-10 V and channel 6 is 4 mA-20 mA.

##### BFM #33 to #34 channel output value

To control analog output, use the TO instruction to write data to BFM #33 to #34. The initial values are 0.

##### BFM #37 data retain mode of the output channel

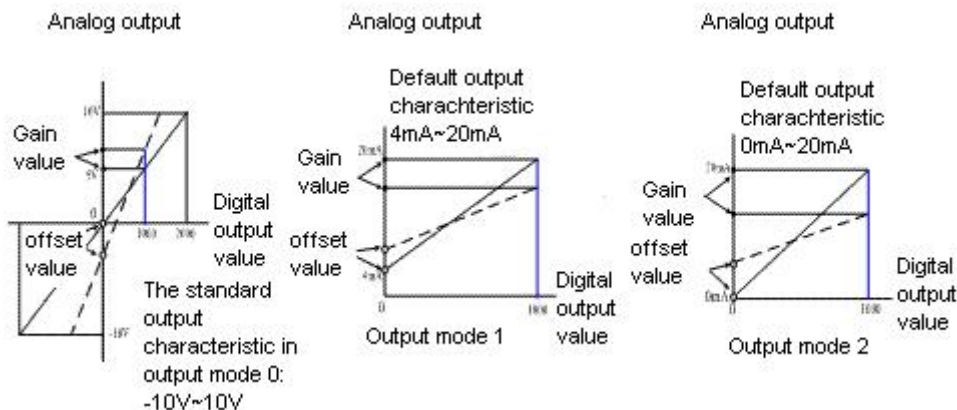
When the main module status changes from RUN to STOP, the last mode of RUN status is retained (X=0) or the offset (X=1) output.

For example, when BFM#32 is H00 and BFM#37 is H01, the offsets of the 2 channels are 0.1V. When the status changes from RUN to STOP, all values of BFM#33 to #34 are 1500 (7.5V). When the status changes to

STOP, the output voltage of channel 5 is 0.1V, and the output voltages of channel 6 is 7.5V.

BFM #40, BFM #42 to #45, BFM #53 Definition and setting method of offset and gain values:

H2U-6AM(R) has three working modes. The curve is as follows:



The gain is the analog output of digital 1000; the offset is the analog output of digital 0. The offset and gain values can be set separately or together. The units are mV (mode 0) and  $\mu$ A (modes 1 and 2).

BFM #40 are the offset/gain configuration commands. In the hex format, each HEX bit (4 bits) controls prohibit and permit. Note that it is controlled by each bit in the AD module. The offset and gain configuration commands for the DA and AD modules are different. BFM #10 to #17 are the offset and gain values. BFM #21 is the curve setting command.

Before setting the gain/offset, set BFM#53 to 1, and modify the values of BFM#42-#45. Write the permission word to BFM#40. After the modification, set BFM#53 to 2 again, to avoid change.

For example, when BFM#40 is in mode H00, set the offset and gain of channel 5 to 0.2 V and 5.5 V, respectively. The operation is as follows:

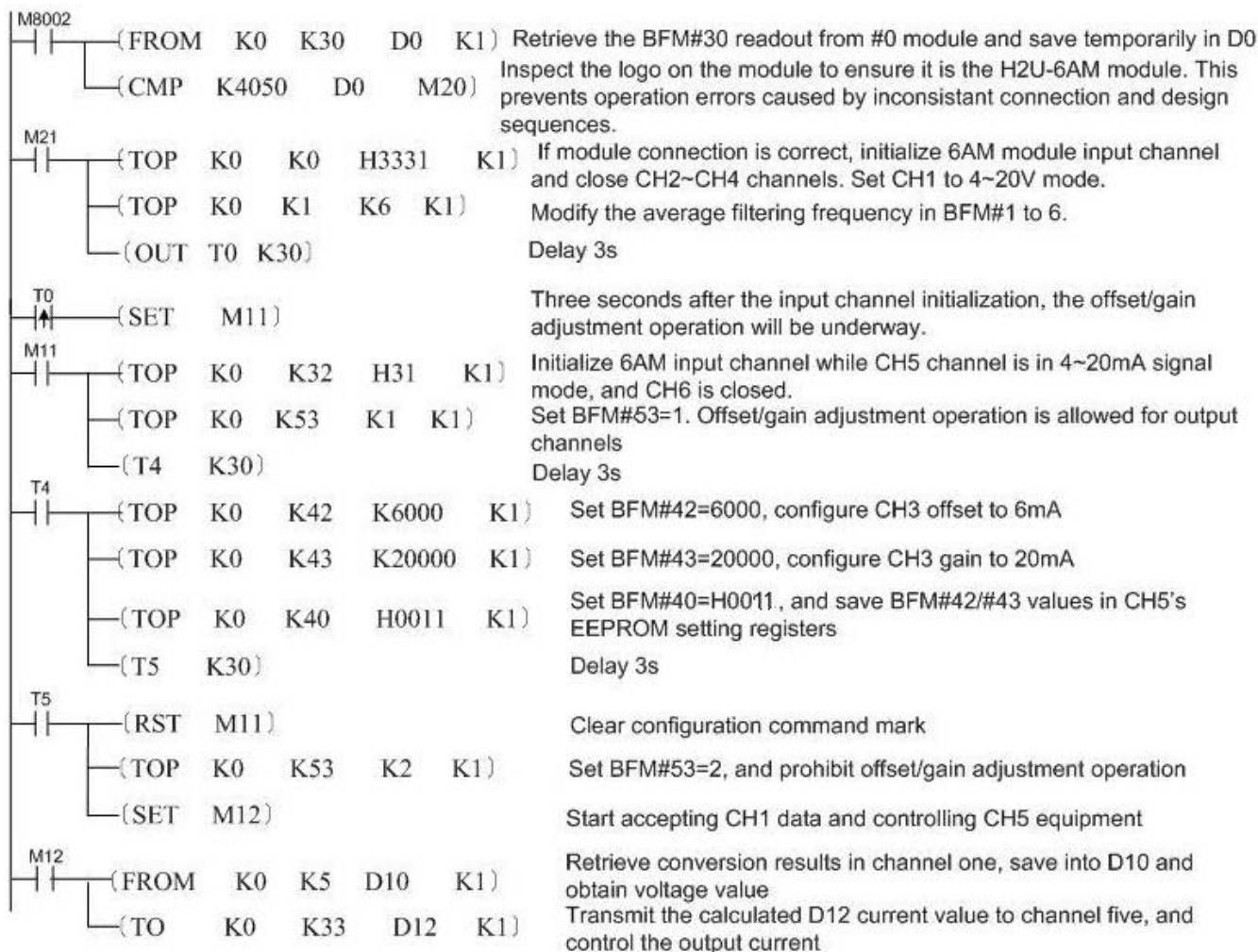
Change BFM#53 to 1. After 300ms, send K200 and K5500 to BFM#42 and #43, respectively. Allow offset and gain in BFM#40, that is, change BFM#40 to H0011. The modification is complete. Then change BFM#53 to 2, to prevent from the data being modified again.

BFM #52 factory restore setting of the output channel.

To restore the default setting, set BFM#52 to 1.

Programming example 1:

A remote extension module H2u-6AM is inserted behind the PLC main module, numbered #0. The CH1 port needs to collect the 4 mA-20 mA voltage signal. It is required to change the filtering times to 6 and store the collected data to D10. The CH5 outputs 6 mA-20 mA current signal through D12. The user program is as follows:



#### Precautions:

Online removal or installation is forbidden. The extension module can be connected or removed only when the main module and application systems are powered off; otherwise, there may be a harm to human body, or the components may be damaged.

### 6.5.3 H2U-6CM 4-channel analog voltage input/2-channel analog voltage and current output hybrid expansion module

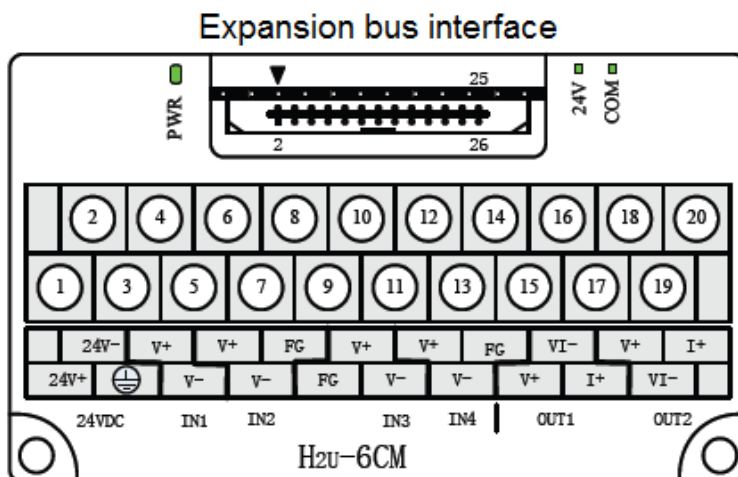
#### 1) Introduction

The H2U local expansion module can work with the H2U series main module to realize the signal detection of 4 analog input channels and the output of 2 analog signal channels. Each input channel converts the 0V~10V signal into a 12bit digital quantity for the PLC main module to read; each output channel has a voltage signal and current signal output port, and the signal amplitude can be -10~10V or 0~20mA respectively. The main module accesses the BFM unit of the register in the expansion module through the FROM/TO instruction to control the analog output signal.

2) Electrical specifications:

Item		Description
Power supply	Analog	24V DC -15%/+20%, maximum allowable ripple voltage 5%, current consumption 100mA (taken from 24V/COM of the main module, or other 24VDC power supply)
	Digital	5V DC 50mA (through the module extension cable, taken from the main module power supply internal power supply)
Item		Index for AD
Occupied I/O points		None
Conversion speed		15ms/channel (normal speed), 6ms/channel (fast), 1ms/channel (fastest)
Analog input range	Voltage input	0~10V DC (input impedance is 200KΩ)
Digital output		Default setting: 0 ~ +2000
Resolution	Voltage input	20mV
Precision		±1% full range
Isolation		The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.
Item		Index for DA
Occupied I/O points		Does not occupy the main module I/O points
Conversion speed		2 channels 2.1ms
Analog input range	Voltage input	-10~10V DC, (input impedance is 2KΩ ~ 1MΩ)
	Current input	0~20mA (The external load impedance is 500Ω or less)
Digital output		Default setting: -2000 ~ 2000, allowable range: -10000 ~ 10000
Resolution	Voltage input	5mV (10V\2000)
	Current input	20μA (20mA\1000)
Precision		±1% (for the full range of 10V) ±1% (for full range of 20mA)
Isolation		The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog channels.

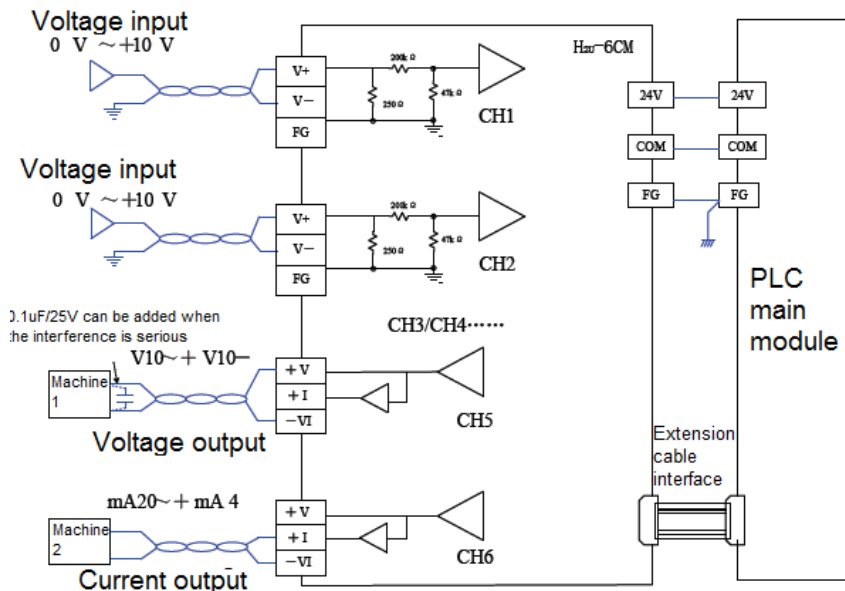
3) Port description:





4) External wiring:

The analog signal is connected to the input/output port of the expansion module through a twisted pair cable. When wiring, do not close to the AC power line or interference signal line; if the interference of the analog signal is serious, a shielded and connect a 0.1uF/25V high-frequency capacitor at the voltage input port of the controlled device; the shell of the signal source/controlled component and its shielding wire is connected to the signal ground terminal FG of H2U-6CM, and grounded together



5) Access the BFM area of 6CM (R):

The PLC main module accesses the BFM units using the FROM/TO instruction.

The extension module has the built-in EEPROM memory unit, for saving the BFM values, for example, the signal type, offset value, and gain value. The unit saving action is implemented by the setting of the corresponding BFM units.

The width of each register in the BFM area is 16 bits (1 word). The BFM area definition of the 6CM(R) module is as follows:

BFM	R/W attribute	Input channel content	
#0 (E)	WR	Signal mode selection of the channel. Each hex bit indicates an input channel. The highest bit is ch4 and the lowest bit is ch1: (default = H0000) 0 = 0V-10V; corresponding digital output 0-2000 3 = local channel is closed; 4 = 0V-10V; corresponding digital output: 0-10000	
#1	WR	Channel 1	Average filter constant, namely, the number of sampled values for average value calculation. Range: 1-4096; default 8. The value can be set to 1 if high-speed collection is needed. When BFM#15 is changed, the value is restored to the default.
#2	WR	Channel 2	
#3	WR	Channel 3	
#4	WR	Channel 4	
#5	R	Channel 1	Mean filtering value of the sampled value of the input channel
#6	R	Channel 2	



BFM	R/W attribute	Input channel content								
#7	R	Channel 3	Currently sampled data of the input channel, namely, burst value without filtering							
#8	R	Channel 4								
#9	R	Channel 1								
#10	R	Channel 2								
#11	R	Channel 2								
#12	R	Channel 2								
#13-14	-	Reserved								
#15	WR	ADC speed selection	0 = normal speed, 15ms/channel (default value); 1 = fast conversion, 6ms/channel; 1000-30000 = high-speed sampling, corresponding to 1ms-30ms/channel							
#16-19	-	Reserved								
#20 (E)	WR	1 = reset; Restore the parameters of all channels (BFM#0~BFM#32) to the default (factory setting). Default = 0								
#21 (E)	WR	2 = Adjusting offset/gain values is not allowed (default); 1 = Adjusting offset/gain values is allowed (default)								
#22 (E)	WR	Operations for the four channels corresponding to the lower 8 bits	G4	04	G3	03	G2	02	G1	01
		Offset/gain value adjusting is enabled. When the value is non-0, the module writes the BFM 23/24 value into the control register of the internal channel.								
#23 (E)	WR	Offset value. When the digital output is 0, the initial value of the analog input value (mode 0, 1, or 2) is 400.								
#24 (E)	WR	Gain value. When the digital output is +1000, the initial value of the analog input value (mode 0, 1, or 2) is 20000.								
#25~26	-	Reserved								
#27	R	6CM module software version								
#28	-	Reserved								
#29	R	Input channel error status								
#30	R	ID of the extension module. The ID of H2U-6CM is K4052.								
#31	-	Reserved, not accessible								
#32 (E)	WR	Output mode selection. Each hex bit indicates one output channel. The highest hex bit in the lowest 8 bits of the 4AM(R) module is ch4 and the lowest hex bit is ch3; (default = H00)	<div style="border: 1px dashed black; padding: 10px; display: inline-block;"> <p><b>Example:</b></p> <p>□ □ □ □</p> <p>ch4 3 2 1</p> </div>							
		0 = -10V to 10V; corresponding digital output: -2000-2000 1 = 4mA-20mA; corresponding digital output: 0-1000 2 = 0mA-20mA; corresponding digital output: 0-1000 4 = -10V-10V; corresponding digital output: -10000-10000 5 = 4mA-20mA; corresponding digital output: 0-10000 6 = 0mA-20mA; corresponding digital output: 0-10000								
#33	WR	Channel 5	Output value of the channel, initial value 0							
#34	WR	Channel 6								
#35-36	-	Reserved								
#37 (E)	WR	Value retain mode when PLC is stopped. Each hex bit indicates an output channel. The highest hex bit of the lower 8 bits of 6CM(R) is ch6, and the lowest hex bit is ch5. When								

BFM	R/W attribute	Input channel content							
		x=0, retain the output before stop; x=1, reset output to the offset value							
#38-39	-	Reserved							
#40 (E)	WR	G2	O2	G1	O1				
		CH6/CH5 offset/gain configuration command. Configured by hex bits, initial value = H0000; 0 = Modification not allowed; 1 = EEPROM allowed to be modified.							
#41	-	Reserved							
#42	WR	Offset data ch5			Unit: mV or uA, initial offset: 0, initial gain: +5000, corresponding mode: 0 (mode 0, 1, 2)				
#43	WR	Gain data ch5							
#44	WR	Offset data ch6							
#45	WR	Gain data ch6							
#46-51	-	Reserved							
#52 (E)	WR	Initial value = 0; when 1 is written, all BFM units (BFM#32~BFM#64) restore the initial values to default values.							
#53 (E)	WR	1 = Adjusting output (initial value) is allowed; 2 = Adjusting output is not allowed.							
#54-60	-	Reserved							
#61	R	Output channel error status							

The meaning of the status information field BFM #29 is as follows:

BFM#29	ON	OFF
b0:	An error exists. When any value in b0-b3 is not 0, A/D conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	24V DC power has an error.	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal.
b10:	Digital output exceeds the range of -2048-2047.	Digital output is normal.
b11:	The sampling filter constant exceeds the range of 1-4096.	The sampling filter constant is normal.
b12:	The value of BFM#21 cannot be set to K2.	The value of BFM#21 can be set to K2.
Other bits such as bits 4-7 and bits 13-15 of BFM#29 are not defined.		

The meaning of the status information field BFM #61 is as follows:

BFM#61	ON	OFF
b0:	An error exists. When any value in b1-b3 is not 0, D/A conversion is stopped.	No error
b1:	The EEPROM offset/gain value setting in the module is incorrect.	The offset/gain values are correct.
b2:	24V DC power has an error.	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal.
b10:	Written value is beyond the acceptable range. (-2350~2350)	Digital output is normal.
b12:	The value of BFM#21 cannot be set to K1.	BFM#21=K1
Other bits such as bits 4-7, bit 11, and bits 13-15 of BFM#61 are not defined.		

In the table, the BFM unit with "(E)" is the project stored in EEPROM. It will not be deleted upon power off.

## Register Modification:

The modification of the BFM units carrying (E), such as BFM#0 and #32 causes the write operation on EEPROM. In the write operation, each Word requires about 300 ms. When the write operation is performed on multiple BFM units carrying (E), the PLC programming requires a delay after each BFM unit is written. The next write operation can be performed only when the previous instruction is complete.

Note: The external 24 V power supply for the local extension module is lost. The system flag M8067 of the PLC main module is reset. The error code is D8067=K6708. This symptom can be detected if you often check the flag in editing.

## 6) Explanation of input channel of BFM area:

## BFM#0 input channel selection

The channels are initialized. By default, all the four channels are 0~10V, which is controlled by the hex string HXXXX of BFM #0. The lowest bit control channel is 1, and the highest bit control channel is 0. The character control is as follows:

X=0 pre-defined range 0V~10V (corresponding digital value 0~2000)

X=3, the channel is closed.

X=4 pre-defined range 0V~10V (corresponding digital value 0~10000)

For example:

If BFM#0 is H0034, channel 1 is 0V~10V, channel 2 is closed, and channel 3 and channel 4 are 0V~10V. The unused channel can be closed or not. The closed channel does not occupy conversion time (BFM#15). In the example, channel 2 is closed, so the total conversion time is the sum of conversion time of the other three channels, namely, 3×BFM#15.

The average input sampling value of BFM #1~#4. The sampling value of each channel (BFM#9~#10) plus the number of samples (BFM #1~#4) divided the number of samples (BFM #1~#4) are stored in BFM #5~#6).

BFM #5~#8 stores the average sampling values.

BFM #9~#12 stores the instant sampling values.

## BFM #15 Input ADC rate

Time of one conversion of each channel. The time of each data update is BFM#15 x number of opened channels.

For example:

BFM #0 is H3311, BFM #1 is K7, BFM #2 is K6, and BFM #15 is K10.

The time of one data update of BFM #9 and BFM #10 is  $\text{BFM}\#0 \times \text{BFM}\#15 = 2 \times 10 = 20\text{ms}$ , the time of one data update of BFM #5 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#1 = 2 \times 10 \times 7 = 140\text{ms}$ , and the time of one data update of BFM #6 is  $\text{BFM}\#0 \times \text{BFM}\#15 \times \text{BFM}\#2 = 2 \times 10 \times 6 = 120\text{ms}$ .

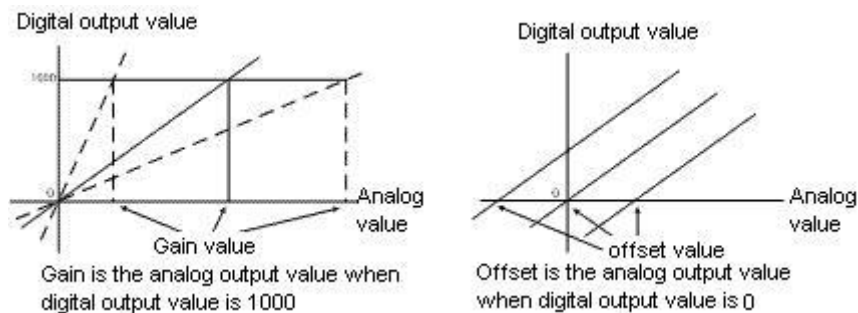
The FROM/TO instruction is time consuming, so the BFM#5 data collected by this module can be read using the LDP M8012 FROM K0 K5 D10 K1 instruction, which has the same effect as using LD M8000 FROM

K0 K5 D10 K1. However, the data is read in each scanning period of the M8000 drive. The program scanning period is prolonged.

BFM#20 restores the input channel to the factory setting. To restore the default setting, set BFM#20 to 1.

BFM #21~#24 Definition and setting method of offset and gain values:

H2U-6CM(R) has two working modes. The curve is as follows:



The offset and gain values can be set separately or together. The normal gain value ranges from 1V to 15V. The normal offset value ranges from 0V to 5V.

Before setting the offset/gain, set BFM#21 to 1, modify BFM#23/24, and then allow offset/gain BFM#22 for each channel. After the modification, set BFM#21 to 2 again, to avoid change.

Note that the channels requiring offset/gain must be the same. It is not allowed that the offset of one channel is 1000 but the offset of another channel is 1200.

For example, when BFM#0 is in mode 0, set the offset and gain of channel 1 and channel 2 to 0.5V and 6V, respectively. The operation is as follows:

Change BFM#21 to 1. After 300ms, send K500 and K6000 to BFM#23 and BFM#24, respectively.

After 300ms, allow gain BFM#22. In this example, BFM#22 should be binary form 1111. That is, change BFM#22 to H000F. The modification is complete.

Then change BFM#21 to 2, to prevent from the data being modified again.

#### 7) Explanation of output channel of BFM area:

BFM#32 Initialize output channel. By default, two channels are -10V~10V, which is controlled by hex form HXX of BFM 32#. The lowest bit controls channel 3, and the second bit controls ch4. The character control is as follows:

X=0 pre-defined range -10V~10V (corresponding digital value -2000~2000)

X=1 pre-defined range 4mA~20 mA (corresponding digital value 0~1000)

X= 2 pre-defined range 0mA~20 mA (corresponding digital value -10000~10000)

X=4 pre-defined range -10V~10V (corresponding digital value -10000~10000)

X=5 pre-defined range 4mA~20 mA (corresponding digital value 0~10000)

X=6 pre-defined range 0mA~20 mA (corresponding digital value 0~10000)

For example, if BFM#32 is H10, channel 5 is -10V-10V and channel 6 is 4mA-20mA.

BFM #33~#34 Output value of the output channel. To control analog output, use the TO instruction to write data to BFM #33~#34. The initial values are 0.

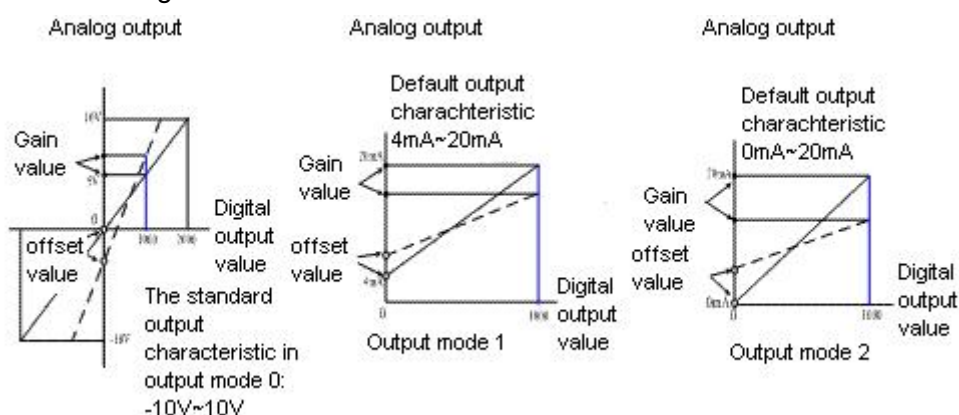
BFM #37 Data retain mode of the output channel. When the main module status changes from RUN to STOP, the last mode of RUN status is retained (X=0) or the offset (X=1) output.

For example:

When BFM#32 is H00 and BFM#37 is H01, the offsets of the 2 channels are 0.1V. When the status changes from RUN to STOP, all values of BFM#33~#34 are 1500 (7.5V). When the status changes to STOP, the output voltage of channel 5 is 0.1V, and the output voltages of channel 6 is 7.5V.

BFM #40, BFM #42~#45, BFM #53 Definition and setting method of offset and gain values:

H2U-4AM(R) has three working modes. The curve is as follows:



The gain is the analog output of digital 1000; the offset is the analog output of digital 0. The offset and gain can be set separately or together. The units are mV (mode 0) and uA (modes 1 and 2). BFM #40 are the offset/gain configuration commands. In the hex format, each HEX bit (4 binary bits) controls prohibit and permit. Note that it is controlled by each bit in the AD module. The offset and gain configuration commands for the DA and AD modules are different.

FM #10~#13 are the offset and gain values. BFM #53 is the curve setting commands. Before setting the gain/offset, set BFM#53 to 1, and modify the values of BFM#42~#45. Write the permission word to BFM#40. After the modification, set BFM#53 to 2 again, to avoid change.

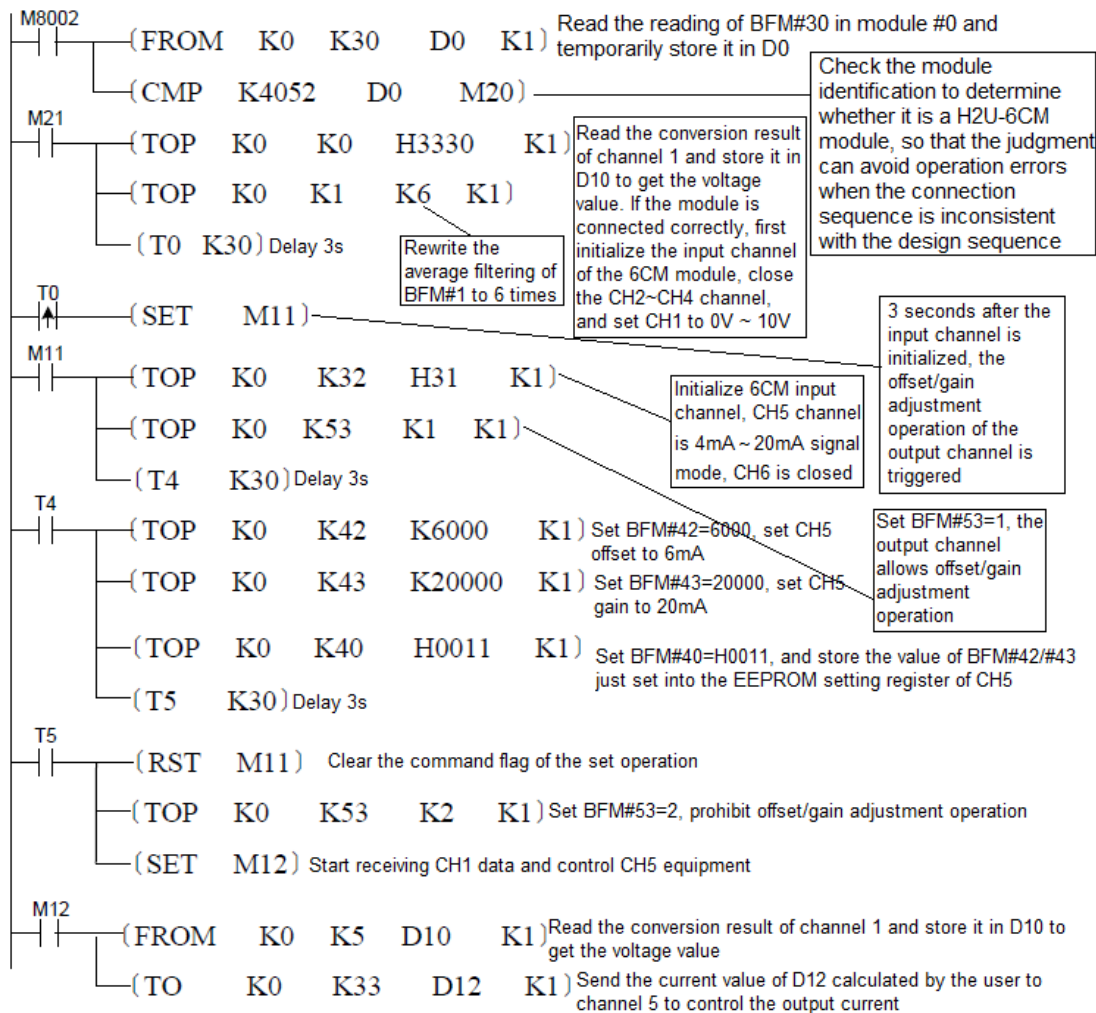
For example:

When BFM#40 is in mode H00, set the offset and gain of channel 3 to 0.2V and 5.5V, respectively. The operation is as follows:

Change BFM#53 to 1. After 300ms, send K200 and K5500 to BFM#42 and #43, respectively. Allow offset and gain in BFM#40, that is, change BFM#40 to H0011. The modification is complete. Then change BFM#53 to 2, to prevent from the data being modified again. BFM #52 restores the output channel to the factory setting. To restore the default setting, set BFM#52 to 1.

Programming example 1:

An extension module H2U-6CM is inserted behind the PLC main module, numbered #0. The CH1 port needs to collect the 0V-10V voltage signal. It is required to change the filtering times to 6, store the collected data to D10.CH5, and the 6mA~20mA current signal is output through the D12 port.



Precautions:

Online removal or installation is forbidden. The extension module can be connected or removed only when the main module and application systems are powered off; otherwise, there may be a harm to human body or the components may be damaged.

## 6.6 Temperature control module

1) H2U-4PT(R) 4-channel thermistor input temperature control module H2U-4PT(R) expansion module can work with the H2U series main module to realize 4-channel thermal resistance temperature control, and convert the temperature signal into a 12bit precision digital signal, For the PLC main module to read. The main module can access the module through CANlink configuration.

The H2U-4PT(R) expansion module mainly adds temperature control functions to the module based on the original 4PT(R) module. The user can set the parameters to the 4PT(R) temperature control module through communication, and directly control the temperature in the expansion module through the collected temperature and set parameters during operation, and finally save the output in the specified address. Since the expansion module has no output points, the PLC needs to perform output control on the host through communication.

The H2U-4PT(R) expansion module has the basic functions of a thermostat. It mainly has ON/OFF control mode, PID control mode, single-side control mode, heating and cooling control mode, digital output control, relay output control, PID auto-tuning function and temperature curve function. At the same time, the basic setting parameters are restricted in real time, and the running status and alarms are displayed in real time.

2) H2U-4TC(R) 4-channel thermocouple input temperature control module

The H2U-4TC(R) expansion module can work with the H2U series main module to realize 4 thermocouple temperature control and convert the temperature signal into a 12-bit precision digital signal for the PLC main module to read. The main module can access the module through CANlink configuration.

H2U-4TC(R) expansion module mainly adds temperature control function to the module based on the original 4TC(R) module. The user can set the parameters to the 4TC(R) temperature control module through communication, and directly control the temperature in the expansion module through the collected temperature and setting parameters during operation, and finally save the output in the specified address. Since the expansion module has no output points, the PLC needs to perform output control on the host through communication.

The H2U-4TC(R) expansion module has the basic functions of a thermostat. It mainly has ON/OFF control mode, PID control mode, single-side control mode, heating and cooling control mode, digital output control, relay output control, PID auto-tuning function and temperature curve function. At the same time, the basic setting parameters are restricted in real time, and the running status and alarms are displayed in real time.

3) Electrical specifications

Power supply	Analog	24V DC -15%/+20%, maximum allowable ripple voltage 5%; current consumption 80mA (taken from the 24V/COM of the main module, or other 24VDC power supply)
	Digital	5V DC 50mA (Through the module extension cable, take the internal power supply of the independent module power supply)
	Remark	The remote expansion module does not require a digital circuit power supply.

## 4PT (R) description:

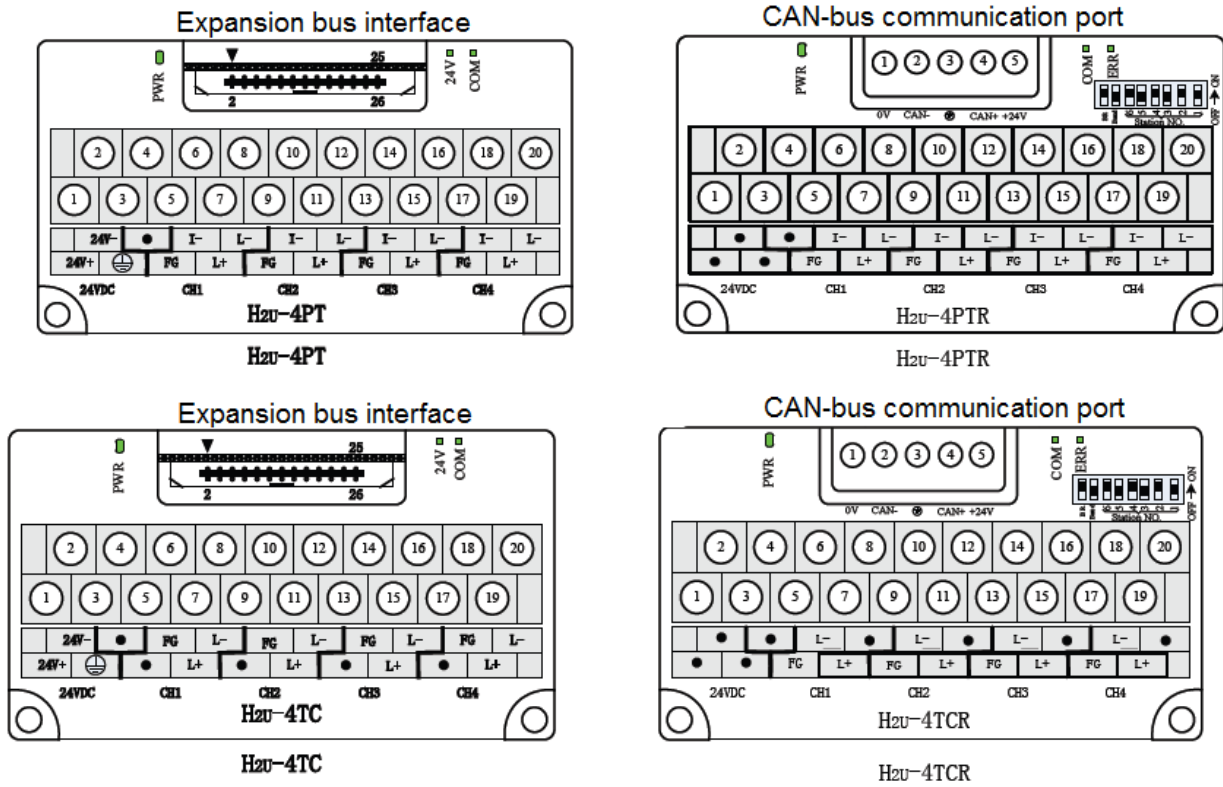
Item	4PT (R)	
Temperature detection sensor	Pt100	Cu50
Temperature detection range	Celsius: -100°C~600°C Fahrenheit: -148°F~1112°F	Celsius: -50°C~150°C Fahrenheit: -58°F~302°F
Number of input channels	4 channels	-
Conversion speed	15ms/channel	-
Digital output	Celsius: -1000°C~6000°C Fahrenheit: -1480°F~11120°F	Celsius: -500°C~1500°C Fahrenheit: -580°F~3020°F
Resolution	0.1°C	-
Accuracy	±1% full range	-
I/O points	8	
Isolation	The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog input signal channels.	

## 4TC (R) description:

Item	Celsius	Fahrenheit
Temperature detection sensor	B/E/N/J/K/R/S/T type thermocouple	
Temperature detection range	Type B: 200~1800°C Type E: -100~800°C Type N: -100~1200°C Type J: -100~850°C Type K: -200~1300°C R type: -10~1700°C S type: -10~1700°C Type T: -200~400°C	Type B: 3920~32720 Type E: -1480~14720 Type N: -1480~21920 Type J: -1480~15620 Type K: -3280~23720 R type: 140~30920 S type: 140~30920 Type T: -3280~7520
Number of input channels	4 channels	-
Conversion speed	800ms/4channel	-
Digital output	Type B: 2000~18000 Type E: -1000~8000 Type N: -1000~12000 Type J: -1000~8500 Type K: -2000~13000 R type: -100~17000 S type: -100~17000 T type: -2000~4000	Type B: 3920~32720 Type E: -1480~14720 Type N: -1480~21920 Type J: -1480~15620 Type K: -3280~23720 R type: 140~30920 S type: 140~30920 Type T: -3280~7520
Resolution	0.1°C	0.1F
Accuracy	±0.5% full range +1°C	
I/O points	8	
Isolation	The analog circuit and the digital circuit are isolated by a photocoupler; the analog circuit and the external power supply are isolated by DC/DC; there is no isolation between the analog input signal channels.	

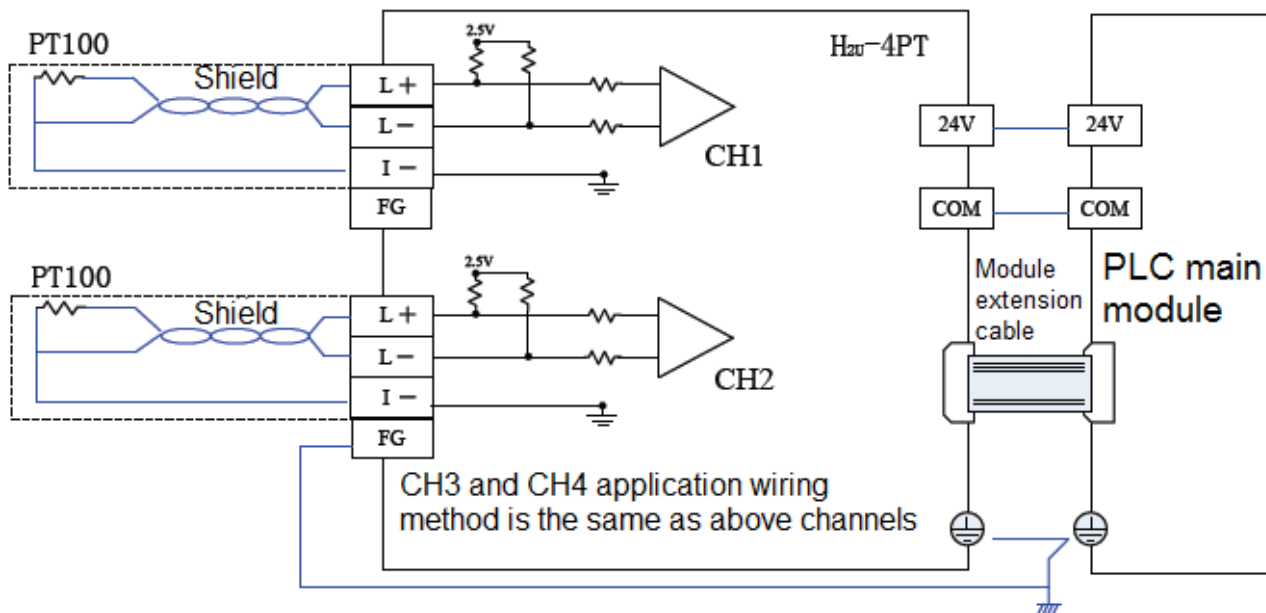


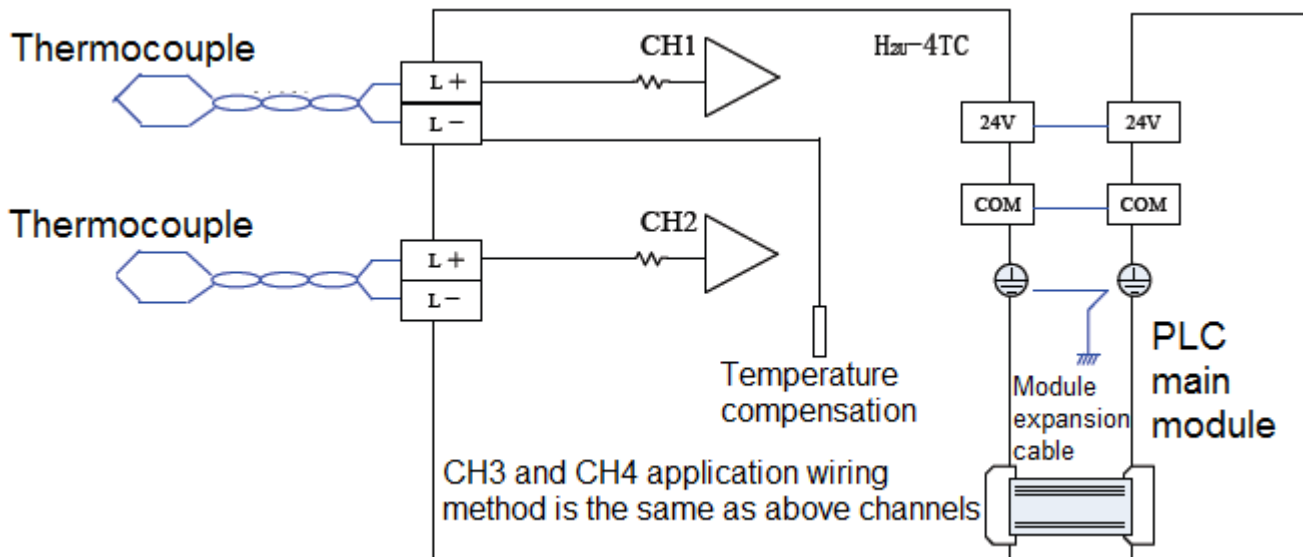
4) Port description:



5) Wiring of input signal and extension cable:

The analog input signal is connected to the input port of the expansion module through a twisted pair cable. When wiring, do not close to the AC power line or the line that interferes with the signal; the shell of the signal source and its shielded wire is connected to the signal ground terminal FG of H2U-4PT(R)/H2U-4TC(R), and are grounded together.





6) Access the BFM area of the 4PT (R) module

The PLC main module reads the AD conversion result by reading the BFM area of the 4PT(R) module and sets the module status by modifying the specific BFM area. The PLC main module accesses the BFM units using the FROM/TO instruction.

The register width in the BFM area is 16 bits (that is, 1 Word). The BFM area definition of the 4PT(R) module is as follows:

BFM	R/W attribute	Content	
#0	WR	4PT (R): Thermistor model selection, each HEX bit represents 1 input channel, the highest bit is ch4, and the lowest bit is ch1: (default = H0000) 0: PT100 5: Cu50	<div style="border: 1px dashed black; padding: 5px; display: inline-block;">                     Example:  <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>                      ch4 3 2 1                 </div>
#1	WR	Channel 1	Average filter constant, namely, the number of sampled values for average value calculation. Range: 1-4096; default 8. The value can be set to 1 if high-speed collection is needed.
#2	WR	Channel 2	
#3	WR	Channel 3	
#4	WR	Channel 4	
#5	WR	Channel 1	The average temperature of CH1 to CH4 is under the unit of 0.1°C.
#6	WR	Channel 2	
#7	WR	Channel 3	
#8	WR	Channel 4	
#9	WR	Channel 1	The current temperature of CH1 to CH4 is under the unit of 0.1°C.
#10	WR	Channel 2	
#11	WR	Channel 3	
#12	WR	Channel 4	
#13	WR	Channel 1	The average temperature (F) of CH1 to CH4 is under the unit of 0.1F.
#14	WR	Channel 2	
#15	WR	Channel 3	
#16	WR	Channel 4	
#17	WR	Channel 1	The current temperature (F) of CH1 to CH4 is under the unit of 0.F.

BFM	R/W attribute	Content	
#18	WR	Channel 2	
#19	WR	Channel 3	
#20	WR	Channel 4	
#21~26	-	Reserved	
#27	R	4PTModule software version	
#28	R/W	The incorrect digital range is locked (disconnection by thermistor can be detected)	
#29	R	Error status	
#30	R	Module ID. The ID of the H2U- 4PTmodule is K2040.	
#31	-	Reserved, not accessible	

The meaning of the status information field BFM #28 is as follows:

b15-b8	b7	b6	b5	b4	b3	b2	b1	b0
Unused	High	Low	High	Low	High	Low	High	Low
	CH4		CH3		CH2		CH1	

Low: When the measured temperature is lower than the minimum value, lock is ON.

High: When the measured temperature is higher than the highest value, lock is ON.

If an error occurs, the temperature falls into the normal range. The error is still locked in BFM#28.

Use the TO instruction to write K0 to BFM28 or switch off the power, to clear the error.

The meaning of the status information field BFM #29 is as follows:

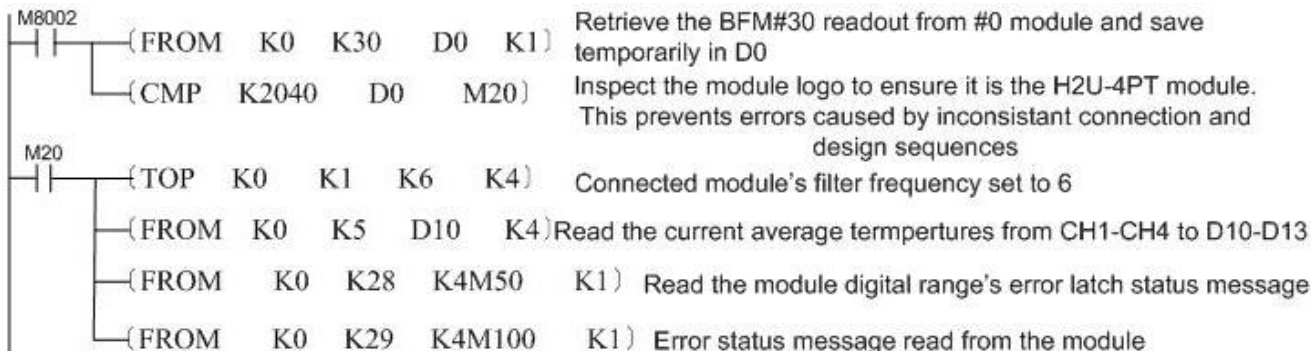
BFM# 29 bit	ON	OFF
b0:	An error exists. When any value in b0-b3 is not 0, A/D conversion is stopped.	No error
b1:	Reserved	Reserved
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal
b10:	Digital output is beyond the specified range.	Digital output is normal.
b11:	The sampling filter constant exceeds the range of 1-4096.	The sampling filter constant is normal.
b12:	Reserved	Reserved
Bits 4-7 and bit13-15 of BFM#29 are not defined.		

Note: if the external 24 V power supply for the local extension module is lost then the system flag M8067 of the PLC main module is reset. The error code is D8067 = K6708. This symptom can be detected if you often check the flag in editing.

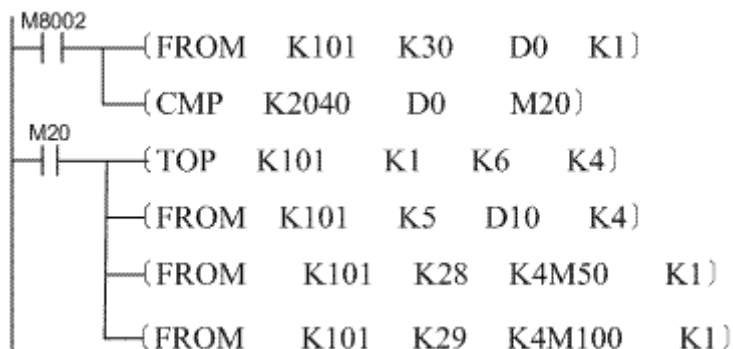
Programming example:

An H2U-4PT module is inserted behind the PLC main module and is numbered #0. The CH1-CH4 ports need to collect the voltage signal of PT100. It is required to change the filtering times to 6. The data collected

from the two channels is stored in D10, D11, D12, and D13. The user program is as follows:



In this example, if the module is changed to H2U-4PTR, and the CAN station address is 1, then the program in this example is as follows:



7) Access the BFM area of 4TC (R)

The PLC main module reads the AD conversion result by reading the BFM area of the 4TC (R) module, and sets the module status by modifying the specific BFM area. The PLC main module accesses the BFM units using the FROM/TO instruction.

The width of each register in the BFM area is 16 bits, that is, 1 Word. The BFM area definitions according to the 4TC (R) module is as follows:

BFM	R/W attribute	Content	
#0	WR	4TC (R): Thermocouple model selection, each HEX bit represents 1 input channel, the highest bit is ch4, and the lowest bit is ch1: (default = H0000) 0: K type 3: B type 6: S type 1: J type 4: N type 7: T type 2: E type 5: R type	<div style="border: 1px dashed black; padding: 5px; display: inline-block;">                     Example:                      ch <input type="checkbox"/>4 <input type="checkbox"/>3 <input type="checkbox"/>2 <input type="checkbox"/>1                 </div>
#1	WR	Channel 1	Average filter constant, namely, the number of sampled values for average value calculation. Range: 1-256; default 8. The value can be set to 1 if high-speed collection is needed.
#2	WR	Channel 2	
#3	WR	Channel 3	
#4	WR	Channel 4	
#5	WR	Channel 1	The average temperature of CH1-CH4 with the unit 0.1°C.
#6	WR	Channel 2	

BFM	R/W attribute	Content	
#7	WR	Channel 3	
#8	WR	Channel 4	
#9	WR	Channel 1	
#10	WR	Channel 2	
#11	WR	Channel 3	The current temperature of CH1-CH4 with the unit 0.1°C
#12	WR	Channel 4	
#13	WR	Channel 1	
#14	WR	Channel 2	
#15	WR	Channel 3	The average temperature of CH1-CH4 with the unit 0.1F.
#16	WR	Channel 4	
#17	WR	Channel 1	
#18	WR	Channel 2	
#19	WR	Channel 3	The current temperature of CH1-CH4 with the unit 0.1F.
#20	WR	Channel 4	
#21~26	-	Reserved	
#27	R	4TC module software version	
#28	R/W	The incorrect digital range is locked (disconnection by thermocouple can be detected)	
#29	R	Error status	
#30	R	ID of the extension module. The ID of H2U-4AD is K2030.	
#31	-	Reserved, not accessible	

The meaning of the status information field BFM #29 is as follows:

b15 to b8	b7	b6	b5	b4	b3	b2	b1	b0
Unused	High	Low	High	Low	High	Low	High	Low
	CH4		CH3		CH2		CH1	

Low: When the measured temperature is lower than the minimum value, lock is ON.

High: When the measured temperature is higher than the highest value, lock is ON.

If an error occurs, the temperature falls into the normal range. The error is still locked in BFM#28.

Use the TO instruction to write K0 to BFM28 or switch off the power, to clear the error.

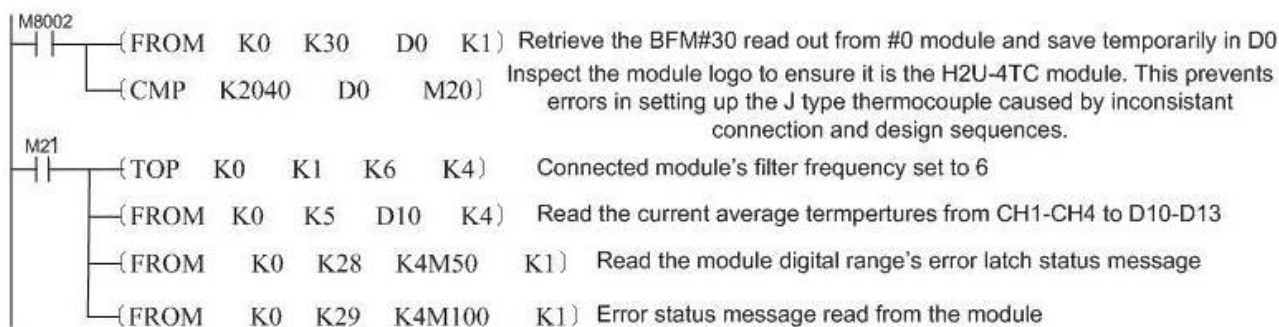
Note: The external 24 V power supply for the local extension module is lost. The system flag M8067 of the PLC main module is reset. The error code is D8067 = K6708. This symptom can be detected if you often check the flag in editing.

The meaning of the status information field BFM #29 is as follows:

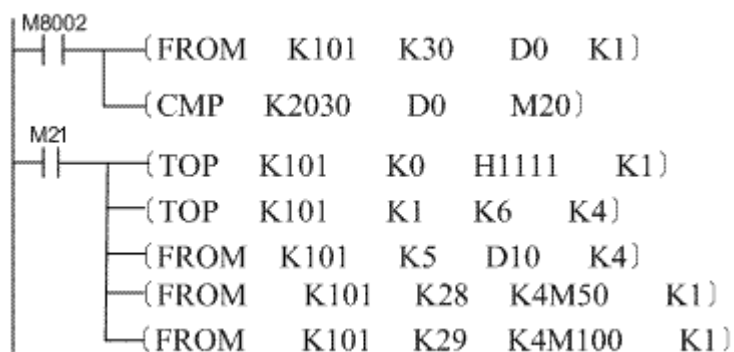
BFM# 29 bit	ON	OFF
b0:	An error exists. When any value in b0-b3 is not 0, A/D conversion is stopped.	No error
b1:	Reserved	Reserved
b2:	(Impossible)	Power is normal.
b3:	Module hardware is faulty.	Hardware is normal
b10:	Digital output is beyond the specified range.	Digital output is normal.
b11:	The sampling filter constant exceeds the range of 1-256.	The sampling filter constant is normal.
b12:	Reserved	Reserved
Bits 4-7 and bits 13-15 of BFM#29 are not defined.		

### Programming Example

The extended module H2U-4TC is inserted behind the PLC main module, and is numbered #0. The CH1-CH4 ports need to collect the temperature of J thermocouple. It is required to change the filtering times to 6. The data collected from the two channels is stored in D10, D11, D12, and D13. The user program is as follows:



In this example, if the module is changed to H2U-4TCR, and the CAN station address is 1. The program in this example is as follows:



- 8) 4PT(R) temperature control part (only the temperature control function with XP added after the model. For example: H2U-4PT-XP)

The original parameters in the 4PT(R) temperature control module retain their original meanings, and some new parameters are added. In BFM#40~#69, perform temperature control start control and save the setting of current cycle output value, as shown in the table temperature control start and output data parameter definition. BFM#100~#199 are the first channel temperature control setting parameters, BFM#200~#299 are the second channel temperature control setting parameters, BFM#300~#399 are the 3rd channel temperature control setting parameters, BFM#400~# 499 is the 4th channel temperature control setting parameter. The position of each channel parameter is the same but the first address is different, so take the first channel as an example to introduce the parameters after BFM#100.

Table: Temperature control start and output data parameter definition

BFM	R/W attributes	Content	
#51	WR	Channel 1	Temperature control start flag 0: No temperature control 1: Start temperature control
#52	WR	Channel 2	
#53	WR	Channel 3	
#54	WR	Channel 4	
#55	WR	Channel 1	PID control pause flag 0: PID control runs normally 1: PID pause
#56	WR	Channel 2	
#57	WR	Channel 3	
#58	WR	Channel 4	
#71	R	Channel 1	Heating control digital output value
#72	R	Channel 2	
#73	R	Channel 3	
#74	R	Channel 4	
#75	R	Channel 1	Heating control relay output, only output 1 and 0 If it is 1, the relay is on If it is 0, the relay is off
#76	R	Channel 2	
#77	R	Channel 3	
#78	R	Channel 4	
#81	R	Channel 1	Cooling control digital output
#82	R	Channel 2	
#83	R	Channel 3	
#84	R	Channel 4	
#85	R	Channel 1	Cooling control relay output, only output 1 and 0 If it is 1, the relay is on If it is 0, the relay is off
#86	R	Channel 2	
#87	R	Channel 3	
#88	R	Channel 4	
#91	WR	Channel 1	Set temperature Unit: 0.1°C
#92	WR	Channel 2	
#93	WR	Channel 3	
#94	WR	Channel 4	

Note: If the digital output does not set the range, the default is -32768~32767.



Table: Definition of basic parameters of channel 1

BFM	RW attributes	Content
#100	WR	Temperature control mode setting Bit0: 0: Relay output 1: Digital output Bit1: 0: PID control 1: ON/OFF control Bit2: 0: Single-sided control 1: Heating and cooling control Bit3: 0: Positive action (cooling) 1: Reverse action (heating) Bit4: 0: No action 1: PID auto-tuning Bit5: 0: Normal control 1: Temperature curve control
#101	WR	Parameter limit setting (set 1 for each bit to indicate limit, set 0 to indicate no limit) Bit0: Input value (PV) limit Bit1: Deviation band ( PV-SV ) limit Bit2: Output (MV) limit Bit3: Set value (SV) limit Bit4: Input change value limit Bit5: Output change value limit Bit6: Cold junction output limit
#110	WR	KP (PID proportional coefficient) 1~32767[%]
#111	WR	TI (PID integral coefficient) 0~32767(*100ms) 0: no points
#112	WR	TD (PID differential coefficient) 0~32767(*100ms) 0: no points
#113	WR	TS (PID sampling period) 1~32767(ms)
#114	WR	Filter coefficient 0~99[%] 0: no sampling
#116	WR	Auto-tuning output value
#117	R	Current target temperature
#118	WR	PID pause output value
#121	WR	Relay output control cycle (heating) unit: second
#122	WR	Relay output control cycle (cooling) unit: second
#124	WR	Heating hysteresis (ON/OFF control is effective)
#125	WR	Cooling hysteresis (ON/OFF and PID control are both effective)
#126	WR	Cooling coefficient (digital output cooling proportional coefficient)
#130	WR	Ramp time (temperature curve time unit) 0: unit minute 1: unit second
#131	WR	Ramp value (unit: 0.1°C)
#140	WR	Input (PV) upper limit
#141	WR	Input (PV) lower limit
#142	WR	Deviation band (  PV-SV  )
#143	WR	Output (MV) upper limit
#144	WR	Output (MV) lower limit
#145	WR	Cooling output upper limit
#146	WR	Cooling output lower limit
#148	WR	Set (SV) upper limit
#149	WR	Set (SV) lower limit



BFM	RW attributes	Content
#150	WR	Input change upper limit
#152	WR	Input change lower limit
#153	WR	Output change upper limit
#154	WR	Lower limit of output change
#156	R	Operating status 0: stop running 1: ON/OFF control 2: PID self-tuning 3: PID operation
#157	R	Alarm (stop running) specific alarm code see table BFM157 alarm meaning
#158	R	Alarm (continue to run) specific alarm code see table BFM158 alarm meaning
#160-199	R	PID running intermediate value

Variables not involved are reserved for future expansion. BFM#200~#499 are used for channels 2, 3 and 4 respectively, and the address offset is the same as that of channel 1.

Note: After starting the temperature control module, it is prohibited to rewrite the variable settings of BFM#100 and BFM#101. After the temperature control module stops, the variables can be rewritten.

It is recommended not to modify PID related parameters when the temperature control module is running.

Table: BFM#157 alarm meaning

Alarm code	BFM#157 alarm (stop running) meaning
1	Input limit parameter error
2	Deviation with parameter error
3	Output limit parameter error
4	Set value limit parameter error
5	Input change limit parameter error
6	Output change limit parameter error
7	Cold end limit parameter error
8	Temperature curve parameter error
9	Heating cycle parameter error
10	Incorrect cooling cycle parameters
30	Sampling time $TS < 0$
31	Input filter constant is abnormal
32	The input scale factor is abnormal
33	Abnormal cooling coefficient
34	Integration time is abnormal
35	Differential time is abnormal
50	PID self-tuning set value and deviation value are too small *1
51	The initial direction of PID auto-tuning is not consistent with the action *2
52	PID auto-tuning action is not obvious *3

\*1: When the difference between the measured value and the target value at the start of auto-tuning is 150 or

less, it will end. After confirming the measured value and target value, perform auto-tuning again.

\*2: From the start of auto-tuning, the predicted action direction of the measured value is inconsistent with the actual action direction during auto-tuning output. After correcting the relationship between the target value, auto-tuning output value, and measured value, perform auto-tuning again.

\*3: The measured value during auto-tuning cannot be operated correctly due to the up and down changes.

Please make the sampling time much longer than the output change period and increase the input filter constant. Please perform auto tuning again after setting changes.

Table: BFM#158 alarm meaning

Alarm code	BFM#158 alarm (continuous operation) meaning
bit 0	Input value (PV) overrun
bit 1	Set value (SV) overrun
bit 2	Output value (MV) exceeds limit
bit 3	Deviation band overrun
bit 4	Input change exceeds limit
bit 5	Output change exceeds limit
bit 6	Cold junction output exceeds limit

9) 4TC(R) temperature control part (only those with XP added after the model have temperature control function. For example: H2U-4TC-XP)

The original parameters in the 4TC temperature control module retain their original meanings, and some new parameters are added.

Table 7: Temperature control start and output data parameter definition

BFM	RW attributes	Content	
#32	WR	Channel 1	Set temperature Unit: 0.1°C
#33	WR	Channel 2	
#34	WR	Channel 3	
#35	WR	Channel 4	
#36	WR	Channel 1	KP (PID proportional coefficient) 1~32767[%]
#37	WR		TI (PID integral coefficient) 0~32767(*100ms) 0: no points
#38	WR		TD (PID differential coefficient) 0~32767(*100ms) 0: no points
#39	WR	Channel 2	KP (PID proportional coefficient) 1~32767[%]
#40	WR		TI (PID integral coefficient) 0~32767(*100ms) 0: no points
#41	WR		TD (PID differential coefficient) 0~32767(*100ms) 0: no points
#42	WR	Channel 3	KP (PID proportional coefficient) 1~32767[%]
#43	WR		TI (PID integral coefficient) 0~32767(*100ms) 0: no points
#44	WR		TD (PID differential coefficient) 0~32767(*100ms) 0: no points
#45	WR	Channel 4	KP (PID proportional coefficient) 1~32767[%]

BFM	R/W attributes	Content	
#46	WR		TI (PID integral coefficient) 0~32767(*100ms) 0: no points
#47	WR		TD (PID differential coefficient) 0~32767(*100ms) 0: no points
#51	WR	Channel 1	Temperature control start flag 0: No temperature control 1: Start temperature control
#52	WR	Channel 2	
#53	WR	Channel 3	
#54	WR	Channel 4	
#55	WR	Channel 1	Temperature control mode setting Bit0: 0: Relay output 1: Digital output Bit1: 0: PID control 1: ON/OFF control Bit2: 0: Single-sided control 1: Heating and cooling control Bit3: 0: Positive action (cooling) 1: Reverse action (heating) Bit4: 0: No action 1: PID auto-tuning (please refer to the "Using Method of Auto-tuning" below)
#56	WR	Channel 2	
#57	WR	Channel 3	
#58	WR	Channel 4	
#59	WR	Channel 1	PID control intensity, 1~10 The greater the PID control intensity, the faster it will reach the set temperature, but the greater the overshoot will be
#60	WR	Channel 2	
#61	WR	Channel 3	
#62	WR	Channel 4	
#63	WR	Channel 1	Relay output dead time, 1~32767(ms) The minimum time interval of continuous action of relay output. When the time is less than this value, the relay output will not act.
#64	WR	Channel 2	
#65	WR	Channel 3	
#66	WR	Channel 4	
#67	WR	Channel 1	Heating PID sampling period 1~32767(ms)
#68	WR	Channel 2	
#69	WR	Channel 3	
#70	WR	Channel 4	
#71	WR	Channel 1	Heating digital output When using the relay control mode, it indicates the duty cycle (0.00~100.00%)
#72	WR	Channel 2	
#73	WR	Channel 3	
#74	WR	Channel 4	
#75	R	Channel 1	Heating digital/relay output Digital output, 0~10000 Relay output, only output 1 and 0 If it is 1 means the relay is on, 0 means the relay is off
#76	R	Channel 2	
#77	R	Channel 3	
#78	R	Channel 4	
#81	WR	Channel 1	Cooling digital output When using the relay control mode, it indicates the duty cycle (0.00~100.00%)
#82	WR	Channel 2	
#83	WR	Channel 3	
#84	WR	Channel 4	
#85	WR	Channel 1	Cooling digital/relay output Digital output, 0~10000 Relay output, only output 1 and 0 If it is 1 means the relay is on, 0 means the relay is off
#86	WR	Channel 2	
#87	WR	Channel 3	
#88	WR	Channel 4	
#95	R	Channel 1	Stop outputting temperature difference, unit: 0.1°C When sampling temperature-set temperature is greater than this value, stop heating output When the set temperature-sampling temperature is greater than this value, stop cooling output
#96	R	Channel 2	
#97	R	Channel 3	
#98	R	Channel 4	

Remarks: For the temperature control mode using single-side heating PID control, it is fully compatible with 52213-0000 and the previous software's BFM setting mode for 4TC.

Note: After starting the temperature control module, it is forbidden to rewrite the temperature control mode setting. After the temperature control module stops, you can rewrite the variables.

It is recommended not to modify PID related parameters when the temperature control module is running.

#### 10) Explanation of part of the BFM area

BFM#100 Relay output: The on-off duty cycle of relay output, the output unit is percentage, the default range is 0~100. Take the heating control of the first channel as an example:

When BFM#75=1, it means heating is on; when BFM#75=0, it means heating is off;

Digital output: When the heating device is controlled by an analog signal, the digital output mode can be selected. The user can perform digital-to-analog conversion according to the output digital signal to control the analog signal. The default is 0~32767. If you set the output upper and lower limits, suppose it is set to 0 and 10000. Then the minimum output is 0 and the maximum is 10000.

PID control: select the PID control mode, the temperature control module can automatically adjust the temperature according to the PID parameters.

ON/OFF control: ON/OFF control is the most basic on-off control. When the current temperature is less than the set temperature, heating is started, and when the current temperature is greater than the set temperature, heating is stopped.

Single-side control/heating and cooling control: When selecting single-side control, only the heating end has output; when selecting heating and cooling control, both heating and cooling ends have output.

Direct action/reverse action: When the temperature control is a refrigeration device, the positive action should be selected, and the active action is used as cooling at this time; when the temperature control is a heating device, the reverse action should be selected, and the active action is used as the heating at this time.

PID auto-tuning: When Bit4 of BFM#100=1, it will enter PID auto-tuning state first. When auto-tuning is over, Bit4 will automatically become 0, and then it will automatically enter PID adjustment state.

Note: When Bit1=1 of BFM#100, ON/OFF control is selected, and PID auto-tuning function is invalid at this time.

Temperature curve control: When the temperature curve control is selected, the set temperature will start from the current temperature and gradually increase to the target temperature in the set temperature curve mode.

BFM#101 parameter limit setting: When a bit of BFM#101 is 1, it means that the corresponding parameter limit is performed. If the limit is exceeded, an alarm will be generated to remind the user.

BFM#110~BFM#114 PID parameters: The temperature control module controls the temperature according to the PID parameters. The sampling period and filter coefficient need to be set by the user. If the auto-tuning mode is selected, the system will automatically assign the tuning parameters to the BFM after the auto-tuning is over. #110~BFM#112; If the auto-tuning mode is not selected, it needs to be set by the user.

Note: The sampling period of auto-tuning should be much larger than the sampling period of PID control, and the

user needs to change it here.

**BFM#116 Auto-tuning output value:** If the output value needs to be set during auto-tuning, input the output value into this variable at this time. The auto-tuning output value is the same as the output value. If BFM #101 has an output limit, the auto-tuning output value will not exceed the values of BFM#143 and #BFM144. If there is no limit, the digital output defaults to 0~32767, and the self-tuning output value range of the relay mode is 0~100. The unit is a percentage (assuming that 1000 exceeds 100, an output value overrun error will be reported). If the limit is exceeded, the program will automatically correct it to Limit value, and output alarm appears at the same time.

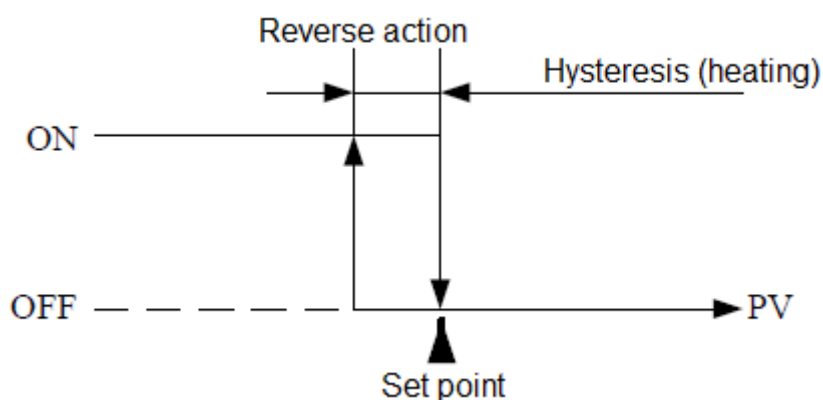
**BFM#117 Current target temperature:** Set the data to be read-only. It is only valid when the temperature curve mode is selected. During temperature curve control, the target temperature will increase in stages according to the set parameters, and BFM#117 displays the current target temperature value.

**BFM#118 PID pause output value:** When 1 is set in BFM#55~BFM#58, PID control stops, and the output value at this time is controlled by BFM#118. When PID control is restarted, PID control will continue, and if the sensor is disconnected, the output value is also controlled by BFM#118. This parameter is only valid in PID control mode.

**BFM#121~BFM#122 Relay output control cycle:** When the relay output control mode is selected, the relay output control cycle parameters need to be set; although a shorter cycle, it can provide better control performance.

For example: set 2s, when the output is 50, the system will heat on for 1s, and the heating will stop for 1s; if the output is 80, the system will heat on for 1.6s, and the heating will stop for 0.4s.

**BFM#124~BFM#125 Hysteresis:** In ON/OFF control, hysteresis is used for stable operation when switching between ON and OFF. The control output (heating) and control output (cooling) can be set in the "lag (heating)" and "lag (cooling)" parameters respectively. The function of heating hysteresis is: the value obtained by subtracting the heating hysteresis from the set temperature is the ON/OFF critical point. It only affects the heating function regardless of the positive action or the reverse action.



**BFM#126 Cooling coefficient:** It is valid when PID control and heating and cooling control are selected at the same time. When PID control is used, the cooling side only has proportional control, and the cooling coefficient represents the P value of the cooling side.

**BFM#130~BFM#131 Temperature curve parameters:** When selecting temperature curve control, two parameters, BFM#130 and BFM#131, need to be set. BFM#130 represents the ramp time unit, and BFM#131

represents the change value of the unit temperature.

For example: BFM#130=0, BFM#131=10, it means that the target temperature will increase by 1°C in 1 minute from the current temperature until it reaches the set temperature.

BFM#140~BFM#154 Restriction parameters: When a certain position of BFM#101 is 1, it is necessary to set related restriction parameters. If a parameter exceeds the set limit during the control process, an alarm will be issued. If the upper and lower limits of the output are set, for example, 30 and 70 respectively, it means that the maximum output is 70% and the minimum output is 30%.

Input and output change limit: indicates the change limit of PID control for each sampling, ON/OFF and auto-tuning do not have this function.

Note: The upper limit of each limit to be set cannot be less than the lower limit, otherwise it will shut down and alarm.

BFM#156 Running status: Display the current running status.

BFM#157 Shutdown alarm: When a fatal error is encountered during the control process, the temperature control function will stop, and BFM#157 will display the corresponding alarm code.

BFM#158 Running alarm: When a certain limit parameter exceeds the limit, the corresponding alarm code will be displayed by BFM#158, but the exceeding limit will not stop and the system will continue to run. The alarm here is only used to remind the user.

BFM#160~BFM#199 PID running intermediate value: The intermediate value cannot be written into data.

## 11) How to use auto-tuning

Use the auto-tuning function in order to get the best PID control, use the step response method to automatically set important constants: action direction, proportional gain, integral time, derivative time.

Self-tuning method:

- a) Transmit the auto-tuning output value to BFM#116. The auto-tuning output value should be used within the range of 50%~100% of the maximum output according to the output device.
- b) Set auto-tuning related parameters (sampling time, input filtering, etc.) and target values.
- c) After the auto-tuning is over, the auto-tuning parameters are automatically saved in the corresponding variables (if the PID function is turned on, the PID calculation function will continue).

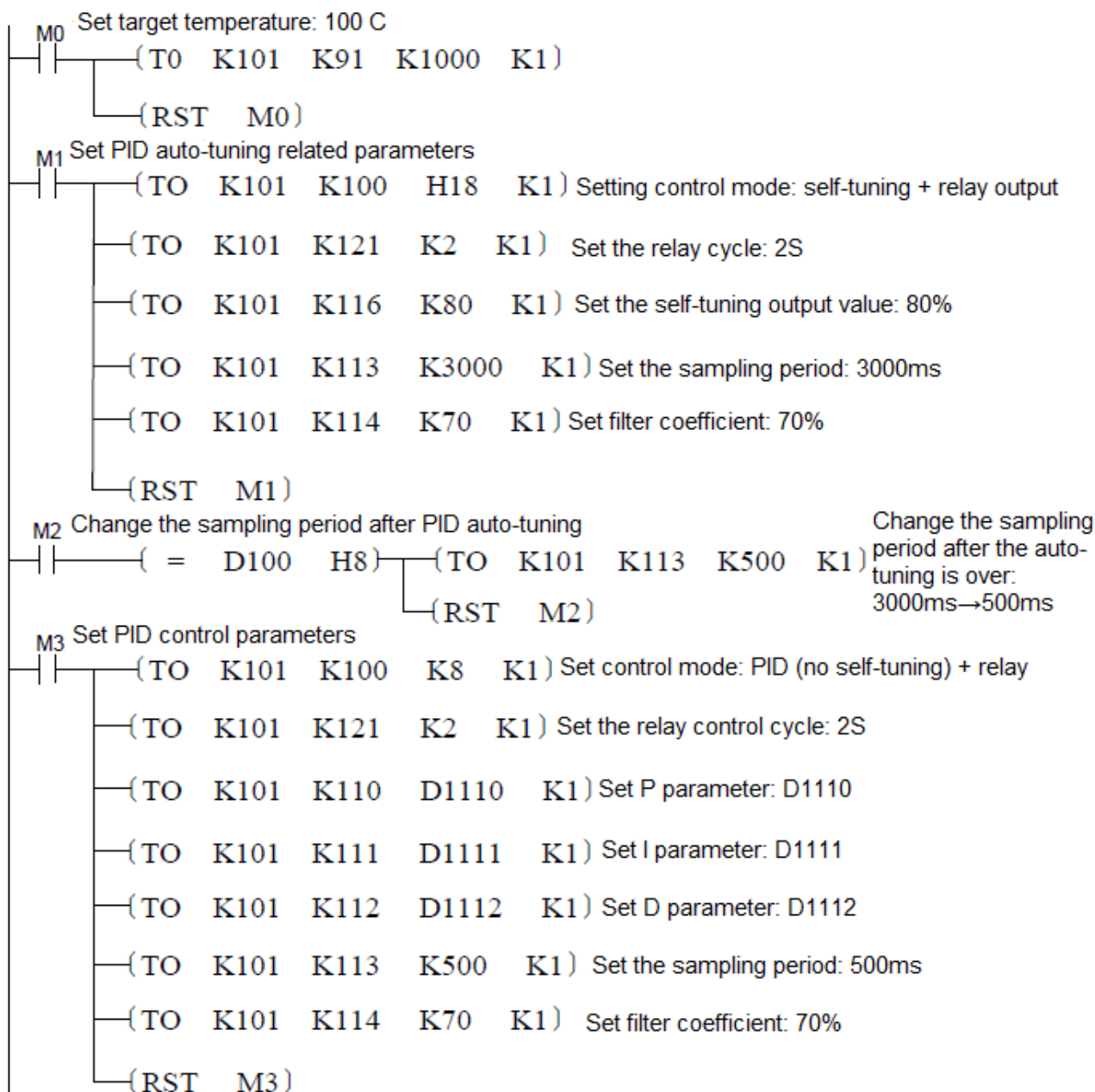
Note:

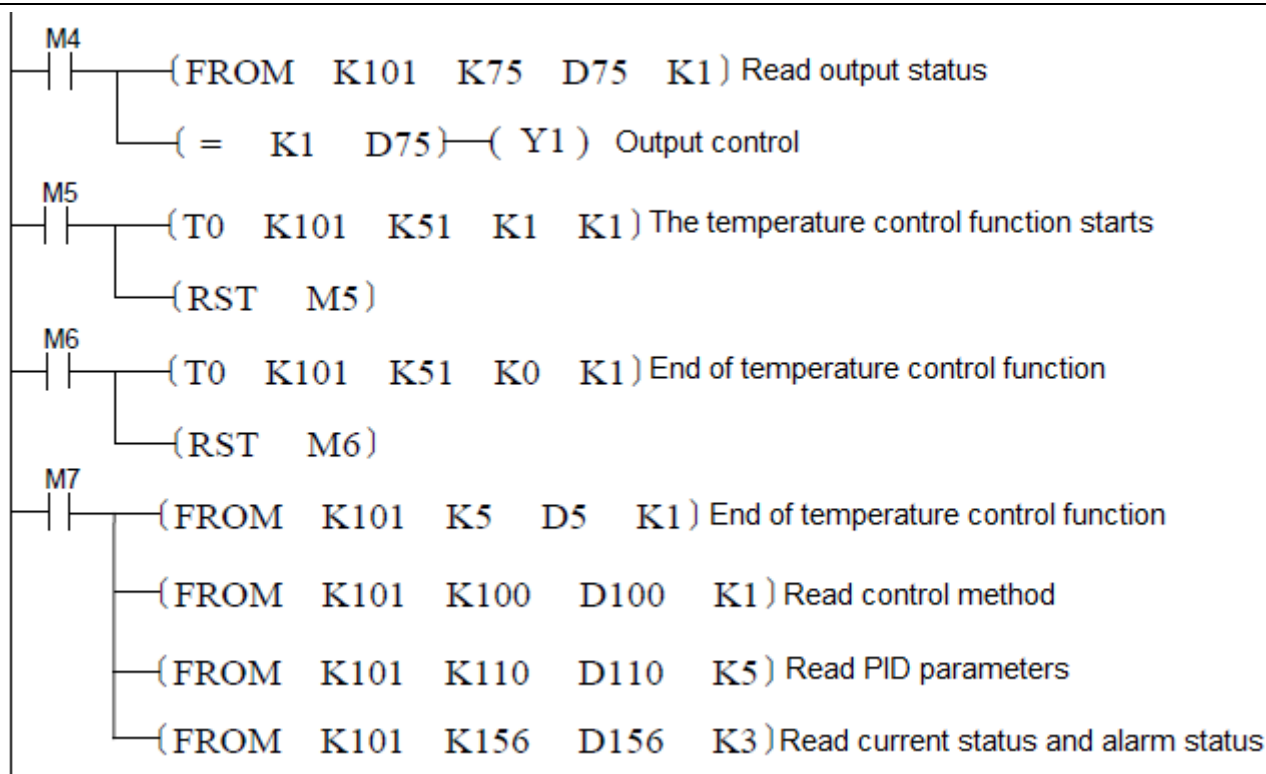
- a) When performing auto-tuning, if the above-mentioned auto-tuning methods (1) and (2) are not satisfied, the situation of incorrect auto-tuning may occur. Please pay special attention.
- b) If the difference between the measured value and the target value at the start of auto-tuning is less than 150, auto-tuning cannot be performed. Therefore, if it is below 150, set the target value of auto-tuning first, and set the target value again after auto-tuning is completed.
- c) The sampling time during sampling time auto-tuning must be more than 1 second (1000ms). In addition, it is recommended to use a time value much longer than the output change period for this sampling time.
- d) Please start the auto-tuning function when the system is in a stable state. If it starts in an unstable state, auto-tuning cannot be performed correctly.

## Precautions for use:

It is forbidden to plug and unplug under power. Only when the main module and application system are out of power, can the expansion module be connected or removed to ensure personal safety and prevent damage to the device due to plugging and unplugging.

Example: A H2U-4TCR-XP remote expansion module, CAN station number is 1, with CH1, relay output, set temperature: 100°C. The user program is as follows:







## 6.7 Expansion Card Instructions

### 6.7.1 Instructions for H2U Series 6A/6B Expansion Card

#### 1) Introduction

H2U-6A-BD expansion card can realize 2 channels of voltage input (0V~10V), 2 channels of current input (0~20mA), 2 channels of voltage and current analog output, which can meet the application requirements of analog detection and control.

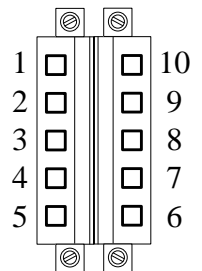
The H2U-6B-BD expansion card can realize 4 channels of current (0-20mA) input and 2 channels of voltage and current analog output, which can meet the application requirements of analog detection and control.

Model	Number of independent channels	Input ports		Output ports	
		0~10V type	0~20mA type	0~10V type	0~20mA type
H2U-6A-BD	4 input, 2 output	V1/V2	I3/I4	V1/V2	I1/I2
H2U-6B-BD	4 input, 2 output	—	I1/I2/I3/I4	V1/V2	I1/I2

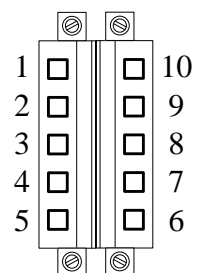
#### 2) Function definition of wiring port

6A expansion card pin function:

Pin	Signal	Function description	Pin	Signal	Function description
1	V1+	Ch1 voltage signal input terminal	10	VO2+	Ch2 voltage output terminal
2	V2+	Ch2 voltage signal input terminal	9	IO2+	Ch2 current output terminal
3	I3+	Ch1 current signal terminal	8	V01+	Ch1 voltage output terminal
4	I4+	Ch2 current signal terminal	7	IO1+	Ch1 current output terminal
5	GND	Input GND	6	GND	Output GND



Pin	Signal	Function description	Pin	Signal	Function description
1	I1+	Ch1 current signal input terminal	10	VO2+	Ch2 voltage output terminal
2	I2+	Ch2 current signal input terminal	9	IO2+	Ch2 current output terminal
3	I3+	Ch3 current signal input terminal	8	V01+	Ch1 voltage output terminal
4	I4+	Ch4 current signal input terminal	7	IO1+	Ch1 current output terminal
5	GND	Input GND	6	GND	Output GND



#### 3) Programming reference

The PLC main module and the expansion card adopt the communication frame for data exchange. The communication frame is automatically organized and sent and received by the PLC system software. When using, the user program only needs to write the type and output value of the expansion card that needs to be accessed in a specific system buffer, and read the measured data from the input port in a specific register. Two functions can be achieved by setting D8220: 1. When D8220 writes C000H to C000H, the expansion card is suspended, and each channel can only read the current value and cannot modify the parameters; if the

expansion card is re-enabled, please clear D8220 to 0. 2. When D8220 is written into C001H, the filter time can be modified. After modification, please keep it for 200ms, and then clear D8220 to 0. See the table below:

#### Set expansion card pause command:

PLC sends to expansion card	Write value when accessing expansion card
D8220	C000H pause command. The relevant data cannot be modified at this time.
D8221~D8228	Reserved
D8229	PLC system automatically calculates, user program cannot be changed
PLC receives from expansion card	Data returned by the expansion card
D8230	The C000H suspension command has taken effect.
D8231	The current value of analog input channel 1 (range: 0-10000)
D8232	The current value of analog input channel 2 (range: 0-10000)
D8233	The current value of analog input channel 3 (range: 0-10000)
D8234	The current value of analog input channel 4 (range: 0-10000)
D8235	The current value of analog output channel 1 (range: 0-10000)
D8236	The current value of analog output channel 1 (range: 0-10000)
D8237~ D8238	Reserved
D8239	Communication error count, if the communication is normal, it will be automatically set to 0; if it is not 0, it means the number of consecutive communication errors

#### Set filter time:

PLC sends to expansion card	Write value when accessing expansion card
D8220	C001H write filter time command.
D8221	Filter time. Range 1~32767 Unit: ms The default filter time is 1ms, if it exceeds the set range, it will be processed according to the default time.
D8222~D8228	Reserved
D8229	PLC system automatically calculates, user program cannot be changed
PLC receives from expansion card	Data returned by the expansion card
D8230	C001H write filter time command has taken effect.
D8231	The currently effective filter time. Range 1~32767 Unit: ms
D8232~ D8238	Reserved
D8239	Communication error count, if the communication is normal, it will be automatically set to 0; if it is not 0, it means the number of consecutive communication errors

### Related registers and their definitions when using expansion cards:

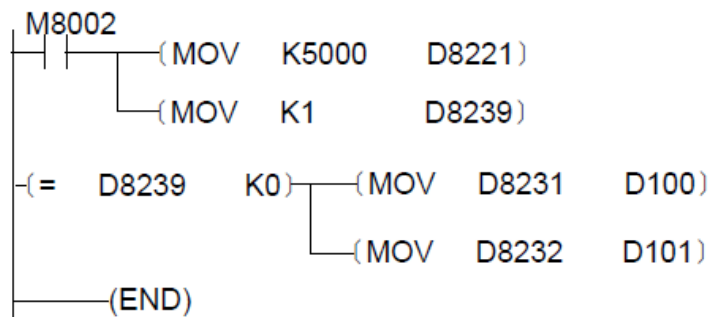
PLC send to expansion card	Writing value during access to H2U-6A-BD	Writing value during access to H2U-6B-BD
D8220	/	/
D8221	Setting value of analog output channel 1 (range: 0-10000)	Setting value of analog output channel 1 (range: 0-10000)
D8222	Setting value of analog output channel 2 (range: 0-10000)	Setting value of analog output channel 2 (range: 0-10000)
D8223~D8238	Reserve	Reserve
PLC receives from expansion card	H2U-6A-BD response data	H2U-6B-BD response data
D8230	6A42h	6B42h
D8231	The current value of analog input channel 1 (range: 0~10000) corresponds to 0~10V	The current value of analog input channel 1 (range: 0~10000) corresponds to 0~20mA
D8232	The current value of analog input channel 2 (range: 0~10000) corresponds to 0~10V	The current value of analog input channel 2 (range: 0~10000) corresponds to 0~20mA
D8233	The current value of analog input channel 3 (range: 0~10000) corresponds to 0~20mA	
D8234	The current value of analog input channel 4 (range: 0~10000) corresponds to 0~20mA	
D8235	The current value of analog output channel 1 (range: 0~10000) Corresponding to 0~10V or corresponding to 0~20mA	
D8236	The current value of analog output channel 2 (range: 0~10000) Corresponding to 0~10V or corresponding to 0~20mA	
D8237~ D8238	Reserve	Reserve
D8239	Communication error count, if the communication is normal, it will be automatically set to 0; if it is not 0, it means the number of consecutive communication errors	

From the description of the above table, we can see that the programming method of 6A/6B expansion card is the same, only need to read (input) or write (output) operation to the dedicated D variable of the system. The following is a programming example of 6A expansion card.

Instructions programming example 1:

Use the H2U-6A-BD card, require the AO output channel to output 5V; put the sample value of the AI1 input channel into D100, and the sample value of the AI2 input channel into D101.

The procedure is as follows:



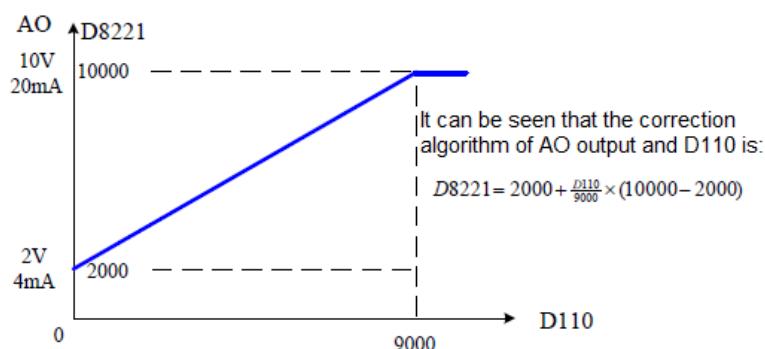
Note: When the value of D8239 is not 0, it means that the PLC has not read the value of the expansion card correctly. Pay attention when programming; the output value D8221 of AO power-on can be refreshed in the user

program according to the specific application.

#### Application programming example 2:

Following the example above, using the H2U-6A-BD card, put the sample value of the AI1 input channel into D100, and the sample value of the AI2 input channel into D102; but the AO current output is required to be related to the current value of the D110 register, when D110=0 When D110=9000, it outputs 4mA; when D110=9000, it outputs 20mA.

Analysis: The current output signal and voltage output signal of AO are respectively [0~20mA] or [0~10V] corresponding to [0~10000] of D8221 special register. If 4~20mA corresponds to user register value, user program must correct the value and write it to D8221 to get the desired signal current. (Note: The voltage signal characteristics of AO have also changed after correction.)



The procedure is as follows:

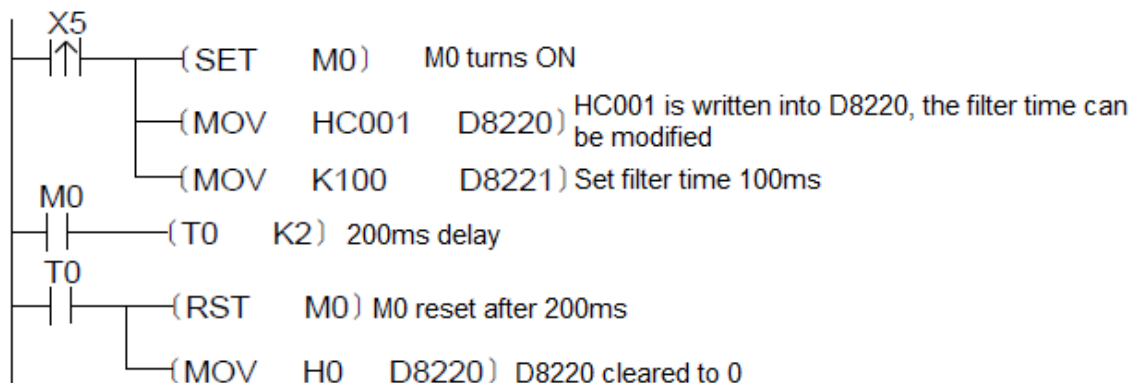
```

M8002
| | (MOV K1 D8239)
| |
| (= D8239 K0) | (MOV D8231 D100)
| | | (MOV D8232 D102)
| |
M8000
| | (> D110 K9000) | (MOV K9000 D110)
| | |
| | (MOV D110 D0) //In the calculation, the result of the multiplication operation will exceed
| | | // 1 word, and 32bit operation is required
| | (MOV K0 D1) //Place D110 in (D1, D0) 32bit temporary unit
| | (DMUL D0 K8000 D2) //Multiply (D1, D0) by 8000, and store the result (D3, D2)
| | (DDIV D2 K9000 D4) //Divide (D3, D2) by 9000, and store the result in (D5, D4)
| | | // temporary unit
| | (ADD D4 K2000 D8221) //Add 2000 to (D4) and write to AO unit
| |
| (END)

```

Application programming example 3:

Set the filter time to 100ms, the procedure is as follows:

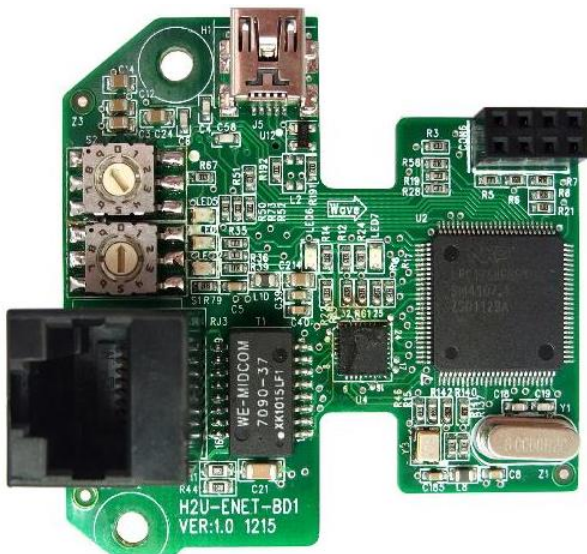


### 6.7.2 Instructions for H2U Series Ethernet Communication Expansion Card

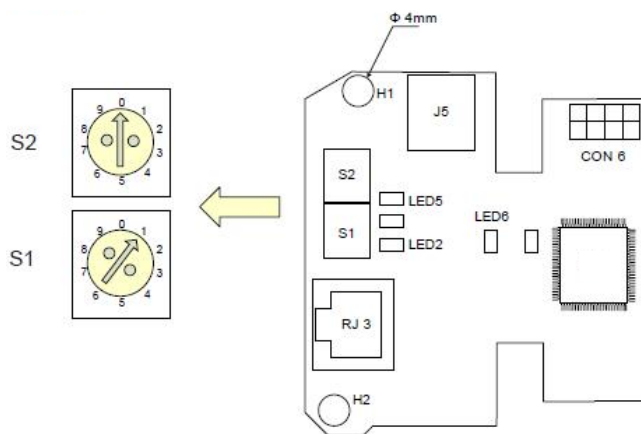
#### H2U-ENET-BD Ethernet card installation instructions

To implement Ethernet communication of H2U-XP series PLCs, configure H2U-ENET-BD Ethernet communication card on the H2U-XP series PLC main module. After the installation, PLC can access the TCP/IP Ethernet and data switching can be implemented using Modbus TCP at the application layer. (The master and slave stations can have a maximum of 8 devices, with at most 100 meters communication distance.)

The product does not have independent power supply and cannot work independently. It must work with the PLC main module. The Ethernet extension card must be inserted into the PLC functional extension card socket, and fixed with screws.



H2U-ENET-BD Ethernet communication card appearance



H2U-ENET-BD Ethernet communication card terminal distribution

## Terminal distribution and function description

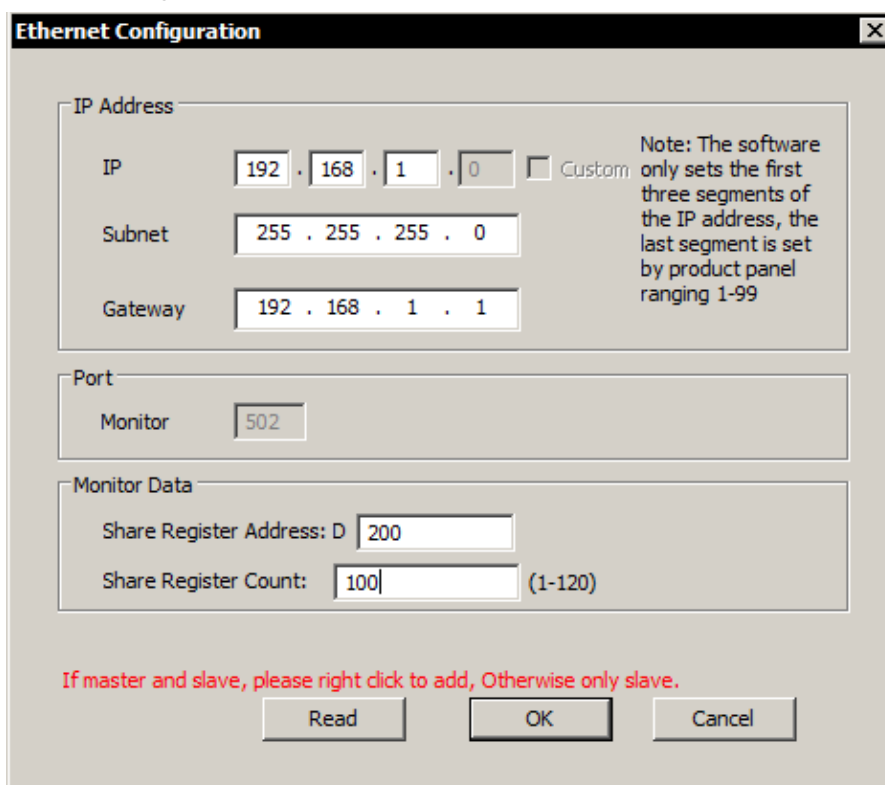
Item	Terminal Name	Function
RJ45	RJ45 interface	Ethernet communication interface
S1	DIP switch	The unit digit in the last segment of the IP address (0-9)
S2	DIP switch	The tens digit in the last segment of the IP address (unit + tens digits: 0-99)
H1/H2	Screw hole	The H2U-ENET-BD uses 6 M3 tapping screws.
J5	Mini USB	Manufacturer reserved
LED 5	Communication indicator (green)	When the communication between Ethernet and PLC is normal, the indicator blinks. When the communication is interrupted, the indicator is off.
LED 2	Communication failure indicator (red)	If the red indicator is on, the Ethernet card cannot communicate with the PLC. If the indicator blinks, communication between the Ethernet card and other devices is abnormal. For fault information, see Error Code Information.
LED 6	Network Card Indicator	-
CON 6	Programming port	Manufacturer reserved

## Ethernet card parameter setting

## 1) Ethernet slave configuration

When the Ethernet card is used as a slave (server), it supports reading holding registers (0x03), writing a single register (0x06), and writing multiple registers (0x10). Other function codes are temporarily not supported. The host computer can access the PLC with Ethernet card and read and write the component data in the PLC. At this time, the Ethernet slave needs to be configured. The user program sets D8270 to C1h (decimal 193), D8271 to 1, and D8276 to 2.

- a) In the "Project Management" window of the AutoShop software, double-click "Ethernet Configuration" to open the "Ethernet Configuration" window.



**Ethernet Configuration**

IP Address

IP: 192 . 168 . 1 . 0  Custom

Subnet: 255 . 255 . 255 . 0

Gateway: 192 . 168 . 1 . 1

Note: The software only sets the first three segments of the IP address, the last segment is set by product panel ranging 1-99

Port

Monitor: 502

Monitor Data

Share Register Address: D 200

Share Register Count: 100 (1-120)

If master and slave, please right click to add, Otherwise only slave.

Read OK Cancel

## b) IP address

- i. IP address: It is the identity of the device in the network. The uniqueness of each device's IP address must be ensured, otherwise the device will not be able to connect to the network. The IP address is set by the AutoShop software, only the first 3 segments are set, and the last segment is controlled by the DIP switch on the Ethernet card panel, and the value range is 1-99.
- ii. Subnet mask: The function of the subnet mask is to address multiple physical networks under the same network address. The mask is used to divide the subnet address and the device address of the host ID. The method of obtaining the subnet address is to reserve the bit corresponding to the position of the mask containing 1 in the IP address, and then replace other bits with 0. If there is no special requirement, the subnet mask is 255.255.255.0.
- iii. Gateway: It can route network messages to devices not in the current network. If there is no gateway, the gateway address is set to 0.0.0.0.

## c) Port

TCP port 502 is reserved for MODBUS communication, and the Ethernet card is forced to listen to this port.

## d) Monitor Data

Shared register address: Set the first address of the PLC word component that the user wants to access, and it also serves as the data storage area of the master station.

Number of shared registers: the number of shared registers.

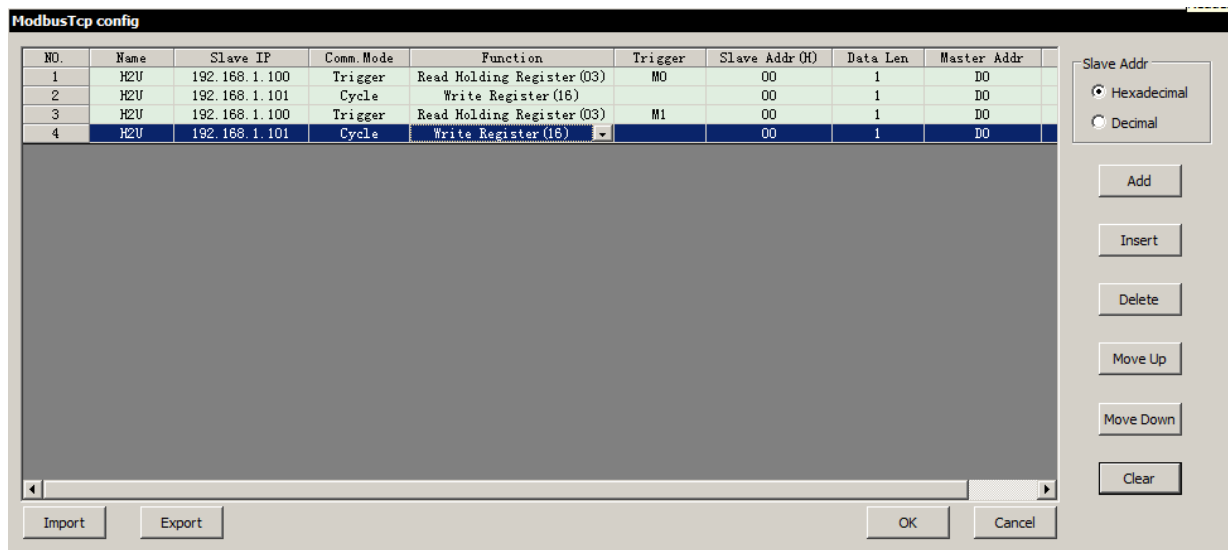
For example, the setting in the figure above can read or change the value of any register of D200-D300 through ModbusTCP.

In order to ensure the data security in the PLC main module, this product can only monitor a continuous segment of D register, the length does not exceed 120 words. If other components need to be monitored, they can be synchronized through the PLC user program. For example, to monitor M10, you can write: "MOV M10 D200" instruction in the user program. At this time, monitoring D200 is equivalent to monitoring M10.

## 2) Ethernet master station configuration

The Ethernet card can be used as a host (client), supporting reading coils (0x01), reading holding registers (0x03), writing a single coil (0x05), writing multiple coils (0x0F), writing multiple registers (0x10), to access MODBUS-TCP-enabled slave devices, other function codes are temporarily not supported.

- a) In the "Ethernet Configuration" window, click the "Ethernet Master Configuration" button to open the "MODBUS-TCP Config" window;



### b) Configuration interface:

New: Add a new configuration record at the end of configuration data

Insert: Add a configuration record at the current position

Delete: delete the selected configuration record

Move up: adjust the position of the selected row

Move down: adjust the position of the selected row

Clear: Clear all configuration information in the window

OK: save the changes, close the window

Cancel: Do not save changes, close the window

### c) Configuration data

Communication methods: 1) Cycle: it will be executed once in each communication cycle; 2) Trigger: When the trigger condition is ON, it will be executed once in each communication cycle, otherwise it will not be executed.

Function: Including read register, write register, read coil, write coil.

Trigger condition: When the communication mode is trigger, the trigger condition needs to be set. The trigger element can be M, S, T, C element.

Slave register address: the address of the slave register in the Modbus protocol, which must be hexadecimal, and the data must be readable and writable.

Master buffer address: the storage address when the Modbus command reads back data. It must be within the shared register address range set by the slave, otherwise it will report "not a valid master buffer address".

Port number: The port number for TCP/IP link, determined by the specific device settings. When Inovance PLC is used as a slave station, the port number is fixed at 502.

### 3) Examples

For example, there is a PLC A which is used as a ModbusTCP master station, IP address is set to 192.168.50.54, you need to periodically read the register D500 of PLC B with the IP address of 192.168.50.12, and save it to the register D205 of PLC A.

#### a) Set the Ethernet slave and master parameters of PLC A



- Set IP address

Set the tens place and ones place of the Ethernet card IP address DIP switch to 5 and 4 respectively. At the same time, set other IP parameters in the software.

IP Address configuration window showing IP: 192 . 168 . 50 . 0, Subnet: 255 . 255 . 255 . 0, Gateway: 192 . 168 . 50 . 1. A note states: "Note: The software only sets the first three segments of the IP address, the last segment is set by product panel ranging 1-99".

- Set up shared registers

Since the read data needs to be stored in D205 of PLC A, D205 needs to fall within the shared register address range. Set as follows, the shared register address range is D200-D300.

Monitor Data window showing Share Register Address: D 200 and Share Register Count: 100 (1-120).

- Master station configuration

Open the Ethernet master configuration interface, set the IP address of the slave station (IP address of PLC B), read cyclically, the hexadecimal corresponding to D500 is 1F4, read 1 data length, and put the read data into PLC A master station buffer D205.

**ModbusTcp config**

NO.	Name	Slave IP	Comm. Mode	Function	Trigger	Slave Addr (H)	Data Len	Master Addr
1	H2U	192.168.50.12	Cycle	Read Holding Register (03)		1F4	1	D205

b) Set the slave station parameters of PLC B

PLC B does not need to set up a master station, it only serves as MODBUS-TCP server for PLC A to read data.

- Set IP address

Set the tens place and ones place of the Ethernet card IP address DIP switch to 1 and 2 respectively. At the same time, set other slave parameters in the software.

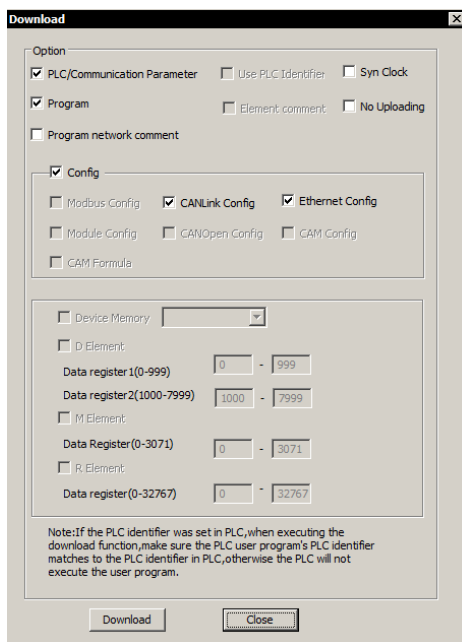
IP Address configuration window showing IP: 192 . 168 . 50 . 0, Subnet: 255 . 255 . 255 . 0, Gateway: 192 . 168 . 50 . 1. A note states: "Note: The software only sets the first three segments of the IP address, the last segment is set by product panel ranging 1-99".

- Set up shared registers

The shared register provided by PLC B for PLC A to access is D500, so D500 needs to fall into the shared register area. As follows, the shared register area is D500-D600.

Monitor Data window showing Share Register Address: D 500 and Share Register Count: 100 (1-120).

- c) Download the Ethernet configuration to PLC A and PLC B respectively. Check the Ethernet configuration when downloading:



- d) After the download is complete, PLC A and PLC B need to be powered off and on for the configuration to take effect.
- 4) Error code information

When the Ethernet communication error occurs, the error information can be viewed through the value of PLC register D8063, as shown in the following table:

Type	Error Code	Error Content
D8063 Ethernet communication error	81xxh Not supported function code.	Change the IP address and configure Modbus function code for xx master station.
	82xxh Number of registers exceeded limit	Check whether the data length in the IP address configured for the xx master station exceeds the limit.
	83xxh Register address exceeded range	Check whether the register address of the slave station configured for the xx master station exceeds the range.
	84xxh Failed to connect to the slave station remotely	Check whether the device with IP address xx can be connected.
	85xxh Communication error on serial port	Check whether the connection between Ethernet card and PLC is normal. Check whether D8270 is C1h (193 in decimal form). Check whether D8271 is 1.
Remarks: In the error code, xx indicates the IP address (last segment) 00h-FFh of the device with the communication error.		

For example, if PLC A (IP address 192.168.50.54), as the master station, connects to the remote slave station of PLC B (IP address: 192.168.50.12), when the connection fails, the data of D8063 is 840Ch, which means PLC A and remote PLC B connection error.

### 6.7.3 Instructions for H1U/ H2U Series Communication Expansion Card

H2U can identify expansion card type number automatically, D8199 displays the type number of this card; H1U cannot automatically identify, you need to set the expansion card type through D8199. The types of expansion cards are as follows:

1	RS232 expansion card	5	6A/6B expansion card
2	RS422/485 communication expansion card	6	High-speed expansion card
3	AD expansion card	7	CAN expansion card
4	DA expansion card	8	Test

For example, if we need to use RS485 expansion card in H1U, we need to execute [LD M8002] [MOV K2 D8199] in the program. The RS232 expansion card and RS422/485 communication expansion card occupy the main module COM1, and their use and setting methods are consistent with the COM1 port.

#### 1) H2U-232-BD/H1U-232-BD expansion card

H2U-232-BD expansion card is a communication expansion card for H2U series PLC main module (H1U-232-BD expansion card is a communication expansion card for H1U series PLC main module). The DB9 signal socket of the card provides a standard RS232 level, Physically it is the COM1 communication port of the PLC main module, so the communication port of this board and the COM1 port of the original RS485 level of the PLC cannot be used at the same time, otherwise communication conflicts will occur. Using this expansion board, it can realize the communication between PLC and PC, HMI, MODEM, PLC, intelligent instrument and other equipment. The communication format can be determined by programming.

Specifications:

Type of port	DB9, male
Signal standard	RS-232C
Transfer distance	The max distance is 15m, depend on the baud rate
Communication mode	Full duplex, half duplex
Protocol	MODBUS Master, Slave protocol. Self defined protocol
Power supply	5VDC, provided by PLC internal power supply
Signal isolation	No isolation with PLC internal logic circuit

Pin definition:

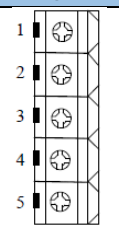
Pin number	Signal name	Function description	Diagram
1	CD	Check terminal of data carrier (input)	
2	RXD	Receiving data terminal (input)	
3	TXD	Sending data terminal (input)	
4	DTR	Sending data request	
5	GND	Signal ground	
6	DSR	Sending enable (input)	
7, 8	—	Internal direct connection between two pins	
9	—	No internal connection	

Please refer to "H2U Communication Expansion Card User Manual" for connection methods and settings with computers, HMIs, etc.

## 2) H2U-485-BD (or H2U-485IF-BD) expansion card

H2U-485-BD (or H2U-485IF-BD, IF: products with isolation, recommended) expansion card is an expansion communication card for H2U series PLC main module. This card provides RS485 level. Connect TX and RX signal lines separately and can be used as RS422 signal port. Physically, it is the COM1 communication port of the PLC main module, so the communication port of this board and the COM1 port of the original RS485 level of the PLC cannot be used at the same time, otherwise communication conflicts will occur. Use this expansion board to realize the communication between PLC and PC, HMI, PLC, intelligent instrument and other equipment, and the communication format can be determined by programming. This card needs M8126 to be ON to be valid.

Port function definition:

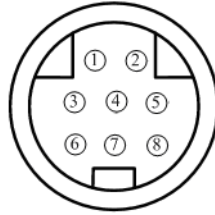
Pin number	Signal name	Function description	Diagram
1	RA	Signal RXD + of RS485	
2	RB	Signal RXD - of RS485	
3	TA	Signal RXD + of RS485	
4	TB	Signal RXD - of RS485	
5	GND	Signal ground	

Please refer to "H2U Communication Expansion Card User Manual" for connection methods and settings with computers, touch screens, etc.

## 3) H2U-422-BD expansion card

The H2U-422-BD expansion card is a communication expansion card for the H2U series PLC main module. The Mini DIN8 signal socket of the card provides RS422 level. It is physically the COM1 communication port of the PLC main module. Therefore, the communication port of the board and the PLC's original RS485 level COM1 port cannot be used at the same time, otherwise communication conflicts will occur. Use this expansion board to realize the communication between PLC and PC, HMI, PLC, intelligent instrument and other equipment, and the communication format can be determined by programming. After H2U series PLC uses this type of expansion card, it can make PC and HMI communicate with it at the same time.

Port function definition:

Pin number	Signal name	Function description	Diagram
1	RB	Signal RXD - of RS485	<p>The function definition of the socket pins is the same as that of the COM0 communication port socket</p> 
2	RA	Signal RXD + of RS485	
3	GND	Singal ground	
4	TB	Signal RXD - of RS485	
5	VDD	5V+	
6	NC	NC	
7	TA	Signal RXD + of RS485	
8	NC	NC	
Metal casing	GND	Singal ground	

Please refer to "H2U Communication Expansion Card User Manual" for connection methods and settings with computers and touch screens.

## 4) H1U-CAN-BD/H2U-CAN-BD expansion card

Note: For detailed introduction of H1U-CAN-BD/H2U-CAN-BD expansion card, please refer to "5.8 CAN Communication Command Description".

## 6.7.4 Instructions for H1U Series 4A Expansion Card

## 1) H1U-4A-BD electrical specifications

Item		Parameter description
AI	Number of input channels	2
	Signal level (differential mode)	Voltage channel: 0V~10V, corresponding to digital output 0~10000 Current channel: 0mA ~20mA, corresponding to digital output 0~10000
	Allow common mode voltage	5Vpp
	Input resistance	Not less than 100kΩ
	ADC sampling resolution	12bit
	Quantization error	0.3%
	Sampling rate	1ms/each channel (sampling rate can be set)
AO	Number of output channels	2
	Signal level	Output voltage: 0~10VDC
	DAC resolution	12bit
	Voltage signal allows minimum load	2kΩ
	Refresh rate	1ms (DA response time is related to user program scan)

## 2) H1U-4A-BD expansion card pin function

Pin	Signal	Function description	Pin	Signal	Function description
1	24V+	Expansion card power supply voltage positive input	7	I2+	Ch2 voltage output terminal
2	0V	Expansion card power supply voltage negative input	8	V/I2-	Channel 2 voltage/current negative input
3	V1+	Channel 1 voltage positive input	9	VO1+	Channel 1 voltage output positive terminal
4	I1+	Channel 1 current positive input terminal, short-circuit with V1+ in application	10	VO2+	Channel 2 voltage output positive terminal
5	V/I2+	Channel 2 voltage/current positive input	11	VO2-	Channel 2 voltage output negative terminal
6	V2+	Channel 2 voltage positive input	-		

Note: The expansion card must be connected to a 24V power supply, otherwise unpredictable measurement values may be generated.

## 3) Programming reference

## Address unit description

Address	R/W attributes	Function description
D8199	WR	Expansion card configuration: H1U-4A-BD card should be configured as 5
D8120	WR	Command register (1)
D8121	WR	Set value of analog output channel 1 (range: 0~10000) (2)
D8122	WR	Set value of analog output channel 2 (range: 0~10000)
D8130	R	Identify the expansion card type: 4A22H (represents H1U-4A-BD card)
D8131	R	Input the current value of channel 1 (range: 0~10000) (2)
D8132	R	Input the current value of channel 2 (range: 0~10000)
D8133	R	The current value of analog voltage output channel 1 (range: 0~10000)
D8134	R	The current value of analog voltage output channel 2 (range: 0~10000)
D8139	WR	Communication error count, if the communication is normal, it is automatically set to 0; if it is not 0, it means the number of consecutive communication errors

Note:

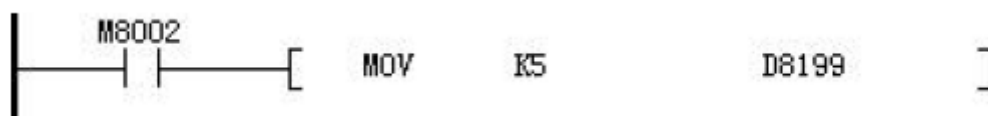
(1) Command description: C000H pause command; C001H filter constant setting command; FFFFH version number view command.

(2) When D8220=C001H, D8221 is the set value of the filter constant, and D8231 is the current value of the filter constant. The filter constants of the other two AI channels are the same and cannot be set separately.

(3) WR means the register is readable and writable; R means read-only, and writing to it is invalid.

### Application programming example

Since H1U cannot automatically identify the type of expansion card, it is necessary to set a specific register D8199 to start the expansion card at the beginning of the user program. The setting statement is as follows:

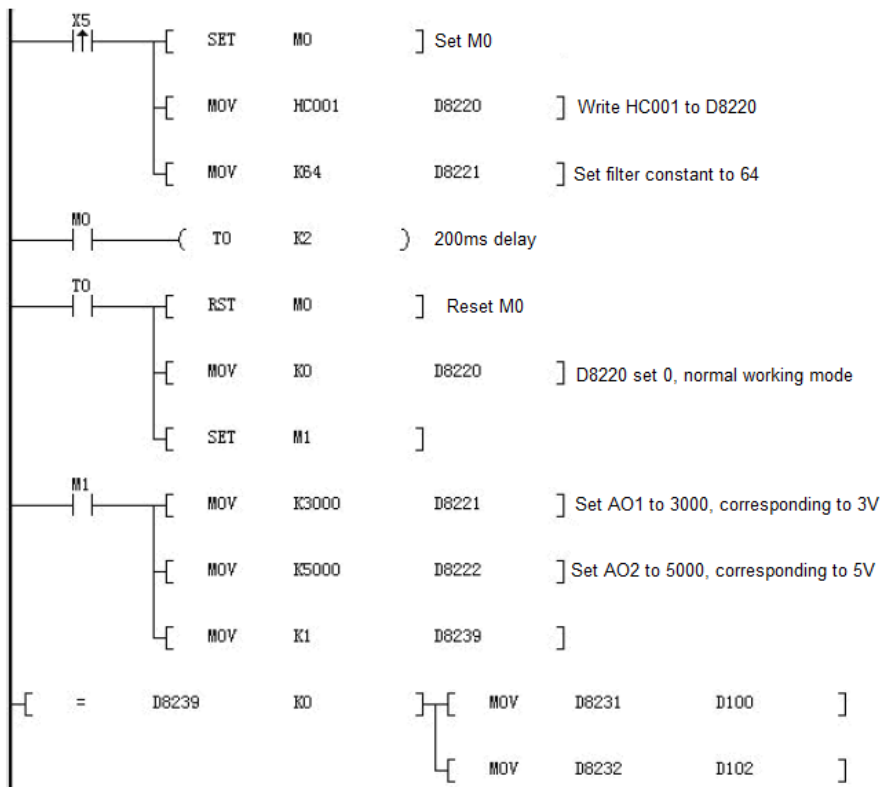


Common use and filter constant setting:

After starting the operation, you can set the number in D8221 and D8222 to determine the output size of AO1 and AO2. The limited range is 0~10000 (corresponding to 0~10V). At the same time, the detection value of AI1 and AI2 can be read from D8231 and D8232. It is 0~10000, when voltage input is selected, it corresponds to 0~10V, when current input is corresponding to 0~20mA.

Note: Only one input can be selected for the same channel, that is, when voltage input is selected, the current input terminal needs to be left floating, otherwise the detection result will be affected.

For example, using the H1U-4A-BD card, it is required to set the filter constant to 64, AO1 outputs 3V, AO2 outputs 5V, and at the same time, put the sample value of the AI1 input channel into D100 and the sample value of the AI2 input channel into D102. The procedure is as follows:



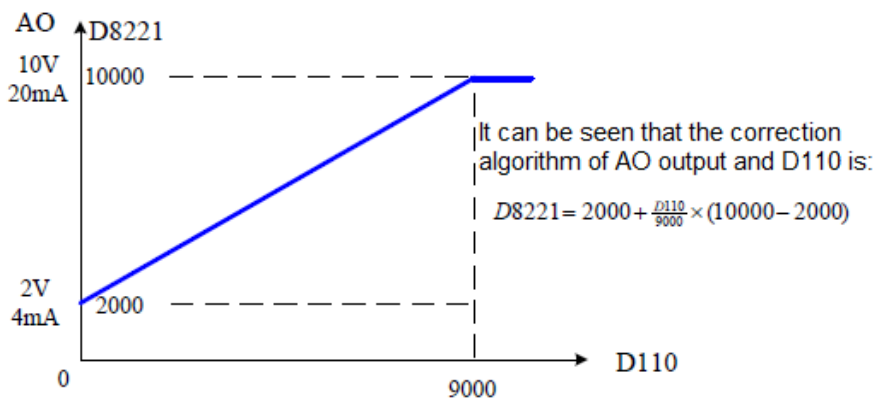
Note: The effective setting range of the filter constant is 1~256. If the written value is out of the range, the effective value written last time will be kept, and the value written over the limit this time will not take effect. The current value of the filter constant is stored in the register D8231 (when D8220=HC001).

When the value of D8239 is not 0, it means that the PLC has not read the value of the expansion card correctly. Pay attention when programming; the output value of AO channel D8221 and D8222 can be refreshed in the user program according to the specific application.

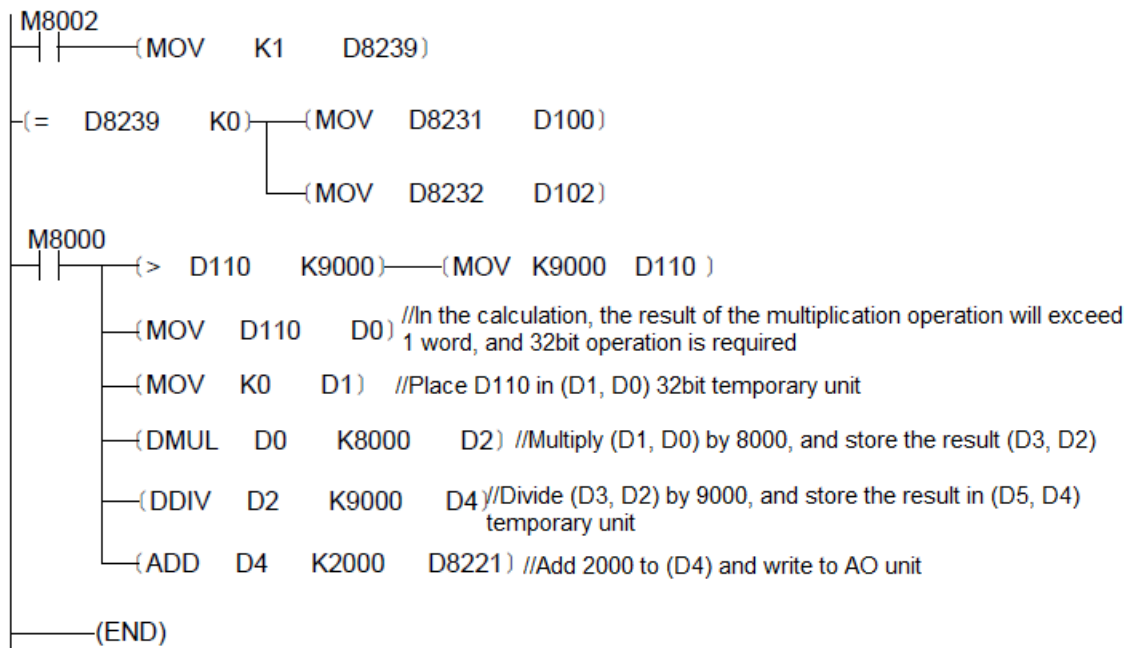
**Offset gain setting**

Continuing the example above, the storage address of the sampled value of the 2 AI channels remains unchanged, and the AO1 voltage output is required to be related to the current value of the D110 register. When D110=0, 2 V is output; when D110=9000, 10V is output.

Analysis: AO voltage output signal, (0~10V) corresponds to D8221 special register (0~10000), if 2~10V corresponds to the user's register value, it needs to be corrected by the user program and written into D8221 to get the desired signal voltage.



The procedure is as follows:





## 6.8 Instructions for use of differential to collector output module

### 1) Electrical specifications of H2U-2DC module

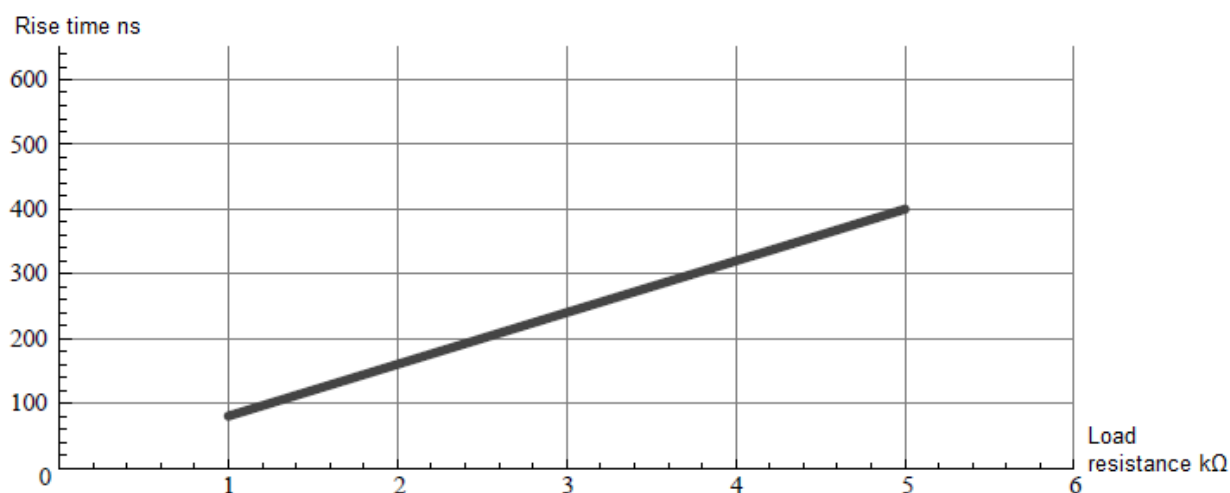
Range:

Parameter	Range
Power supply voltage	32V
Differential input voltage $ (V+)-(V-) $	7V
Collector output peak current	400mA
Collector output working voltage	33V

Characteristic parameter:

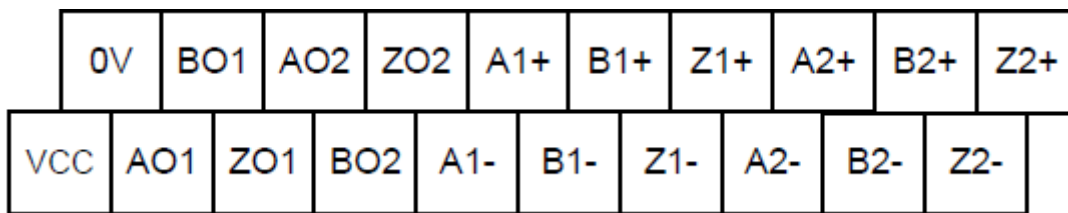
Parameter	Rated	Range	
		Minimum	Maximum
Power supply voltage	DC24V	DC10V	DC30V
Differential input VIL	0V	-6V	1.5V
Differential input VIH	5V	3V	6V
Input low level pulse time	/	600ns	$\infty$
Input high level pulse time	/	600ns	$\infty$
Collector output conduction voltage drop	0.3V (100mA Load)	/	/
Collector output working voltage	24V	9V	30V
Collector output peak current	/	/	20mA
Collector output continuous current	/	/	100Ma
Collector output falling edge	20ns	/	/
Collector output rising edge	/	/	300ns (3K ohm RL)
Maximum operating frequency	/	/	600kHz

The rising edge time of the collector output is related to the load resistance, and the relationship between the rising edge time and the load resistance is as follows:



If the load resistance is too large, the rising edge time of the collector output will become longer. For example, when working at 600kHz, the width of a pulse is 1670ns. If a load resistance of 5kΩ is used, it will cause a rising edge of 400ns, which is about 25% of the entire cycle, so the waveform distortion will be more serious.

2) Terminal definition

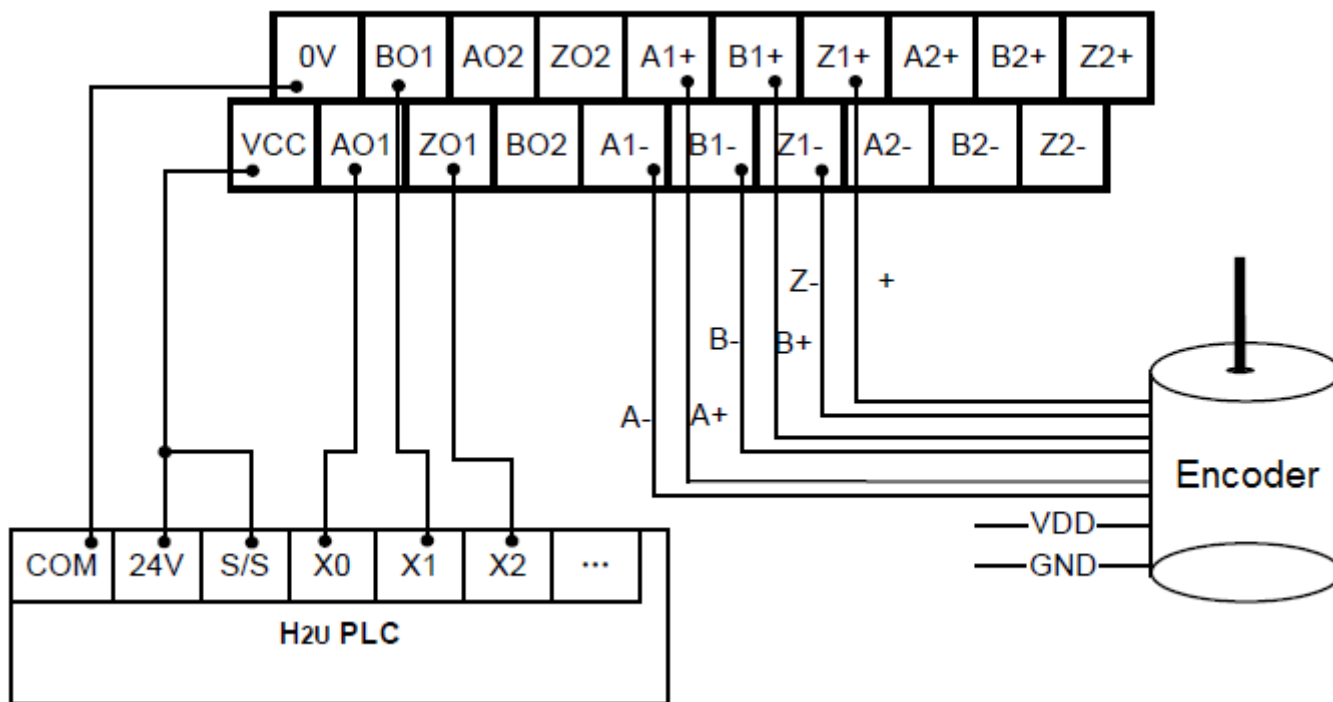


Where, 0V and VCC are the power supply of the module, A1+, A1- constitute a set of differential inputs, and the corresponding collector output is AO1, and the rest are deduced by name.

The input and output of the product are electrically isolated, and the 6 input channels are also isolated, and the common end of the 6 collector output terminals is 0V for the power supply of the product.

3) Application examples of differential to collector output

The differential output signal of the ABZ type differential encoder is converted by this product H2U-2DC, and then connected to the Inovance universal PLC. The connection diagram is as follows:



H2U-2DC uses the external 24V power supply of Inovance PLC. Take A phase as an example. The ABZ type differential encoder outputs A+, A- connects to H2U-2DC A1+, A1-; AO1 connects to PLC X0, phase B and C follow the same logic as phase A. In addition, the S/S pin on the PLC input side is connected to 24V. Then PLC can count ABZ type differential encoder.



## Chapter 7 Features

### Content

7.1 Upload is prohibited.....	- 456 -
7.1.1 Introduction to upload prohibition .....	- 456 -
7.1.2 PLC encryption settings.....	- 457 -
7.2 Project authorization protection use (firmware version 24147 and above support) .....	- 459 -
7.2.1 Function introduction .....	- 459 -
7.2.2 Project authorization protection .....	- 459 -
7.3 Subroutine encryption/decryption.....	- 461 -
7.4 PLC identifier .....	- 462 -
7.4.1 Setting interface of PLC identifier .....	- 462 -
7.4.2 Modification of PLC identifier .....	- 464 -
7.5 Firmware Upgrade Step Guide .....	- 465 -
7.5.1 Preparation .....	- 465 -
7.5.2 Upgrade operation .....	- 465 -

## Chapter 7 Features

### 7.1 Upload is prohibited

#### 7.1.1 Introduction to upload prohibition

1. Check "No Uploading", you cannot upload the downloaded user program and other data; on the contrary, if you don't check "No Uploading", you can directly upload the user program and other data, set the upload password, and upload password. After the verification is correct, the user program and other data can be uploaded;
2. If you want to clear the upload prohibition function, you need to download the user program again. When downloading, do not check the "No Uploading", then the function will be cleared;

[Note]: If the identification code is set in the PLC, you must ensure that the identification code used by the user program matches the one in the PLC when downloading, otherwise the PLC will not be able to execute the user program.

**Download**

Option

PLC/Communication Parameter     Use PLC Identifier     Syn Clock

Program     Element comment     No Uploading

Program network comment

Config

Modbus Config     CANLink Config     Ethernet Config

Module Config     CANOpen Config     CAM Config

CAM Formula

Device Memory

D Element

Data register 1(0-999)    0 - 999

Data register 2(1000-7999)    1000 - 7999

M Element

Data Register(0-3071)    0 - 3071

R Element

Data register(0-32767)    0 - 32767

Note: If the PLC identifier was set in PLC, when executing the download function, make sure the PLC user program's PLC identifier matches to the PLC identifier in PLC, otherwise the PLC will not execute the user program.

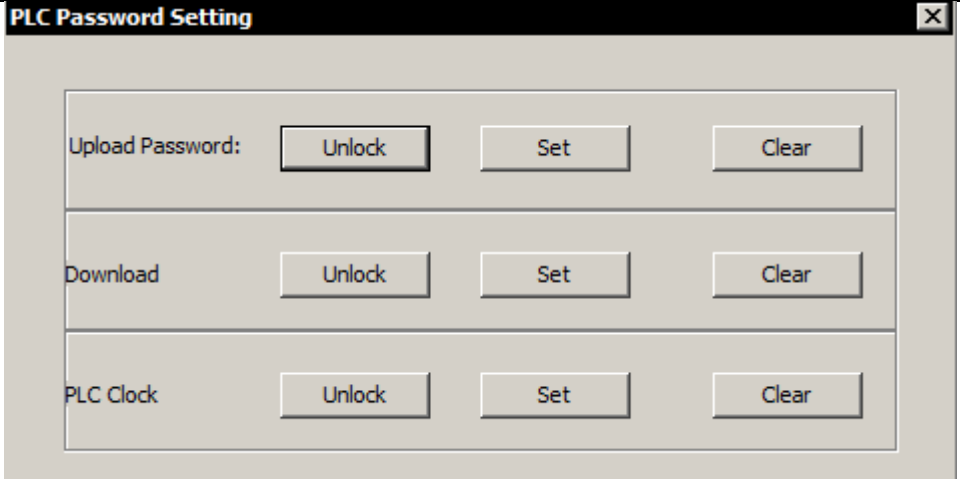
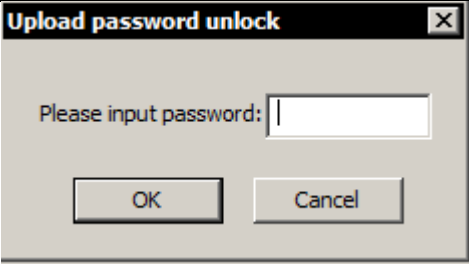
Download    Close

### 7.1.2 PLC encryption settings

The PLC encryption setting function can confirm the authority for PLC upload, download, and clock setting operations; if a password is set, the operation can be completed after successful password verification; The role of upload and download encryption is to protect PLC programs written by users and prevent unauthorized access. After setting the upload password, you need to enter the upload password every time you perform an upload operation. Similarly, after setting the download password, you need to enter the download password every time you perform a download operation.

The password length is 8 digits, which can be numbers, letters, and all other input characters (such as +, \*, / and other characters are also acceptable).

[Note]: When setting the password, the PLC must be in stop state. And after the setting is completed, the password will not take effect immediately. You need to power off the PLC and restart it.

<p>1) Encryption settings Click the menu bar [PLC]- [Encryption Settings] option, and the PLC password setting dialog box will pop up:</p>	
<p>2) Unlock the password Click the "Unlock Password" button in the upload password, if the user has not set the upload password to the PLC before, the system will pop up a prompt box saying "The PLC does not set the upload password, no need to unlock"; otherwise, the right dialog box will pop up :</p>	 <p>After entering the correct password, click "OK". At this time, the message output window will prompt "Upload password unlocking communication command executed successfully". When the user uploads the program, the upload password verification is not required.</p>

### 3) Set password

Click the "Set Password" button in the upload password. If the user has not set the upload password for the PLC before, the pop-up dialog box is shown as the right picture:

As shown in the figure above, you need to enter the new password and confirm the new password, and then click "OK". At this time, the message output window will prompt "Upload password modification communication command executed successfully".

If the user has previously set the PLC upload password, the following dialog box will pop up:

The user needs to enter the correct old password, then enter the new password, and after confirming the new password, click "OK". At this time, the message output window will prompt "Upload password modification communication command executed successfully".

### 4) Clear password

Click the "Clear Password" button in the upload password, and the "Password Check" dialog box will pop up:

If the user has not set the upload password before, after clicking the "OK" button, the system will pop up a prompt dialog box; otherwise, after entering the correct upload password, click "OK", and the message output window will prompt "Upload password clear communication command" execution succeed".

## 7.2 Project authorization protection (available for firmware version 24147 and above)

### 7.2.1 Function introduction

The project that has been authorized and protected can only be used on a specific authorized PLC to bind the user project and the specific PLC, that is, after user A sets the authorization protection for the project 'PrjA', the project can only run on the PLC authorized by user A; The scope of use is restricted;

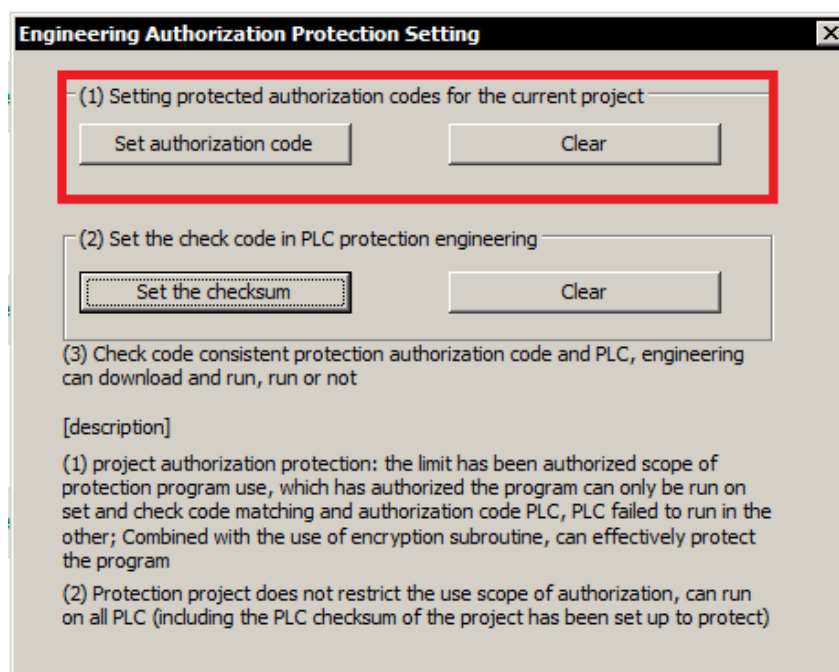
Subroutine encryption can encrypt and protect part of the code in the project, and can effectively protect the user's core algorithm and process;

Through the combination of project authorization protection and subprogram encryption, it can prevent the core algorithm and process from being stolen in the project, and can also prevent the project from being illegally used, and realize flexible and effective protection of user projects;

### 7.2.2 Project authorization protection

#### 1) Set authorization protection in the project

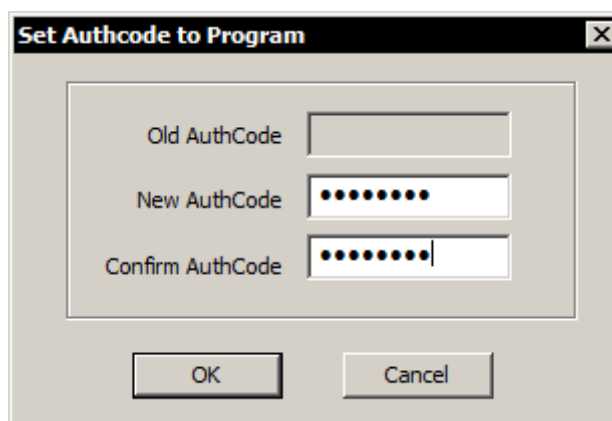
Click the menu item PLC -> Engineering Authorization Protection Settings to open the "Engineering Authorization Protection Settings" window, you can set the authorization protection code, change the authorization protection code, and clear the authorization protection code for the project, as shown in the figure below:



#### ▪ Set up authorization protection

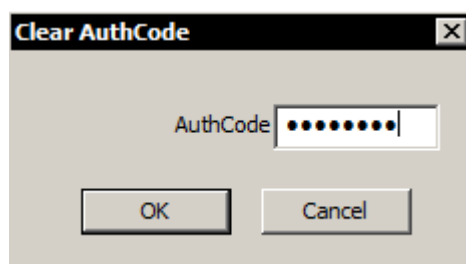
Click the "Set Authorization Code" button to open the authorization code setting window. If you set up authorization protection for the first time, you can directly enter the password setting; if the current project has set authorization protection, enter the old password (authorization protection code) and new password at the

same time (Authorization code), the old password is successfully checked and the new authorization code is set to the project; as shown in the figure below:



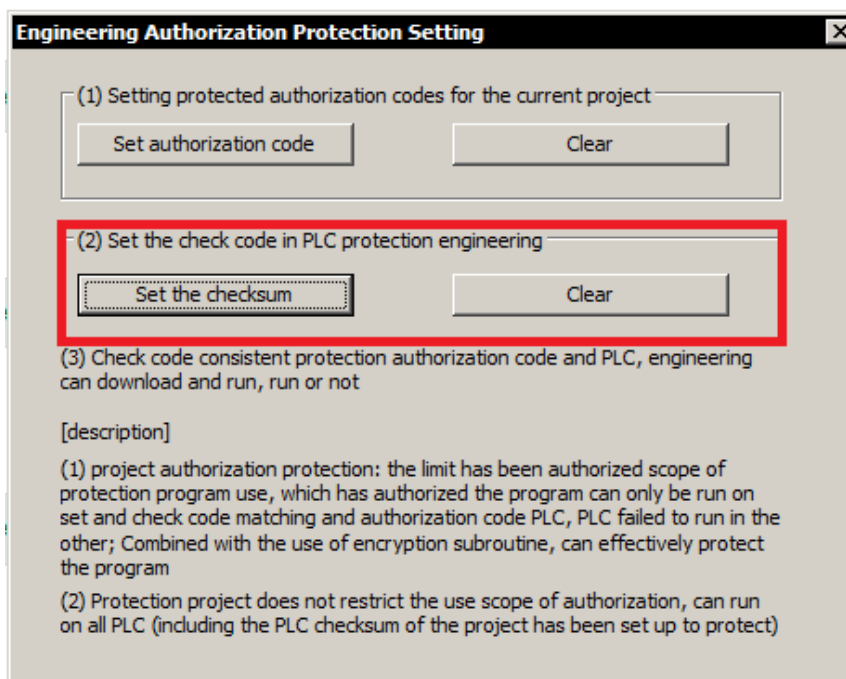
- Delete authorization

Click the "Clear" button to pop up the "Clear AuthCode" window, enter the authorization code and click "OK" to clear it, as shown in the following figure:



## 2) Set check code in PLC

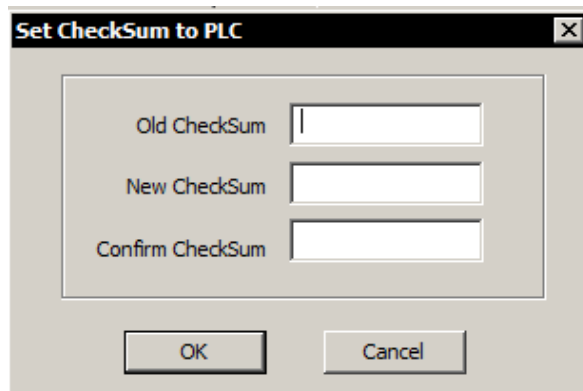
Click the menu item PLC -> Engineering Authorization Protection Settings to open the "Engineering Authorization Protection Settings" window, as shown below:





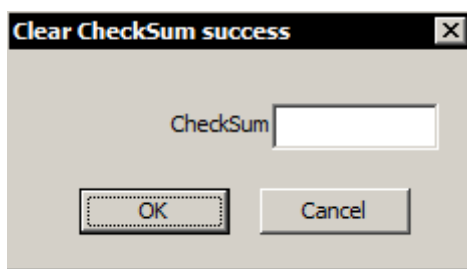
- Set check code to PLC

After connecting to the PLC, click the "Set the CheckSum" button to open the verification code setting window, enter the password and click "OK" to complete the setting; if the current PLC has set the verification code, you need to enter the old password and the new secret at the same time, The new password can be set to the PLC only after the old password is successfully verified; as shown in the figure below:



- Clear check code

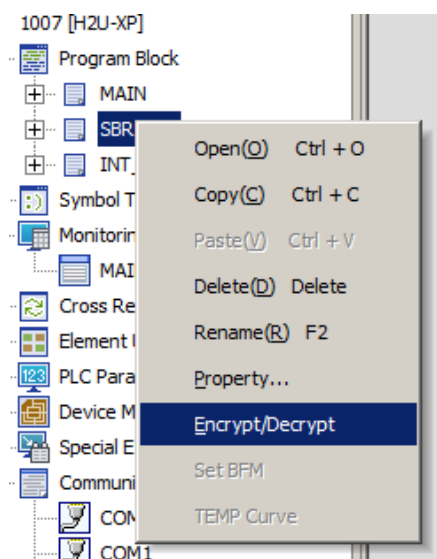
After clicking the "Clear" button, a window as shown in the figure below pops up, enter the password and clear the verification code after successful verification; as shown in the figure below:



## 7.3 Subroutine encryption/decryption

The function of subprogram encryption is to hide part of the content of the program, so that key content can be protected, and non-critical parts can be disclosed for other people to use.

Right-click the selected subroutine, and the following menu items will pop up:



Encryption: Select a common subroutine, and then execute the "Encrypt/Decrypt" menu, and a password input window will pop up. After entering the password you want to set, and then click "OK", AutoShop will encrypt the selected subroutine. The encrypted subroutine will display a lock icon, and the encrypted subroutine will automatically close and cannot be viewed.

Decryption: Double-click the encrypted program and a password window will pop up, and you can view it after entering the password correctly.

Remarks: The project authorization protection verification code set in the PLC cannot be cleared by "Clear PLC Program Storage Space", but can only be cleared by the "Clear" button in the "Project Authorization Protection Settings" window; The verification code can only be cleared after the PLC is stopped. It cannot be cleared if you forget the password.

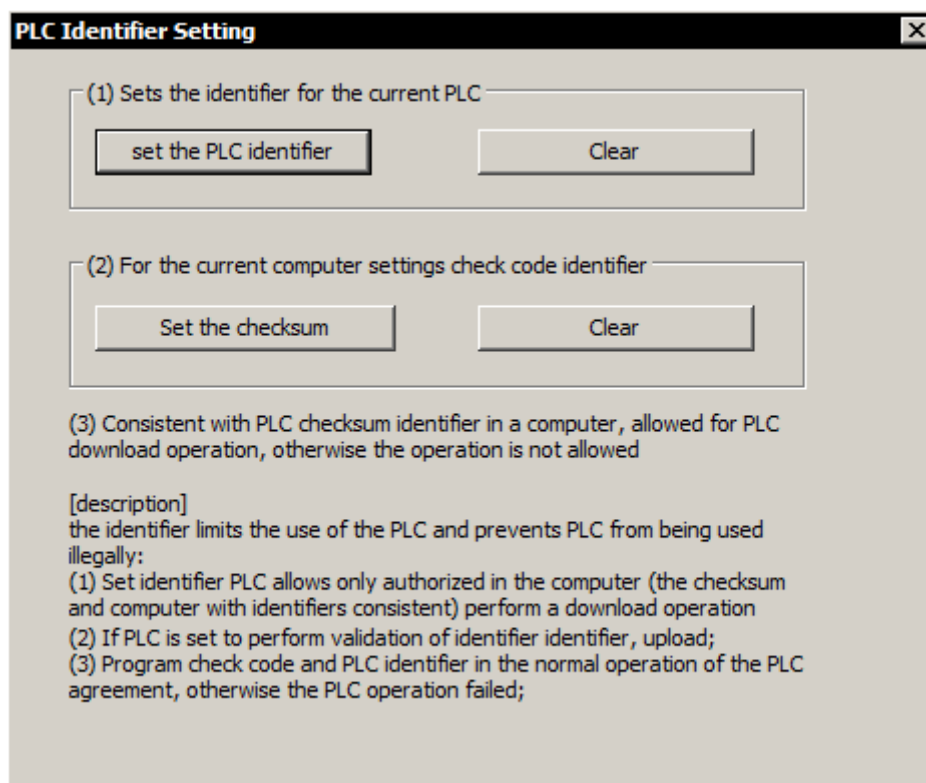
## 7.4 PLC identifier

The PLC identifier is a serial number used to mark the PLC, which is equivalent to adding a unique mark to the PLC. Its function is to help PLC users protect their customers: the PLC with the PLC identifier must match the program with the PLC identifier before the PLC can work normally. This plays two roles:

The PLC with the identifier can only run the program with the identifier;

The program with the identifier can only be run on the PLC with the identifier.

### 7.4.1 Setting interface of PLC identifier

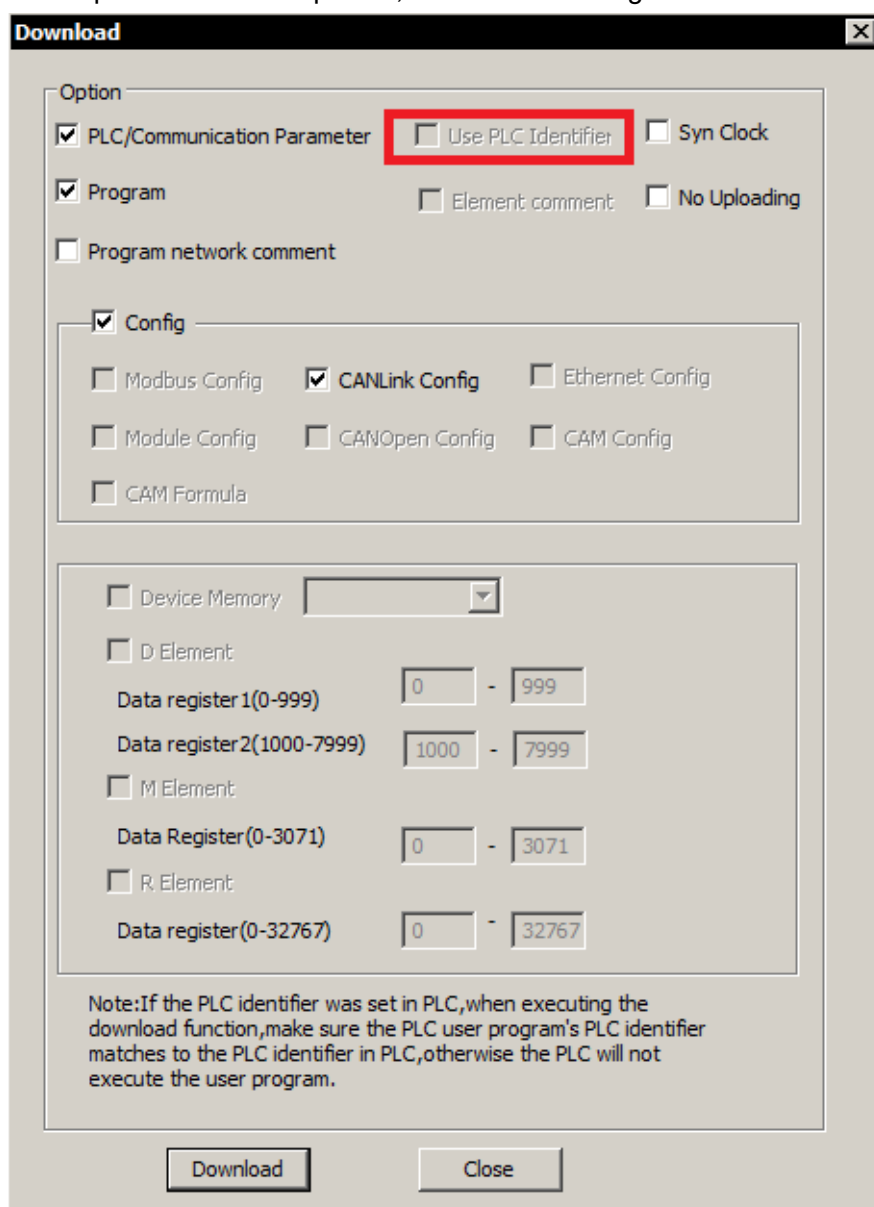


In the above figure, "(1) Sets the identifier for the current PLC" refers to the identifier stored in the PLC, "(2) For the current computer settings check code identifier" refers to the identifier stored in the current computer (using this Programs written by the computer's AutoShop will use this identifier). Only when the two match, the program can run normally. So usually the two are modified together to ensure that the PLC can operate normally.

The software also allows individual modification of "(1) Sets the identifier for the current PLC " or "(2) For the current computer settings check code identifier". When "(1) Sets the identifier for the current PLC " is modified separately, only the identifier saved in the PLC is modified. This operation can be performed when the identifier does not match, and the PLC must be connected. When "(2) For the current computer settings check code identifier" is individually modified, only the identifier saved in the current computer is modified.

Same as other passwords, the identifier has a length of 8 digits, which can be numbers, letters and all other input characters (such as +-\* / and other characters are also acceptable).

When the machine is set with the "identifier in the program", when downloading the program, the "Use PLC Identifier" in the download options becomes optional, as shown in the figure:



Before downloading, AutoShop will first check whether the PLC has set the identifier. If it has been set, it is judged whether it matches the PLC identifier in the current program. If it matches, the download will continue. If it does not match, the download cannot be performed; if the identifier has not been set, AutoShop will set the current identifier to the PLC.

The upload operation is similar. If the identifier in the PLC does not match the identifier in the current program, the upload cannot be performed.

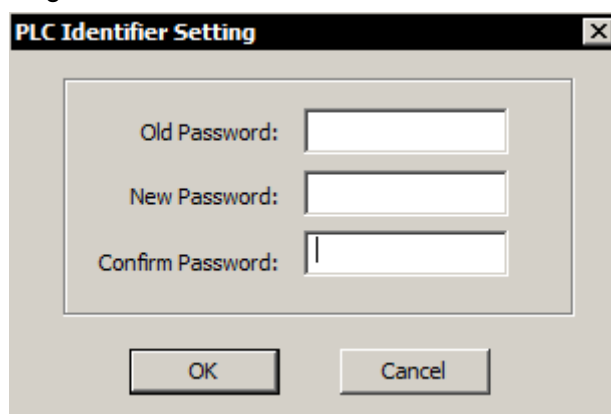
[Note]: Once the identifier in the PLC is set, it cannot be cleared, only modification is allowed. So if an identifier is set, please remember the password! If you forget it, you can only return it to Inovance for processing.

#### 7.4.2 Modification of PLC identifier

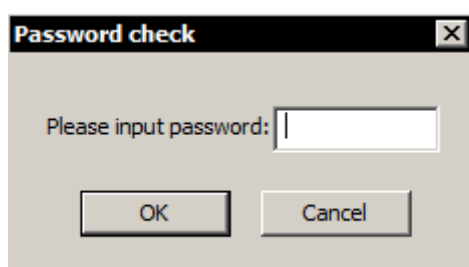
Set the PLC identifier, the PLC terminal identifier will automatically match the background save identifier when downloading the user program, and the download operation will continue if the match is successful; otherwise the download operation will be terminated;

- Check "Identifier in PLC"

Click the "Settings" button to pop up "PLC identifier settings", which can modify the PLC equipment and the "PLC identifiers" saved in the background;



- Click "Clear" to clear the "PLC identifier" saved in the PLC equipment and software



- Check "Identifier in Program"

The operation is the same as the "Identifier in PLC", and the corresponding operation is only effective for the "PLC Identifier" in the software.

## 7.5 Firmware Upgrade

### 7.5.1 Preparation

Autoshop version: version 3.02 and above;

PLC version: H2U-XP: 24136 and above; H1U-XP: 26127 and above.

Upgrade cable: mini USB cable (company material code 15041200)

There is no need to prepare 220VAC power supply, and the PLC can be powered directly from the computer USB port.

The firmware is an “.upgrade” file. Place the upgrade file required for the upgrade in the Autoshop installation directory: C(D):\Inovance Control\AutoShop\Upgrade

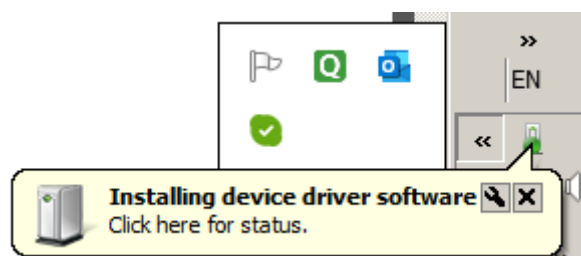
### 7.5.2 Upgrade operation



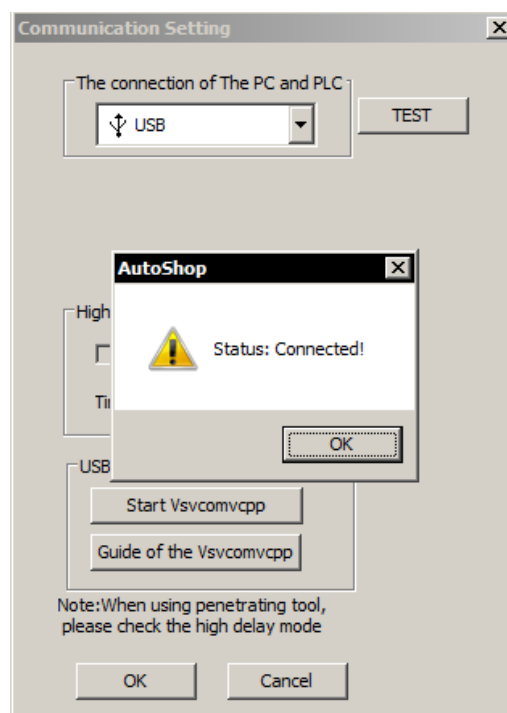
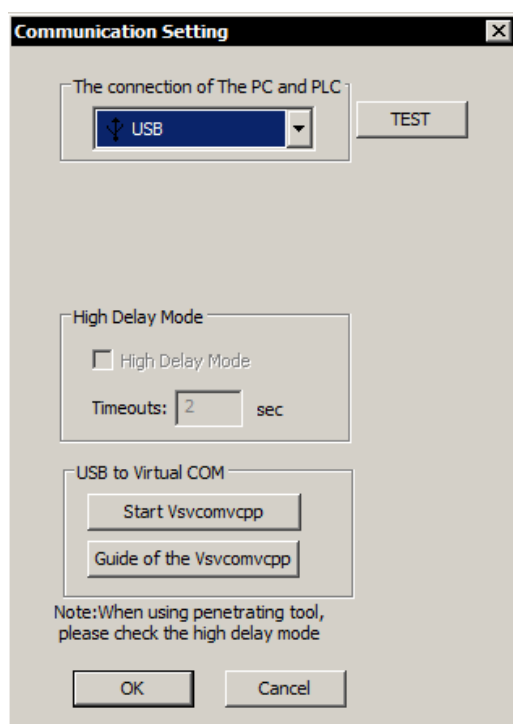
- 1) Open Inovance PLC programming software
- 2) Use a mini USB cable to connect the PLC that needs to firmware upgrade and connect the other end to the computer's USB port. The PLC is powered by the USB port and does not require 220VAC power supply.



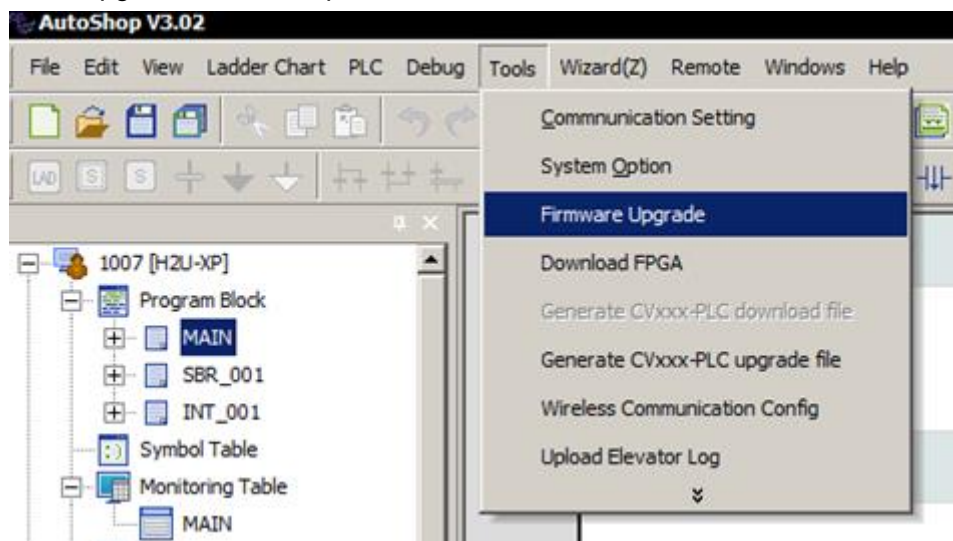
If the connection is successful, the icon will appear in the lower right corner of the monitor (confirm that the USB driver installation is successful); when the driver is not installed, the following hardware installation prompt will appear, just click to install the software automatically.



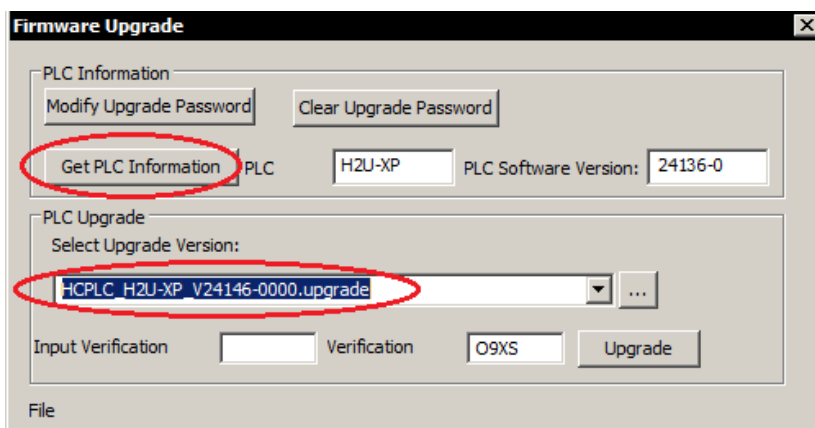
- 3) In Autoshop communication settings, select USB as the connection method between PC and PLC. You can click TEST to test the connection.



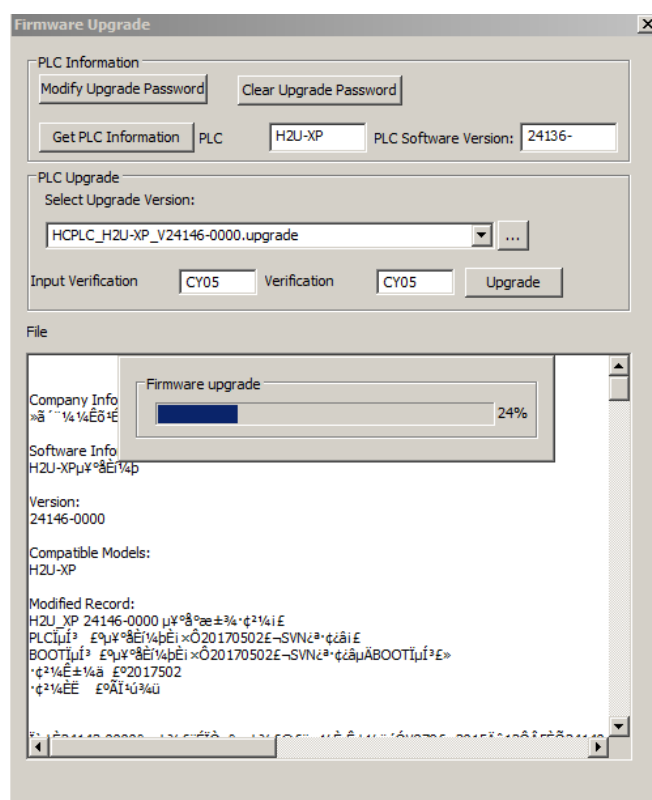
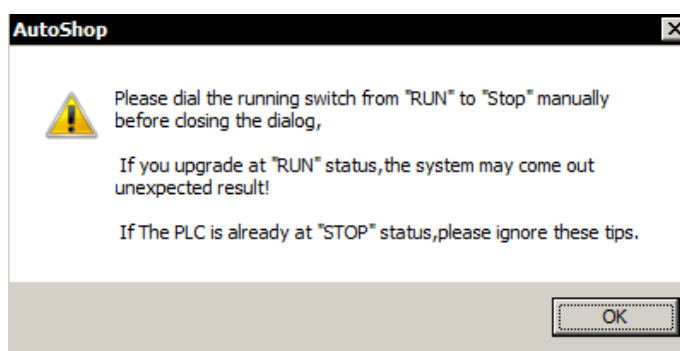
- 4) In the PLC programming software AutoShop interface, refer to the following prompts to find the Tools -> "Firmware Upgrade" function option



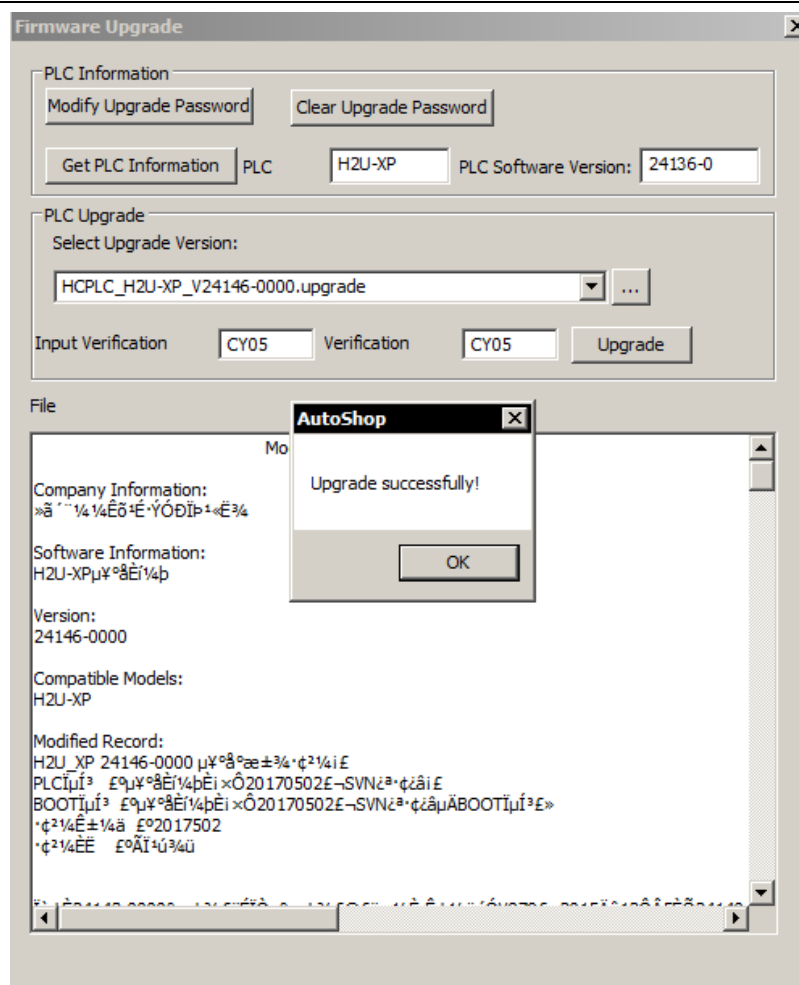
- 5) Confirm that the USB driver is installed successfully, as shown in the figure below, click to get the PLC version information; then the current PLC software version can be displayed; click the "Select the version to upgrade" drop-down arrow to select the firmware version to be upgraded.



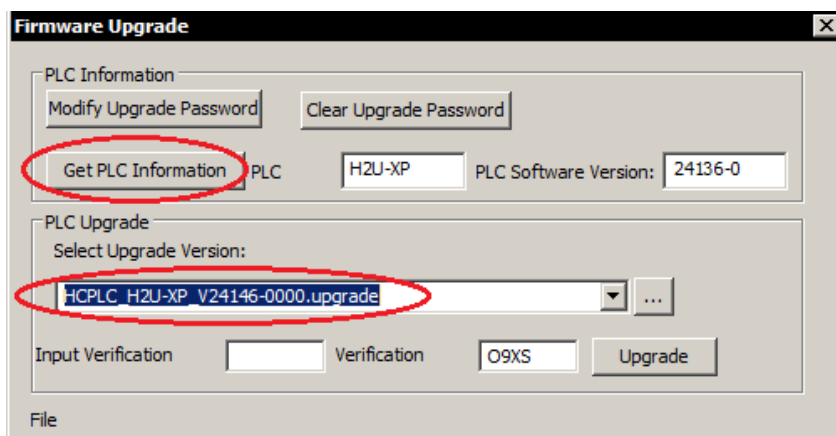
- 6) After entering the verification code, follow the prompts on the pop-up interface, you need to set the DIP switch to the "stop" state; click the "upgrade" button; the entire upgrade process takes a few seconds



- 7) The following prompts after the upgrade is successful



- 8) Obtain PLC information again, this time is the latest downloaded version number; software programming is complete.



- 9) Download operation

If the project has been authorized for protection, download the program to the PLC that has set the check code and is consistent with the authorization code in the project and can run normally, otherwise the run fails;

If the project is not authorized to protect, it can run on any PLC, even if the check code has been set on the PLC, it can run normally.





## Chapter 8 Appendix

### Content

8.1 System special soft components.....	- 470 -
8.2 System error recording function .....	- 488 -
8.2.1 Error code and content .....	- 488 -
8.2.2 PLC running time/abnormal record .....	- 494 -
8.2.3 Use Autosop software for fault diagnosis and handling .....	- 494 -
8.3 Error code register.....	- 496 -
8.4 Software difference between H2U series MTQ type and MT type.....	- 497 -
8.5 Supplementary Instructions for H2U Series MTP Software.....	- 499 -
8.5.1 Description of DRVI and DRVA special functions .....	- 499 -
8.5.2 Description of ZRN instruction.....	- 502 -
8.6 List of commands not supported by H2U series MTQ/MTP models.....	- 503 -
8.7 Description of enhanced functions of H2U series high-speed processing instructions.....	- 504 -
8.8 Interrelationship of program flow control instructions .....	- 510 -
8.9 Function description of some special relays and registers .....	- 513 -

## Chapter 8 Appendix

### 8.1 System special soft components

M8000~M8511, D8000~D8511 are defined as special component types, and their functions are described in the table below.

M Element	Description	D Element	Description
System operating status			
M8000	ON during operation of the user program	D8000	Watchdog timer for user program operation
M8001	Inverted M8000 state	D8001	Board program version, for example, 24100 H2U = 24, 100 = V1.00
M8002	ON in the first operation cycle of the user program	D8002	Program capacity, for example, 4 KB, 8 KB, and 16 KB
M8003	Inverted M8002 state	D8003	0x10, stored in the Programmable Logic Controller (PLC)
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	D8005	Current battery voltage BCD value
M8006	Action upon low battery voltage (latch)	D8006	Detected battery undervoltage value, 2.6 V by default
M8007	M8007 & M8008 act after 5 ms of AC de-energization; 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	When the de-energization time value is not greater than the D8008 value and M8008 is switched off, the user program is disabled. M8000 is OFF.	D8008	AC de-energization detection time, 10 ms by default
M8009	Action upon a power failure (24 V) encountered by the expansion unit	D8009	Module number of the de-energized 24 V expansion unit
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	Installation detection	D8018	RTC calendar year (2000 to 2099)
M8019	RTC error	D8019	RTC week
Instruction flag			
M8020	Zero flag	D8020	X000-X007 input filter constant: 0 to 60 (10 ms by default)
M8021	Borrow flag	D8021	Reserved
M8022	Carry flag	D8022	Reserved
M8023	Reserved	D8023	Reserved
M8024	Direction of the BMOV instruction	D8024	Reserved
M8025	HSC instruction mode	D8025	Reserved
M8026	RAMP instruction mode	D8026	Reserved
M8027	PR mode	D8027	Reserved
M8028	Reserved	D8028	Same variable address as Z0
M8029	Part of instructions (for example, the PLSR instruction) executed	D8029	Same variable address as V0
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	DRVI, DRVA bias operating speed, M8135 determines whether it is valid [Y0] [default 100]
M8035	Command for forced running 1	D8035	DRVI, DRVA bias operating speed, M8135 determines whether it is valid [Y1] [default 100]
M8036	Command for forced running 2	D8036	DRVI, DRVA bias operating speed, M8135 determines whether it is valid [Y2] [default 100]
M8037	Command for forced stop	D8037	Reserved
M8038	Communication parameter setting flag	D8038	Reserved
M8039	Constant scan control	D8039	Constant scan time (ms), 0 ms by default
Step ladder			
M8040	Transfer disabling	D8040	Save the minimum action
M8041	Transfer start	D8041	
M8042	Pulse output corresponding to startup input	D8042	

M8043	Zero return end flag	D8043	address number of S0~S899 in D8040, and the other order, the largest address number in D8047.		
M8044	Mechanical origin action detected	D8044			
M8045	Output reset disabling	D8045			
M8046	After M8047 acts, if any of S0 to S999 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 I00X interrupt disabling	D8050	The maximum operating speed of DRVI, DRVA [default 100000] is determined by M8135 whether it is valid or not [Y0]		
M8051	Driver I10X interrupt disabling	D8051			
M8052	Driver I20X interrupt disabling	D8052	The maximum operating speed of DRVI, DRVA [default 100000] is determined by M8136 whether it is valid or not [Y1]		
M8053	Driver I30X interrupt disabling	D8053			
M8054	Driver I40X interrupt disabling	D8054	The maximum operating speed of DRVI, DRVA [default 100000] is determined by M8137 whether it is valid or not [Y2]		
M8055	Driver I50X interrupt disabling	D8055			
M8056	Driver I6XX interrupt disabling	D8056	Reserved		
M8057	Driver I7XX interrupt disabling	D8057	Reserved		
M8058	Driver I8XX 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	Head I/O address number of I/O composition error
M8061	PLC hardware error	Blinking	STOP	D8061	PLC hardware error code number
M8062	PLC communication error	OFF	RUN	D8062	PLC communication error code
M8063	Connection error/general communication error	OFF	RUN	D8063	Parallel connection 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	Reserved			D8069	M8065-M8067 error step numbers
Connection					
M8070	Connected master station driver			D8070	Duration of incorrect parallel connection: 500 ms
M8071	Connected slave station driver			D8071	Reserved
M8072	ON during parallel connection			D8072	Reserved
M8073	Incorrect M8070/M8071 setting for parallel connection			D8073	Reserved
Sampling tracking					
M8074	Reserved			D8074	Residual opportunities of sampling
M8075	Sampling tracking start instruction			D8075	Sampling count setting (1 to 512)
M8076	Start instruction executed after sampling tracking is prepared			D8076	Sampling cycle
M8077	Monitoring during sampling tracking			D8077	Trigger specifying
M8078	Monitoring after sampling tracking is completed			D8078	Setting address numbers of elements as trigger conditions
M8079	Sampling tracking for values greater than the D8075 value			D8079	Sampling data pointer
M8080	Reserved			D8080	Bit element address number: 0
M8081	Reserved			D8081	Bit element address number: 1
M8082	Reserved			D8082	Reserved
M8083	Reserved			D8083	Reserved
M8084	Enabling multiple high-speed counter interrupts (OFF by default)			D8084	Counter sequence number of high-speed counter multi-segment interrupt (not available for XP type)
M8085	Y0 port output initialization flag			D8085	The default data of multi-segment interrupt is 0 (not available for XP type)
M8086	Y1 port output initialization flag			D8086	Corresponding D component serial number (not available for XP type)
M8087	Y2 port output initialization flag			D8087	Reserved
M8088	Reserved			D8088	Reserved
M8089	Reserved			D8089	Reserved
M8090	Y0 output complete interrupt enabling			D8090	Reserved
M8091	Y1 output complete interrupt enabling			D8091	Reserved
M8092	Y2 output complete interrupt enabling			D8092	Reserved
M8093	Reserved			D8093	Reserved
M8094	Reserved			D8094	Reserved
M8095	Y0 uses the PWM function, and the output pulse unit is changed from ms to us.			D8095	Reserved

M8096	Y1 uses the PWM function, and the output pulse unit is changed from ms to us.	D8096	Word element address number: 0
M8097	Y2 uses the PWM function, and the output pulse unit is changed from ms to us.	D8097	Word element address number: 1
M8098	Reserved	D8098	Word element address number: 2
<b>High-speed ring counter</b>			
M8099	High-speed ring counter started	D8099	(0 to 32,767) rising action ring counter (0.1 ms)
<b>Other functions</b>			
M8100	SPD (X000) - number of pulses/minute	D8100	Reserved
M8101	SPD (X001) - number of pulses/minute	D8101	Board program version, for example, 24100 H2U = 24, 100 = V1.00
M8102	SPD (X002) - number of pulses/minute	D8102	User program capacity provided by the system
M8103	SPD (X003) - number of pulses/minute	D8103	Reserved
M8104	SPD (X004) - number of pulses/minute	D8104	Acceleration time (100 by default) during execution of DRVI and DRVA instructions, the validity of which is determined by M8135 (Y0)
M8105	SPD (X005) - number of pulses/minute	D8105	Acceleration time (100 by default) during execution of DRVI and DRVA instructions, the validity of which is determined by M8135 (Y1)
M8106	Reserved	D8106	Acceleration time (100 by default) during execution of DRVI and DRVA instructions, the validity of which is determined by M8135 (Y2)
M8107	Reserved	D8107	Reserved
M8108	Reserved	D8108	Reserved
M8109	Output refreshing error	D8109	Output address number of output refreshing error
<b>COM0 communication and connection</b>			
M8110	Reserved	D8110	Communication format, set on the user interface (UI), 0 by default
M8111	Sending (RS instruction)	D8111	Station number, set on the UI, 1 by default
M8112	Flag of sending (RS instruction) and execution state (MODBUS)	D8112	Volume of residual data transmitted (for RS instruction only)
M8113	Flag of receipt (RS instruction) and communication error flag (MODBUS)	D8113	Volume of received data (for RS instruction only)

M8114	Receiving (for RS instruction only)	D8114	Start of Text (STX) (for RS instruction only)
M8115	Reserved	D8115	End of Text (ETX) (for RS instruction only)
M8116	Reserved	D8116	Communication protocol, set on the UI, 0 by default
M8117	Reserved	D8117	PC link protocol - head data address required
M8118	Reserved	D8118	PC link protocol - volume of sent data required
M8119	Timeout criterion	D8119	Communication timeout criterion, set on the UI, 10 (100 ms) by default
<b>COM1 communication and connection</b>			
M8120	Reserved	D8120	Communication format, set on the user interface (UI), 0 by default
M8121	Sending (RS instruction)	D8121	Station number, set on the UI, 1 by default
M8122	Flag of sending (RS instruction) and execution state (MODBUS)	D8122	Volume of residual data transmitted (for RS instruction only)
M8123	Flag of receipt (RS instruction) and communication error flag (MODBUS)	D8123	Volume of received data (for RS instruction only)
M8124	Receiving (for RS instruction only)	D8124	STX (for RS instruction only)
M8125	Reserved	D8125	End of Text (ETX) (for RS instruction only)
M8126	485BD expansion card is valid when ON	D8126	Communication protocol, set on the UI, 0 by default
M8127	Reserved	D8127	PC link protocol - head data address required SETEX protocol: The starting address of the buffer zone sent from the slave station (PLC) to the master station (display screen), the buffer size is 18.
M8128	Reserved	D8128	PC link protocol - volume of sent data required SETEX protocol: The master station (display screen) sends the starting address of the buffer zone to the slave station (PLC), the buffer size is 19.
M8129	Timeout criterion	D8129	Communication timeout criterion, set on the UI, 10 (100 ms) by default
<b>High-speed positioning</b>			
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	
M8134	Reserved	D8134	Use of HSZ and PLSY speed model comparison pulses
M8135	Y0 deceleration time (ON-PLSR, DRVI, and DRVA) supported	D8135	
M8136	Y1 deceleration time (ON-PLSR, DRVI, and DRVA) supported	D8136	Total of Y000 and Y001 output pulses
M8137	Y2 deceleration time (ON-PLSR, DRVI, and DRVA) supported	D8137	
M8138	Reserved	D8138	Reserved
M8139	Reserved	D8139	Reserved
M8140	ZRN CLR signal output function enabled	D8140	Accumulated number of pulses corresponding to Y000 output by executing PLSY and PLSR instructions
M8141	X0~X5 interrupt stop Y0 pulse output enable	D8141	
M8142	X0~X5 interrupt stop Y1 pulse output enable	D8142	Accumulated number of pulses corresponding to Y001 output by executing PLSY and PLSR instructions
M8143	X0~X5 interrupt stop Y2 pulse output enable	D8143	
M8144	Reserved	D8144	Reserved
M8145	Y000 pulse output stop	D8145	Offset speed during execution of DRVI and DRVA instructions [default 100]
M8146	Y001 pulse output stop	D8146	Maximum speed during execution of DRVI and DRVA instructions (100,000 by default)
M8147	Y000 pulse output monitoring	D8147	
M8148	Y001 pulse output monitoring	D8148	Acceleration/deceleration time during execution of DRVI and DRVA instructions (100 by default)
M8149	Y002 pulse output monitoring	D8149	Reserved
M8150	Reserved	D8150	Accumulated number of pulses corresponding to Y002 output by executing PLSY and PLSR instructions
M8151	Reserved	D8151	
M8152	Y002 pulse output stop	D8152	Accumulated number of pulses corresponding to Y003 output by executing PLSY and PLSR instructions
M8153	Reserved	D8153	
M8154	Reserved	D8154	Accumulated number of pulses corresponding to Y004 output by executing PLSY and PLSR instructions
M8155	Reserved	D8155	



M8156	Reserved	D8156	Definition of Y0 clearing signals (ZRN) (by default, 5 = Y005)
M8157	Reserved	D8157	Definition of Y1 clearing signals (ZRN) (by default, 6 = Y006)
Expansion function			
M8158	Reserved	D8158	Definition of Y2 clearing signals (ZRN) (by default, 7 = Y007)
M8159	Reserved	D8159	Reserved
M8160	SWAP function of XCH	D8160	Reserved
M8161	ASC/RS/ASCII/HEX/CCD processing mode	D8161	Reserved
M8162	High-speed parallel connection mode	D8162	Reserved
M8163	Reserved	D8163	Reserved
M8164	(FROM/TO) variable number of transmitted points	D8164	(FROM/TO) specified number of transmitted points
M8165	Reserved	D8165	Deceleration time (100 by default) during execution of PLSR, DRVI and DRVA instructions, the validity of which is determined by M8135 (Y0)
M8166	Reserved	D8166	Deceleration time (100 by default) during execution of PLSR, DRVI and DRVA instructions, the validity of which is determined by M8136 (Y1)
M8167	(HEY) HEX data processing function	D8167	Deceleration time (100 by default) during execution of PLSR, DRVI and DRVA instructions, the validity of which is determined by M8137 (Y2)
M8168	(SMOV) HEX data processing function	D8168	Reserved
M8169	Reserved	D8169	Reserved
Pulse Capture			Communication and connection
M8170	X000 pulse capture	D8170	Reserved
M8171	X001 pulse capture	D8171	Reserved
M8172	X002 pulse capture	D8172	Reserved
M8173	X003 pulse capture	D8173	Station number state
M8174	X004 pulse capture	D8174	Communication substation state
M8175	X005 pulse capture	D8175	Refresh range state
M8176	Reserved	D8176	Station number setting
M8177	Reserved	D8177	Setting of number of communication substations

M8178	Reserved	D8178	Refresh range setting
M8179	Reserved	D8179	Setting of number of retries
M8180	Reserved	D8180	Communication timeout setting
Communication and connection			Address indexing
M8181	Reserved	D8181	Reserved
M8182	Reserved	D8182	Bit element address number: 2; Z1 register content
M8183	Data transmission master station error	D8183	Bit element address number: 3; V1 register content
M8184	Data transmission slave station 1 error	D8184	Bit element address number: 4; Z2 register content
M8185	Data transmission slave station 2 error	D8185	Bit element address number: 5; V2 register content
M8186	Data transmission slave station 3 error	D8186	Bit element address number: 6; Z3 register content
M8187	Data transmission slave station 4 error	D8187	Bit element address number: 7; V3 register content
M8188	Data transmission slave station 5 error	D8188	Bit element address number: 8; Z4 register content
M8189	Data transmission slave station 6 error	D8189	Bit element address number: 9; V4 register content
M8190	Data transmission slave station 7 error	D8190	Bit element address number: 10; Z5 register content
M8191	Data - transmitting	D8191	Bit element address number: 11; V5 register content
M8192	CAN receive timeout [N series]	D8192	Bit element address number: 12; Z6 register content
M8193	Free CAN command receiving status [N series]	D8193	Bit element address number: 13; V6 register content
M8194	CAN free command transmission failed [N series]	D8194	Bit element address number: 14; Z7 register content
M8195	C251 multiplication control	D8195	Bit element address number: 15; V7 register content
M8196	C252 multiplication control	D8196	Reserved
M8197	C253 multiplication control	D8197	Reserved
M8198	C254 multiplication control	D8198	This machine is the identification of CANlink remote device, H2U is 10224, H1U is 10226
M8199	C255 multiplication control	D8199 Note 2	Expansion card type number, H2U is automatically recognized, D8199 displays the card type number; H1U cannot be automatically recognized, and the expansion card type needs to be set through D8199. The types of expansion cards are as follows: 1 RS232 expansion card

			2 RS422/485 communication expansion card 3 AD expansion card 4 DA expansion card 5 6A/6B expansion card 6 High-speed expansion card 7 CAN expansion card
Counter increment/decrement or status			Communication and connection
M8200	C200 control	D8200	Inovance software version number XXX.YY, XXX: non-standard software number, YY: software version number. Represents the temporary version number in hexadecimal
M8201	C201 control	D8201	Current connection scan time
M8202	C202 control	D8202	Maximum connection time
M8203	C203 control	D8203	Number of master station communication errors
M8204	C204 control	D8204	Number of slave station 1 communication errors
M8205	C205 control	D8205	Number of slave station 2 communication errors
M8206	C206 control	D8206	Number of slave station 3 communication errors
M8207	C207 control	D8207	Number of slave station 4 communication errors
M8208	C208 control	D8208	Number of slave station 5 communication errors
M8209	C209 control	D8209	Number of slave station 6 communication errors
M8210	C210 control	D8210	Number of slave station 7 communication errors
M8211	C211 control	D8211	Master station communication error code
M8212	C212 control	D8212	Slave station 1 communication error code
M8213	C213 control	D8213	Slave station 2 communication error code
M8214	C214 control	D8214	Slave station 3 communication error code
M8215	C215 control	D8215	Slave station 4 communication error code
M8216	C216 control	D8216	Slave station 5 communication error code
M8217	C217 control	D8217	Slave station 6 communication error code
M8218	C218 control	D8218	Slave station 7 communication error code
M8219	C219 control	D8219	Reserved

M8220	C220 control	D8220	Expansion card sending buffer
M8221	C221 control	D8221	Expansion card sending buffer
M8222	C222 control	D8222	Expansion card sending buffer
M8223	C223 control	D8223	Expansion card sending buffer
M8224	C224 control	D8224	Expansion card sending buffer
M8225	C225 control	D8225	Expansion card sending buffer
M8226	C226 control	D8226	Expansion card sending buffer
M8227	C227 control	D8227	Expansion card sending buffer
M8228	C228 control	D8228	Expansion card sending buffer
M8229	C229 control	D8229	Expansion card sending buffer
M8230	C230 control	D8230	Expansion card sending buffer
M8231	C231 control	D8231	Expansion card sending buffer
M8232	C232 control	D8232	Expansion card sending buffer
M8233	C233 control	D8233	Expansion card sending buffer
M8234	C234 control	D8234	Expansion card sending buffer
M8235	C235 control	D8235	Expansion card sending buffer
M8236	C236 control	D8236	Expansion card sending buffer
M8237	C237 control	D8237	Expansion card sending buffer
M8238	C238 control	D8238	Expansion card sending buffer
M8239	C239 control	D8239	Expansion card communication error counter
M8240	C240 control	D8240	CAN function setting, please refer to CAN communication description [N series]
M8241	C241 control	D8241	CAN receiving timeout setting (ms) [N series]
M8242	C242 control	D8242	CANlink address setting/display address [N series]
M8243	C243 control	D8243	CAN baud rate auxiliary setting, or display the baud rate set by dial code [N series]
M8244	C244 control	D8244	CAN baud rate setting [N series]
M8245	C245 control	D8245	Set CANlink network device information save start register [N series]
M8246	C246 control	D8246	CAN command and status [N series]
M8247	C247 control	D8247	Synchronous clock counter L[N series]
M8248	C248 control	D8248	Synchronous clock counter H[N series]
M8249	C249 control	D8249	Number of network equipment [N series]
M8250	C250 control	D8250	CAN interrupt error [N series]
M8251	C251 control	D8251	Data length received by CAN free command (MCFL) [N series]
M8252	C252 control	D8252	Data received by CAN free

			command MDLL [N series]
M8253	C253 control	D8253	Data received by CAN free command MDLH [N series]
M8254	C254 control	D8254	Data received by CAN free command MDHL [N series]
M8255	C255 control	D8255	Data received by CAN free command MDHH [N series]
M8256	Reserved	D8256	Reserved
M8257	Reserved	D8257	Reserved
M8258	Reserved	D8258	Reserved
M8259	Reserved	D8259	Reserved
<b>COM2 communication and connection</b>			
M8260	Reserved	D8260	Communication format, set on the user interface (UI), 0 by default
M8261	Sending (RS2 instruction)	D8261	Station number, set on the UI, 1 by default
M8262	Flag of sending (RS2 instruction) and execution state (MODBUS2)	D8262	Volume of residual data transmitted (for RS2 instruction only)
M8263	Flag of receipt (RS2 instruction) and communication error flag (MODBUS2)	D8263	Volume of received data (for RS2 instruction only)
M8264	Receiving (for RS2 instruction only)	D8264	STX (for RS2 instruction only)
M8265	Reserved	D8265	End of Text (ETX) (for RS2 instruction only)
M8266	485BD expansion card is valid when ON	D8266	Communication protocol, set on the UI, 0 by default
M8267	Reserved	D8267	PC link protocol - head data address required SETEX protocol: The starting address of the buffer zone sent from the slave station (PLC) to the master station (display screen), the buffer size is 18.
M8268	Reserved	D8268	PC link protocol - volume of sent data required SETEX protocol: The master station (display screen) sends the starting address of the buffer zone to the slave station (PLC), the buffer size is 19.
M8269	Timeout criterion	D8269	Communication timeout criterion, set on the UI, 10 (100 ms) by default
<b>COM3 communication and connection</b>			
M8270	Reserved	D8270	Communication format, set on the user interface (UI), 0 by default
M8271	Sending (RS2 instruction)	D8271	Station number, set on the UI, 1 by default

M8272	Flag of sending (RS2 instruction) and execution state (MODBUS2)	D8272	Volume of residual data transmitted (for RS2 instruction only)
M8273	Flag of receipt (RS2 instruction) and communication error flag (MODBUS2)	D8273	Volume of received data (for RS2 instruction only)
M8274	Receiving (for RS2 instruction only)	D8274	STX (for RS2 instruction only)
M8275	Reserved	D8275	End of Text (ETX) (for RS2 instruction only)
M8276	485BD expansion card is valid when ON	D8276	Communication protocol, set on the UI, 0 by default
M8277	Reserved	D8277	PC link protocol - head data address required SETEX protocol: The starting address of the buffer zone sent from the slave station (PLC) to the master station (display screen), the buffer size is 18.
M8278	Reserved	D8278	PC link protocol - volume of sent data required SETEX protocol: The master station (display screen) sends the starting address of the buffer zone to the slave station (PLC), the buffer size is 19.
M8279	Timeout criterion	D8279	Communication timeout criterion, set on the UI, 10 (100 ms) by default
CAN communication			
M8280	CAN protocol flag	D8280	Current software CAN protocol version number (default 300)
M8281	The CAN extension card is not recognized, the CAN-LINK function cannot be used	D8281	Expansion card type number, H2U is automatically recognition, D8280 displays the card type number; H1U cannot be automatically recognized, and the expansion card type needs to be set through D8281. The types of expansion cards are as follows: 1 RS232 expansion card 2 RS422/485 communication expansion card 3 AD expansion card 4 DA expansion card 5 6A/6B/3A expansion card 6 High-speed expansion card 7 CAN expansion card
M8282	M8282=0: CANLink protocol; M8282=1: CAN free protocol;	D8282	Reserved

M8283	Reserved	D8283	Reserved
M8284	0: Dial the code to set the CAN_LINK address, D8284 displays the address 1: D8284 set CAN_LINK address	D8284	CAN_LINK address setting/display address
M8285	0: DIP switch sets the baud rate, D8285 displays the baud rate 1: M8286 determines the baud rate setting	D8285	Baud rate setting, or display the baud rate set by the DIP switch and gear
M8286	0: The baud rate is set in bins, which is set by D8286, and D8285 displays the baud rate 1: The baud rate is freely set by the 32-bit register composed of D8285 and D8286	D8286	When M8285=1, M8286=0 When D8286 <10 D8286 = other value, 800K D8286 = 6, 600K D8286 = 5, 500K D8286 = 4, 400K D8286 = 3, 300K D8286 = 2, 200K D8286 = 1, 100K D8286 = 0, 10K When D8286 >=10 D8286 = 10, 10K D8286 = 20, 20K D8286 = 50, 50K D8286 = 125, 125K D8286 = 250, 250K
M8287	Allow saving network device information	D8287	M8287=1, set the initial D component number to save the network device information.
M8288	Reserved	D8288	Reserved
M8289	Reconfigure CAN-LINK network D8289 reserved	D8289	Reserved
M8290	Network start-stop control components	D8290	Reserved
M8291	Synchronous transmission trigger element	D8291	Number of network devices
M8292	0: Free CAN command reception completed or idle 1: Free CAN command receiving status	D8292	CANRX received data length (MCFL)
M8293	CAN free command transmission failed	D8293	CAN interrupt register
M8294	CAN free command received successfully	D8294	CAN status register
M8295	CAN status is abnormal	D8295	CANRX receive data MDL low 16 bits
M8296	Device address error	D8296	CANRX receive data MDL high 16 bits
M8297	Reserved	D8297	CANRX receive data MDH low 16 bits
M8298	M8298=0: CANTX is idle; M8298=1: CANTX is sending data;	D8298	CANRX receive data MDH high 16 bits
M8299	CAN receive timeout	D8299	CAN receiving timeout setting (ms)
M8300	Device communication is interrupted	D8300	Device ID
M8301	Network overload	D8301	Fault information of this station

			(PLC fault code:)
M8302	The device sends illegal data (non-CAN-LINK protocol data)	D8302	Device ID
M8303	The device did not execute the configuration information correctly	D8303	Device ID
M8304	Protocol version error	D8304	Device ID
M8305	There is a device alarm in the network (user program reset)	D8305	Network alarm information
M8306	Enable detailed alarm logging	D8306	If detailed alarm record is enabled: Set the initial D component number for saving alarm/error.
M8307	Reserved	D8307	
M8308	Reserved	D8308	
M8309	Reserved	D8309	
M8310	N: N protocol data transmission slave station 8 error	D8310	
M8311	N: N protocol data transmission slave station 9 error	D8311	
M8312	N: N protocol data transmission slave station 10 error	D8312	
M8313	N: N protocol data transmission slave station 11 error	D8313	
M8314	N: N protocol data transmission slave station 12 error	D8314	
M8315	N: N protocol data transmission slave station 13 error	D8315	
M8316	N: N protocol data transmission slave station 14 error	D8316	
M8317	N: N protocol data transmission slave station 15 error	D8317	
M8318	Reserved	D8318	
M8319	Reserved	D8319	
M8320~M8419	Reserved	D8320~D8419	Reserved
<b>High-speed input and output</b>			
M8420	Reserved	D8420	
M8421	Reserved	D8421	
M8422	Reserved	D8422	
M8423	Reserved	D8423	
M8424	Reserved	D8424	
M8425	Reserved	D8425	
M8426	Reserved	D8426	
M8427	Reserved	D8427	
M8428		D8428	
M8429	DSZR instruction execution abnormal end flag	D8429	
M8430		D8430	3G module and PLC heartbeat maintenance (write a fixed value to the register every 10s to realize the anti-disassembly)



			function) ---Cancelled
M8431		D8431	3G module and server heartbeat maintenance
M8432		D8432	Signal strength in 3G mode. When the signal strength is lower than 10, the alarm light is always on, the signal strength
M8433		D8433	When it is 0 (not collected at a certain time), it will only be recorded in the log but not written to Autoshop,
M8434		D8434	View D8432 should display the last value, cannot write 0
M8435		D8435	When the login registration code is wrong or repeated, the value returned by the server is written
M8436		D8436	Ethernet IP address
M8437		D8437	
M8438		D8438	
M8439		D8439	
DSZR instruction Y0 pulse output port			
M8440	Monitoring during pulse output	D8440	Reserved
M8441	Clear signal output function is valid	D8441	
M8442	Origin return direction designation	D8442	Base speed (HZ) [default 200]
M8443	Forward limit	D8443	Maximum speed (HZ) (32 bits) [default 100000]
M8444	Reversal limit	D8444	
M8445	Near-point signal logic inversion	D8445	Creeping speed (HZ) [default 1000]
M8446	Zero signal logic inversion	D8446	Origin return speed (HZ) (32 bits) [default 50000]
M8447	Reserved	D8447	
M8448	Positioning command driving	D8448	Acceleration time (ms) [default 100]
M8449	Pulse stop command	D8449	Deceleration time (ms) [default 100]
DSZR instruction Y1 pulse output port			
M8450	Monitoring during pulse output	D8450	Reserved
M8451	Clear signal output function is valid	D8451	
M8452	Origin return direction designation	D8452	Base speed (HZ) [default 200]
M8453	Forward limit	D8453	Maximum speed (HZ) (32 bits) [default 100000]
M8454	Reversal limit	D8454	
M8455	Near-point signal logic inversion	D8455	Creeping speed (HZ) [default 1000]
M8456	Zero signal logic inversion	D8456	Origin return speed (HZ) (32 bits) [default 50000]
M8457	Reserved	D8457	
M8458	Positioning command driving	D8458	Acceleration time (ms) [default 100]
M8459	Pulse stop command	D8459	Deceleration time (ms) [default 100]
DSZR instruction Y2 pulse output port			

M8460	Monitoring during pulse output	D8460	Reserved
M8461	Clear signal output function is valid	D8461	
M8462	Origin return direction designation	D8462	Base speed (HZ) [default 200]
M8463	Forward limit	D8463	Maximum speed (HZ) (32 bits) [default 100000]
M8464	Reversal limit	D8464	
M8465	Near-point signal logic inversion	D8465	Creeping speed (HZ) [default 1000]
M8466	Zero signal logic inversion	D8466	Origin return speed (HZ) (32 bits) [default 50000]
M8467	Reserved	D8467	
M8468	Positioning command driving	D8468	Acceleration time (ms) [default 100]
M8469	Pulse stop command	D8469	Deceleration time (ms) [default 100]
M8470		D8470	
M8471		D8471	
M8472		D8472	
M8473		D8473	
M8474		D8474	
M8475		D8475	
M8476		D8476	
M8477		D8477	
M8478		D8478	
M8479		D8479	
M8480		D8480	
M8481		D8481	
M8482		D8482	
M8483		D8483	
M8484		D8484	
M8485		D8485	
M8486		D8486	
M8487		D8487	
M8488			
M8489			
M8490	MEAN instruction sum flag	D8490	Time when error 1 occurred [hours]
M8491	N: N protocol verification cancellation flag (default OFF: with verification, ON: cancel verification, compatible with Mitsubishi communication format)	D8491	Time when error 2 occurred [hours]
M8492		D8492	Time when error 3 occurred [hours]
M8493		D8493	Time when error 4 occurred [hours]
M8494		D8494	Time when error 5 occurred [hours]
M8495		D8495	Time when error 6 occurred [hours]
M8496		D8496	Time when error 7 occurred [hours]
M8497		D8497	Time when error 8 occurred

			[hours]
M8498		D8498	Time when error 9 occurred [hours]
M8499		D8499	Time when error 10 occurred [hours]
M8500		D8500	Error 1 error code
M8501		D8501	Error 2 error code
M8502		D8502	Error 3 error code
M8503		D8503	Error 4 error code
M8504		D8504	Error 5 error code
M8505		D8505	Error 6 error code
M8506		D8506	Error 7 error code
M8507		D8507	Error 8 error code
M8508		D8508	Error 9 error code
M8509		D8509	Error 10 error code
M8510		D8510	Current system running time
M8511		D8511	[unit: second]

## 8.2 System error recording function

### 8.2.1 Error code and content

The error codes and contents stored in the special data registers D8060~D8067 are shown in the following table.

Type	Error Code	Error	Solutions						
I/O structure error, M8060 (D8060) keeping running	Example: 1020	When the I/O head element number "1020" is not available, 1 = outputting X (0 = outputting Y), and 020 = element number	For input relays without the number, write output relay numbers in programs. The Programmable Controller (PC) keeps running. Programmers need to modify programs.						
		<table border="1"> <thead> <tr> <th>Error code</th> <th>Remedy</th> </tr> </thead> <tbody> <tr> <td>1000-1377</td> <td>X input signal is wrong, the serial number is wrong or over-limit</td> </tr> <tr> <td>0000-0377</td> <td>Y output signal is wrong, the serial number is wrong or over-limit</td> </tr> </tbody> </table>		Error code	Remedy	1000-1377	X input signal is wrong, the serial number is wrong or over-limit	0000-0377	Y output signal is wrong, the serial number is wrong or over-limit
		Error code		Remedy					
1000-1377	X input signal is wrong, the serial number is wrong or over-limit								
0000-0377	Y output signal is wrong, the serial number is wrong or over-limit								
PC hardware error, M8061 (D8061) stopped	0000	Normal	Check whether expansion wires are correctly connected.  When the calculation time is more than the D8000 value, check programs.						
	6101	RAM error							
	6102	Calculation circuit error							
	6103	I/O bus error (when M8069 is driven)							
	6104	Expansion device voltage below 24 V (when M8069 is ON)							
	6105	Watchdog timer error							
	6106	System FLASH read and write error							
	6107	System IO setting error							
	6108	The number of times of reading FPGA data in FLASH exceeds the limit		PLC power on and off again					
6109	No FPGA data in FLASH	Autoshop download FPGA data package							
PC/PP communication error, M8062 (D8062) keeping running	0000	Normal	Check whether the devices connected to the Program Panel (PP) is correctly connected to the PC.						
	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							
	0000	Normal							
	6301	Odd/even check error, overflow error, and frame error							
	6302	Incorrect communication character							

Parallel communication error, M8063 (D8063) keeping running	6303	Inconsistent communication data sum	Check whether PCs are powered on, whether the adapter is correctly connected to the controller, and whether adapters are correctly connected to each other.
	6304	Incorrect data format	
	6305	Incorrect instruction	
	6306	Monitor timer timeout	
	6307 to 6311	None	
	6312	Parallel (1:1) protocol character incorrect	
	6313	Parallel (1:1) protocol sum incorrect	
	6314	Parallel (1:1) protocol format incorrect	
	6315	Parallel (1:1) protocol communication timeout error	
	6316~6329	Reserved	
	6330+10*X	MODBUS slave address is set incorrectly, the address is greater than 247;	COM0 communication error: Check whether COM0 communication cables are correctly connected, whether communication formats are matched, whether communication protocols are matched, whether devices are powered on (COM0 can be used as a free port only when devices are powered on; otherwise, it can be only used for monitoring or download), and whether the JP0 jumper is inserted (COM0 can be used as an RS485 free port only when the JP0 jumper is disconnected; if the JP0 jumper is connected, it can be only used for monitoring or download in RS422 mode).
	6331+10*X	The data frame length is wrong, the frame length does not meet the requirements, or the frame length is less than 5;	
	6332+10*X	The address is wrong, the standard error frame; or the receiving and sending addresses are inconsistent;	
	6333+10*X	CRC check error	
	6334+10*X	Unsupported command code, standard error frame; or inconsistent sending and receiving commands; or unsupported commands;	
	6335+10*X	Receive timeout	
	6336+10*X	Data error, standard error frame;	
	6337+10*X	Reserved	
	6338+10*X	Frame error, standard error frame;	
	6339+10*X	Serial port protocol error, when using MODBUS command or RS command, it is not defined as the corresponding protocol;	
	6340	MODBUS slave address setting error	COM1 communication error: Check whether COM1 communication cables are correctly connected and whether communication formats are matched.
	6341	Data frame length error	
	6342	wrong address	
	6343	CRC check error	
	6344	Unsupported command code	
	6345	Receive timeout	
	6346	data error	
	6347	Buffer overflow	
	6348	Frame error	
6349	Serial port protocol error		
6350	MODBUS slave address setting error	COM2 communication error: Check whether COM2 communication cables are correctly connected and	
6351	Data frame length error		
6352	wrong address		

	6353	CRC check error	whether communication formats are matched.
	6354	Unsupported command code	
	6355	Receive timeout	
	6356	data error	
	6357	Buffer overflow	
	6358	Frame error	
	6359	Serial port protocol error	
	6360	MODBUS slave address setting error	COM3 communication error: Check whether COM3 communication cables are correctly connected and whether communication formats are matched.
	6361	Data frame length error	
	6362	wrong address	
	6363	CRC check error	
	6364	Unsupported command code	
	6365	Receive timeout	
	6366	data error	
	6367	Buffer overflow	
	6368	Frame error	
	6369	Serial port protocol error	
	6370	Reserve	
	6371	Parity error, exceeding error, frame error	
6372	Communication character error		
6373	Communication data checksum is inconsistent		
6374	Data format error		
6375	Command error		
6376	Communication component range is out of range		
6370	The communication port exceeds the limit or does not exist		
Parameter error, M8064 (D8064) stopped	0000	Normal	Stop the PC and set parameters correctly.
	6401	Inconsistent program sum	
	6402	Incorrect capacity	
	6403	Incorrect storage area	
	6404	Incorrect instruction area	
	6405	Incorrect file register area	
	6406~6424	Reserve	
	6425	User program check error, download data error	
	6426	User program, including motion control subroutine, incomplete	
	6427	The PLC identifier does not match the identifier in the user program	
	6428	Reserve	
	6429	The program authorization protection flag is wrong, please confirm whether the flag matches the program authorization protection flag in the user program	
	6430~6452	Reserve	

	6453~6465	Poor setting of the changeable device power failure retention area	
Syntax error, M8065 (D8065) stopped	0000	No abnormality	Check whether all instructions are correctly used during programming. Correct errors in program mode.
	6501	Reserve	
	6502	Reserve	
	6503	Command parameter error	
	6504	Duplicate label definition	
	6505	Keep	
	6506	Use undefined instructions	
	6507	Poor definition of label P	
	6508	Bad definition of label I	
	6509	Reserve	
	6510	Reserve	
	6511	High-speed counter and high-speed input port used by interrupt are duplicated	
Circuit error, M8066 (D8066) stopped	0000	Normal	For the entire circuit, errors may occur when instructions are incorrectly combined or incorrectly correlated. Correct the correlation between instructions in programs.
	6601~6604	Reserve	
	6605	1. MPS used for more than nine consecutive times 2. MC, MCR, I (interrupt), and SRET in STL 3. No RET in STL	
	6606	1. No P (pointer) or I (interrupt) 2. No SRET or IRET 3. I (interrupt), SRET, and IRET in main programs 4. STL, RET, MC, and MCR in subprograms and interrupt subprograms	
	6607	1. Incorrect FOR-NEXT relationship, more than six nested layers between FOR and NEXT instructions 2. STL, RET, MC, MCR, IRET, SRET, FEND, and END between FOR and NEXT instructions	
	6608	1. Incorrect MC-MCR relationship 2. There is no NO in MCR 3. SRET, IRET, and I (interrupt) between MC and MCR instructions	
	6609~6617	Reserve	
	6618	Instructions that can be used only in main programs are out of main programs (interrupts and subprograms).	
	6619	Unusable instructions existing between FOR and NEXT instructions: STL, RET, MC, MCR, I, and IRET instructions STL, RET, MC, MCR, I, and IRET	

	6620	Excessive nested layers between FOR and NEXT instructions	
	6621	Incorrect FOR-NEXT quantitative relationship	
	6622	No NEXT instruction	
	6623	No MC instruction	
	6624	No MCR instruction	
	6625	STL used for more than nine consecutive times	
	6626	Unusable instructions existing between STL and RET: MC, MCR, I, SRET, and IRET instructions	
	6627	No RET instruction	
	6628	Unusable instructions existing in main programs: I, SRET, and IRET instructions	
	6629	No P or I	
	6630	No SRET or IRET instructions	
	6631	SRET in inapplicable scenarios	
	6632	FEND in inapplicable scenarios	
	0000	Normal	
	6701	1. No jump addresses for CJ and CALL 2. Volume label following the END instruction 3. Separate volume label existing between FOR and NEXT or between subprograms	
	6702	More than six nested layers for CALL	
	6704	More than six nested layers between FOR and NEXT	
	6705	Application instruction operands beyond target elements	
	6706	Element number range and value of application instruction operands overflow	
	6707	File register stored because of failure to set file register parameters	
	6708	The relationship between FOR&NEXT, MC, MCR, STL, subroutine and interrupt program is confused	
	6709	Others (IRET and SRET forgotten; incorrect FOR-NEXT relationship)	
	6710	Local variables are used in the main program	
	6711	The device used in the instruction is repeatedly used incorrectly or conflicted	
	6712	An undefined interrupt is used in the	
Calculation error, M8067 (D8067) keeping running			Check errors occurring during calculation and whether program modification or operands of application instructions are correct. Even if no syntax or circuit errors occur, calculation errors can be caused by many factors. (Example) Although T200Z is correct, when Z is 100, T is 300, which indicates an element number overflow.



		system		
	6720	CALL instruction SRET does not match		
	6713~6729	Reserve		
	6730	Sampling time (TS) out of target range (TS = 0)	PID calculation stopped	
	6731	Reserve		
	6732	Input filter constant (a) out of target range (a<0 or 100<a)		
	6733	Proportional valve (KP) out of target range(KP<0)		
	6734	Integral time (TI) out of target range (TI< 0)		
	6735	Differential valve (KD) out of target range (KD<0 or 201<KD)		
	6736	Differential time out of target range(TD<0)		
	6737~6739	Reserve		
	6740	Sampling time (TS)		
	6741	Reserve		
	6742	Measured variable overflow ( $\Delta PV < -32768$ or $< \Delta PV$ )		
	6743	Deviation overflow (EV<-32768 or 32767<EV)		
	6744	Integral value overflow (out of -32768 to 32767)		
	6745	Differential value overflow because of differential valve (KP) overflow		
	6746	Differential value overflow (out of -32768 to 32767)		
	6747	PID operation result overflow (out of -32768~32767)		
	6748	Reserve		
	6749	Reserve	Check the corresponding instruction and adjust the instruction according to the error content.	
	6750	The accessed expansion module does not exist, or the module serial number exceeds the limit		
	6751~6759	Reserve		
	6760	More than six high-speed instructions (DHSZ)		
	6761~6769	Reserve		
	6770	High-speed output instructions use port duplication or conflict		
	6771	The high-speed output command signal is incorrect		

### 8.2.2 PLC running time/abnormal record

PLC running time/abnormal design is mainly used to record PLC running time and record error information. It is convenient for customers and us to analyse the problems that occur during the operation of the PLC.

Time record: register D8511, D8510 (32 bits), time unit: second.

Error log:

Error time (hours)	Error content
D8490	D8500
D8491	D8501
D8492	D8502
D8493	D8503
D8494	D8504
D8495	D8505
D8496	D8506
D8497	D8507
D8498	D8508
D8499	D8509


Recording principle:

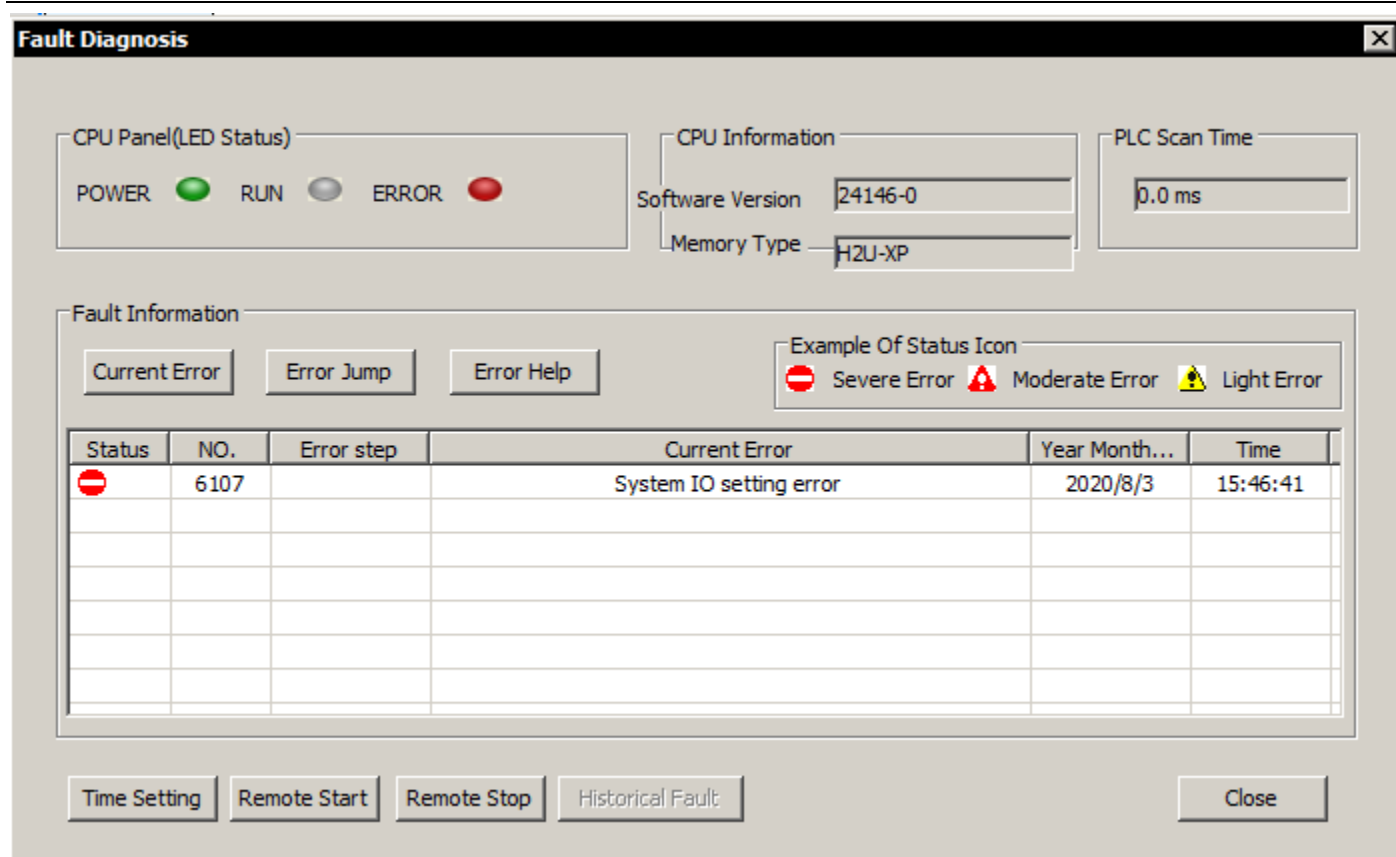
- 1) When D8060--D8067 report error codes, record the error.
- 2) The error code that appears is already in the error record table, only the error time is updated, and the position in the error table does not change.
- 3) When there are more than 10 error code records, the first-in-first-out principle is adopted to discard the error code that appeared first.
- 4) The user clears the program and the component storage space cannot clear the error record and running time.

### 8.2.3 Use Autoshop software for fault diagnosis and handling

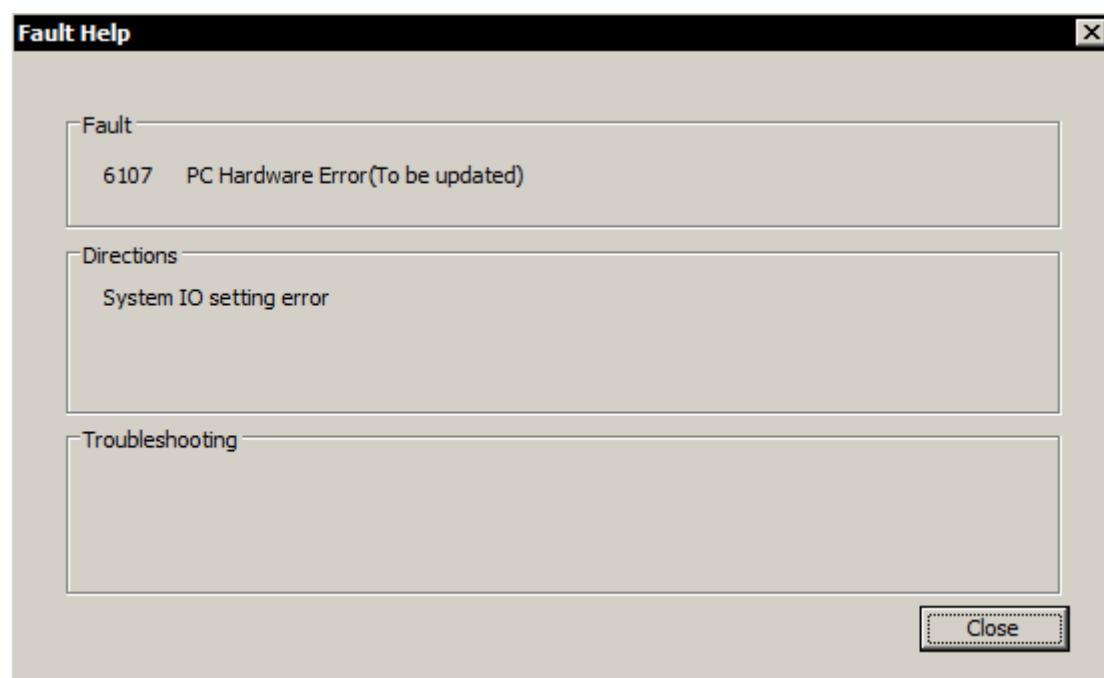
When the user uses Autoshop software to monitor the PLC, the current PLC running status can be viewed through the lower right corner of the software interface, including the PLC RUN/STOP status, fault status, PLC current version, current scan time.



Double-click  the fault status to view the current error information of the PLC, as shown in the figure below.



The current error details can be double-clicked on the error message display area or selected and clicked on the error help, the error help dialog box will pop up, as shown in the figure below.



## 8.3 Error code register

According to the following timing check, store the error code of the preceding item in the special data register D8060~D8067(Error code storage register).

Error	OFF-to-ON Switch	First STOP-to-RUN Switch	Others
M8060 I/O address error	Check	Check	Calculating
M8061 PC hardware error	Check	N/A	Calculating
M8062 PC/PP communication error	N/A	N/A	When signals are received from PP
M8063 module communication error	N/A	N/A	When signals are received from the other
M8064 parameter error M8065 syntax error M8066 circuit error	Check	Check	When programs are changing (STOP) When program are transmitted (STOP)
M8087 calculation error M8088 calculation error lock	N/A	N/A	Calculating (RUN)

D8060 ~ D8067 each store an error content. When multiple errors occur in the same error item, the error code that occurred is still stored whenever the cause of the error is eliminated. When there is no error, "0" is stored.

## 8.4 Software difference between H2U series MTQ type and MT type

The H2U series MTQ PLC has 6 high-speed inputs, 5 high-speed outputs, and high-speed signal processing frequency is increased to meet the needs of multiple high-speed applications.

Comparison of hardware response characteristics:

Input port	Standard H1U/H2U-XP model	MTQ type
X0	60kHz	100kHz
X1	60kHz	100kHz
X2	10kHz	100kHz
X3	10kHz	100kHz
X4	10kHz	100kHz
X5	10kHz	100kHz
Other X ports	Same indicators	
Y0	100kHz	100kHz
Y1	100kHz	100kHz
Y2	100kHz	100kHz
Y3		100kHz
Y4		100kHz
Other Y ports	Same indicators	

Comparison of the high frequency index of AB-phase high-speed counter:

AB phase counter	H1U/H2U-XP model	H2U-3232MTQ
C251~C255	30kHz	50kHz

Supported instruction comparison:

Instruction	H1U/H2U-XP output port	Output ports supported by MTQ
PLSY	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4
PLSR	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4
PLSV	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4
PWM	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4
DRVI	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4
DRVA	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4
ZRN	Y0, Y1, Y2	Y0, Y1, Y2, Y3, Y4

Description of system monitoring registers involved in MT/MTQ type high-speed pulse output:

M8145	Y000 pulse output stop	D8145	DRVI, the offset speed when DRVA is executed DRVI, the highest speed when DRVA is executed [default 100,000]
M8146	Y001 pulse output stop	D8146	
M8147	Y002 pulse output stop	D8147	
M8148	Y001 pulse output monitoring	D8148	DRVI, DRVA acceleration and deceleration time during execution [default 100]
M8149	Y002 pulse output monitoring	D8149	PLSY&PLSR output Y002 corresponding pulse number cumulative value
M8150	Y003 pulse output monitoring	D8150	
M8151	Y004 pulse output monitoring	D8151	PLSY&PLSR output Y003 corresponding pulse number cumulative value
M8152	Y002 pulse output stop	D8152	
M8153	Y003 pulse output stop	D8153	PLSY&PLSR output Y004 corresponding pulse number cumulative value
M8154	Y004 pulse output stop	D8154	
M8155	Reserve	D8155	
M8156	Reserve	D8156	Y0 port clear signal definition (ZRN) [default 5=Y005]
M8157	Reserve	D8157	Y1 port clear signal definition (ZRN) [default 5=Y005]
M8158	Reserve	D8158	Y2 port clear signal definition (ZRN) [default 5=Y005]
M8159	Reserve	D8159	Y3 port clear signal definition (ZRN) [default 5=Y005]
M8160	(XCH) SWAP function	D8160	Y4 port clear signal definition (ZRN) [default 5=Y005]
M8161	ASC/RS/ASCII/HEX/CCD bit processing mode	D8161	Reserve
M8162	High-speed parallel connection mode	D8162	Reserve
M8163	Reserve	D8163	Reserve
M8164	(FROM/TO) Variable transmission point mode	D8164	(FROM/TO) Transmission point designation mode
M8165	Reserve	D8165	Deceleration time during execution of PLSR, DRVI, DRVA [default 100] is determined by M8135 whether it is valid or not [Y0]
M8166	Reserve	D8166	Deceleration time during execution of PLSR, DRVI, DRVA [default 100] is determined by M8136 whether it is valid or not [Y1]
M8167	(HEY)HEX data processing function	D8167	Deceleration time during execution of PLSR, DRVI, DRVA [default 100] is determined by M8137 whether it is valid or not [Y2]

## 8.5 Supplementary Instructions for H2U Series MTP Software

### 8.5.1 Description of DRVI and DRVA special functions

Port	Special function valid		Special function not valid	
Y000	Effective conditions	M8135=ON	Effective conditions	M8135=OFF
	Acceleration time	D8104 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8165 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8034	Bias speed	D8145
	Maximum speed	D8051/D8050 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Invalid to changing the number of pulses during operation	
	General function designation			
	Output stop control	M8145		
	Output monitoring	M8147		
	Pulse re-output	M8085		
	Cumulative pulse count	D8140, D8141		
Y0&Y1 cumulative pulse	D8136, D8137			
Y001	Effective conditions	M8136=ON	Effective conditions	M8136=OFF
	Acceleration time	D8105 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8166 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8035	Bias speed	D8145
	Maximum speed	D8053/D8052 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Invalid to changing the number of pulses during operation	
	General function designation			
	Output stop control	M8146		
	Output monitoring	M8148		
	Pulse re-output	M8086		
	Cumulative pulse count	D8142, D8143		
Y0&Y1 cumulative pulse	D8136, D8137			
Y002	Effective conditions	M8137=ON	Effective conditions	M8137=OFF
	Acceleration time	D8106 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8167 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8036	Bias speed	D8145
	Maximum speed	D8055/D8054 (default 100000)	Maximum speed	D8146/D8147 (default 100000)

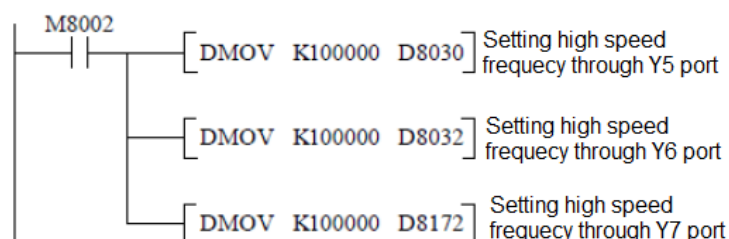
Port	Special function valid		Special function not valid	
	Effective to change the number of pulses during operation		Changing the number of pulses is invalid during operation	
	General function designation			
	Output stop control	M8152		
	Output monitoring	M8149		
	Pulse re-output	M8087		
Cumulative pulse count	D8150, D8151			
Y003	Effective conditions	M8138=ON	Effective conditions	M8138=OFF
	Acceleration time	D8107 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8168 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8037	Bias speed	D8145
	Maximum speed	D8057/D8056 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Changing the number of pulses is invalid during operation	
	General function designation			
	Output stop control	M8153		
	Output monitoring	M8150		
	Pulse re-output	M8088		
Cumulative pulse count	D8152, D8153			
Y004	Effective conditions	M8139=ON	Effective conditions	M8139=OFF
	Acceleration time	D8108 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8169 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8038	Bias speed	D8145
	Maximum speed	D8059/D8058 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Changing the number of pulses is invalid during operation	
	General function designation			
	Output stop control	M8154		
	Output monitoring	M8151		
	Pulse re-output	M8089		
Cumulative pulse count	D8154, D8155			
Y005	Effective conditions	M8141=ON	Effective conditions	M8141=OFF
	Acceleration time	D8087 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8090 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8093	Bias speed	D8145
	Maximum speed	D8030/D8031 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Changing the number of pulses is invalid during operation	



Port	Special function valid		Special function not valid	
	General function designation			
	Output stop control	M8155		
	Output monitoring	M8144		
	Pulse re-output	M8090		
	Cumulative pulse count	D8138, D8139		
Y006	Effective conditions	M8142=ON	Effective conditions	M8142=OFF
	Acceleration time	D8088 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8091 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8094	Bias speed	D8145
	Maximum speed	D8032/D8033 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Changing the number of pulses is invalid during operation	
	General function designation			
	Output stop control	M8156		
	Output monitoring	M8158		
	Pulse re-output	M8091		
	Cumulative pulse count	D8162, D8163		
Y007	Effective conditions	M8143=ON	Effective conditions	M8143=OFF
	Acceleration time	D8089 (default 100ms)	deceleration time	D8148 (default 100ms)
	Deceleration time	D8092 (default 100ms)	deceleration time	D8148 (default 100ms)
	Offset speed	D8095	Bias speed	D8145
	Maximum speed	D8172/D8173 (default 100000)	Maximum speed	D8146/D8147 (default 100000)
	Effective to change the number of pulses during operation		Changing the number of pulses is invalid during operation	
	General function designation			
	Output stop control	M8157		
	Output monitoring	M8159		
	Pulse re-output	M8092		
	Cumulative pulse count	D8072, D8071		

Remarks:

When the valid conditions of Y5 (M8141), Y6 (M8142), and Y7 (M8143) are ON, the maximum speed needs to be set. Please execute the following instructions before powering on, otherwise the maximum speed will not be reached.



## 8.5.2 Description of ZRN instruction

Port	Characteristics	Corresponding value
Y000	Clear port enable	M8140=ON
	Clear port number	D8156[default Y010]
	Output stop control	M8145
	Output monitoring	M8147
	Cumulative pulse count	D8140, D8141
	Y0&Y1 cumulative pulse	D8136, D8137
Y001	Clear port enable	M8140=ON
	Clear port number	D8157[default Y011]
	Output stop control	M8146
	Output monitoring	M8148
	Cumulative pulse count	D8142, D8143
	Y0&Y1 cumulative pulse	D8136, D8137
Y002	Clear port enable	M8140=ON
	Clear port number	D8158[default Y012]
	Output stop control	M8152
	Output monitoring	M8149
	Cumulative pulse count	D8150, D81451
Y003	Clear port enable	M8140=ON
	Clear port number	D8159[default Y013]
	Output stop control	M8153
	Output monitoring	M8150
	Cumulative pulse count	D8152, D8153
Y004	Clear port enable	M8140=ON
	Clear port number	D8160[default Y014]
	Output stop control	M8154
	Output monitoring	M8151
	Cumulative pulse count	D8154, D8155
Y005	Clear port enable	M8140=ON
	Clear port number	D8161[default Y015]
	Output stop control	M8155
	Output monitoring	M8144
	Cumulative pulse count	D8138, D8139
Y006	Clear port enable	M8140=ON
	Clear port number	D8196[default Y016]
	Output stop control	M8156
	Output monitoring	M8158
	Cumulative pulse count	D8162, D8163
Y007	Clear port enable	M8140=ON
	Clear port number	D8197[default Y017]
	Output stop control	M8157
	Output monitoring	M8159
	Cumulative pulse count	D8170, D8171

Note:

- H2U-3232MTP is 8 channels of high-speed output (each channel up to 100K, does not affect each other);
- Does not support high-speed counting, external I/O interrupt, pulse capture, SPD instruction;
- PLSR instruction can separately set the acceleration and deceleration time and the number of pulses can be changed during operation;
- The number of PLSY command pulses can be changed during operation;
- Y0~Y7 cannot be used as signals of pulse output direction.

## 8.6 List of commands not supported by H2U series MTQ/MTP models

Classification	Instruction	Features	Whether to support H2U series MTQ/MTP models
Peripheral equipment	CANTX	CAN data transmission	No
	CANRX	CAN data reception	No
	RS2	Serial data transmission 2	No
	MODBUS2	Data transfer 2	No
Floating point	ECMP	Binary floating-point number comparison	No
	EZCP	Interval comparison of binary floating-point numbers	No
	EMOV	Binary floating-point data movement	No
	EBCD	Binary-10 floating-point number conversion	No
	EBIN	Decimal-2 decimal floating-point number conversion	No
	EADD	Binary floating-point number addition	Yes
	ESUB	Binary floating-point number subtraction	Yes
	EMUL	Binary floating-point multiplication	Yes
	EDIV	Binary floating-point number division	Yes
	EXP	Binary floating-point exponential operation	No
	LOGE	Natural logarithm operation of binary floating-point no.	No
	DLOG	Common logarithmic operations of binary floating-point nos.	No
	ESQR	Square root of binary floating-point number	No
	SIN	Floating-point SIN operation	No
	COS	Floating-point COS operation	No
	TAN	Floating-point TAN operation	No
	ASIN	Binary floating-point ASIN operation	No
	ACOS	Binary floating-point number ACOS operation	No
	ATAN	Binary floating-point number ATAN operation	No
	RAD	Binary floating-point angle → radians	No
DEG	Binary floating-point radian → angle	No	
SINH	Binary floating-point number SINH operation	No	
COSH	Binary floating-point COSH operation	No	
TANH	Binary floating-point number TANH operation	No	
Positioning	DSZR	Return to origin with DOG search	No
	ABS	ABS position reading	No

## 8.7 Description of enhanced functions of H2U series high-speed processing instructions

In addition to compatible FX1n/FX2n instructions, H2U series PLC high-speed programming instructions have enhanced some of the instruction functions. These enhanced functions are generally implemented through the system's M variables. The high-speed command functions involved are as follows:

Instruction	M variables used	When an M flag is ON, enhanced functions are enabled.
SPD	M8100 to M8105 correspond to ports X0 to X5, respectively.	The number of pulses per minute (running speed) can be calculated through real-time sampling of pulse input frequencies and through internal operation.
PLSY	M8135 to M8137 correspond to Y0 to Y0, respectively.	You can change the number of output pulses when the device is running.
	M8085 to M8087 correspond to Y0 to Y2, respectively.	When special flags are set to ON, the system immediately executes the pulse output instruction without disabling the previous power flow.
	M8090 to M8092 correspond to Y0 to Y2, respectively.	A user interrupt can be enabled after pulses are output.
PLSR DRVI DRVA	M8135 to M8137 correspond to Y0 to Y2, respectively.	You can change the number of output pulses when the device is running. You can change the deceleration time (so that the acceleration time is different from the deceleration time).
	M8185 to M8087 correspond to Y0 to Y2, respectively.	When special flags are set to ON, the system immediately executes the pulse output instruction without disabling the previous power flow.
	M8090 to M8092 correspond to Y0 to Y2, respectively.	A user interrupt can be enabled after pulses are output.
High-speed counter interrupt	M8084; D8084 to D8086	High-speed counter multi-user interrupt (a maximum of 24 interrupts are supported: I507 to I530)

The instructions for using enhanced high-speed instructions are as follows:

### 1) Description of SPD instruction

16-bit	32-bit	P	SPD	Pulse density detection
✓				
7 steps			Instruction format: SPD S1 S2 D	

Function enabling flags M8100 to M8105 are used for X000 to X005 respectively, indicating that enhanced functions take effect. Each flag can be set separately.

a) If function enabling flags (M8100 to M8105) are OFF, basic functions of the SPD instruction are enabled. [S1] indicates the pulse signal input port, which is specified within the range X00 to X05.

[S2] indicates the pulse detection duration (ms), which is specified within the range 1 to 32,767.

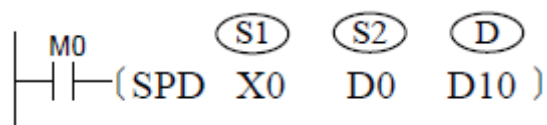
[D] indicates the count of pulses, occupying three consecutive address units starting with [D].

[D + 0] indicates the number of pulses within the set duration [S2], which is 16-bit data.

[D + 1] indicates the real-time number of pulses.

[D + 2] indicates remaining time of the sampling period.

Example:



X0 indicates the pulse signal input port.

D0 indicates the unit time (ms).

D10 indicates the total number of pulse within D0.

D11 indicates the counts within D0.

D12 indicates the remaining time (ms).

- b) If function enabling flags (M8100 to M8105) are ON, enhanced functions of the SPD instruction are enabled.

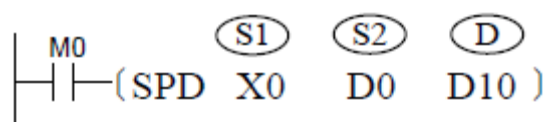
[S2] indicates the pulse detection duration (ms), which is specified within the range 1 to 32,767.

[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], which is 16-bit data.

[D + 1], [D + 2] indicates the number of pulses per minute, which is 32-bit data.

Example:



X0 indicates the pulse signal input port.

D0 indicates the unit time, ranging from 1 to 32767ms.

D10 indicates the total number of pulses within D0.

D11 and D12 indicate the running frequency, equal to (the number of pulse in one minute) x 10 (unit: 0.1).

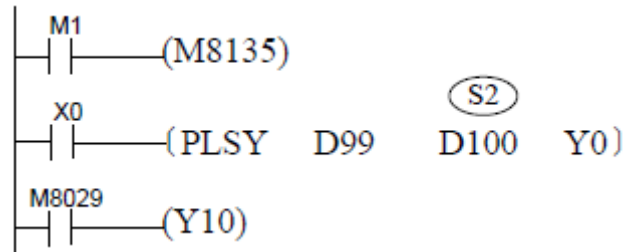
Special instructions for PLSY instruction.

16-bit	32-bit	P	PLSY	Pulse output
✓	✓			
7steps	13steps		Instruction format: PLSY S1 S2 D	

- c) The following functions can be enabled by setting special flags M8135 to M8139 (corresponding to Y0 to Y4, respectively) to ON.

You can change the number of output pulses when the device is running (by increasing or decreasing pulses).

Example:

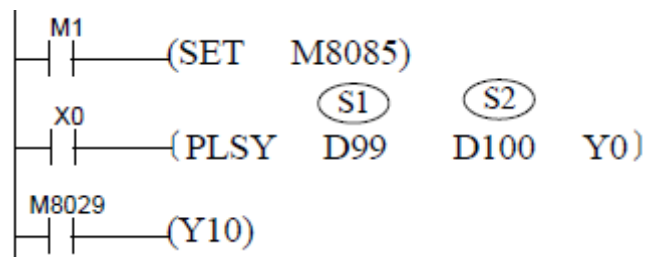


When M1 is ON, the special function takes effect. X0 is connected and the PLSY instruction is executed. You can change D100 (the number of pulses) during pulse transmission by increasing or decreasing pulses. (Note: The new D100 value should be greater than the number of pulses transmitted.)

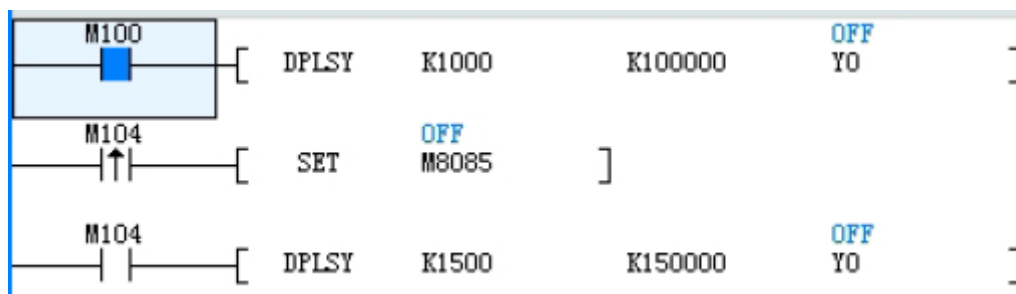
- d) The following functions can be enabled by setting special flags M8085 to M8089 (corresponding to Y0 to Y4, respectively) to ON.

When special flags are set to ON, the system immediately executes the next pulse output instruction without disabling the previous power flow.

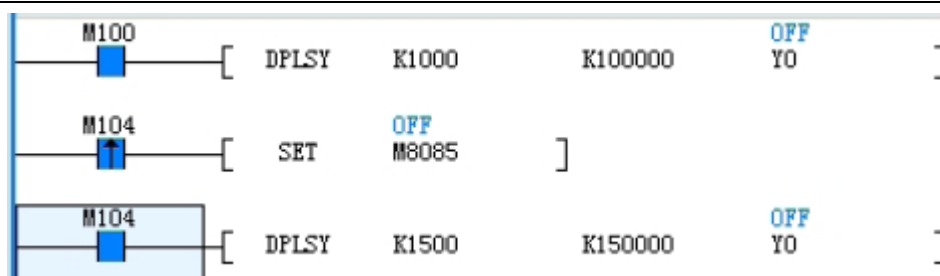
Example:



When the X0 value is 1, during pulse transmission or after pulses are transmitted, if M8085 is set to ON, the PLSY instruction is restarted and the system re-transmits pulses based on the preset frequency and number of pulses S1 and S2, that is, the current D99 and D100 values) without switching X0 from ON to OFF and to ON.



When M100 is activated and is in the process of sending pulses (as shown above), start M104, then the pulse command driven by M100 will be automatically cleared, and the pulse output command driven by M104 (as shown below) will be executed immediately without waiting for M100 to be disconnected.



- e) By using special bits M8090~M8092 (respectively corresponding to Y0~Y2) to be ON, the following functions can be realized:

A user interrupt can be executed after the pulse output is completed; the specific correspondence is as follows:

Port Number	Special Bit	User Interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

When the special function bit (M8090-M8092) is ON, the user interrupt I502 ~ I504 will be executed immediately after the specified number of pulses have been sent.

- 2) Special instructions for PLSR, DRVI, DRVA instructions

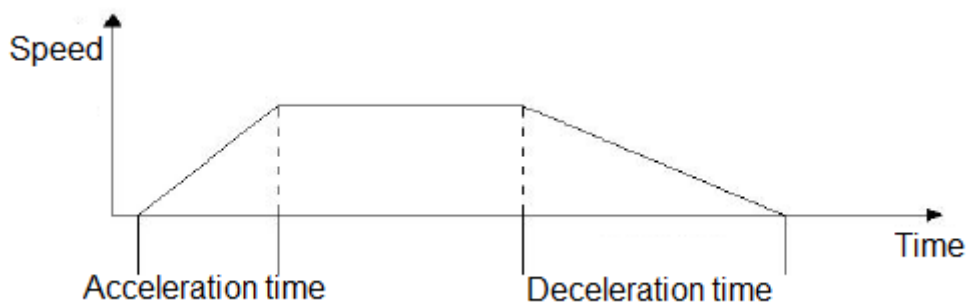
- a) By using special bits M8135~M8137 (respectively corresponding to Y0~Y2) to be ON, the following functions can be realized:

The number of output pulses can be changed during operation (large or small). At the same time, the acceleration and deceleration time are defined by the following registers, and the time unit is ms: (PLSR acceleration time is determined by S3 in PLSR S1 S2 S3 D)

Port Number	Soft element	Register for modifying acceleration time	Register for modifying deceleration time
Y000	M8135	D8104	D8165
Y001	M8136	D8105	D8166
Y002	M8137	D8106	D8167

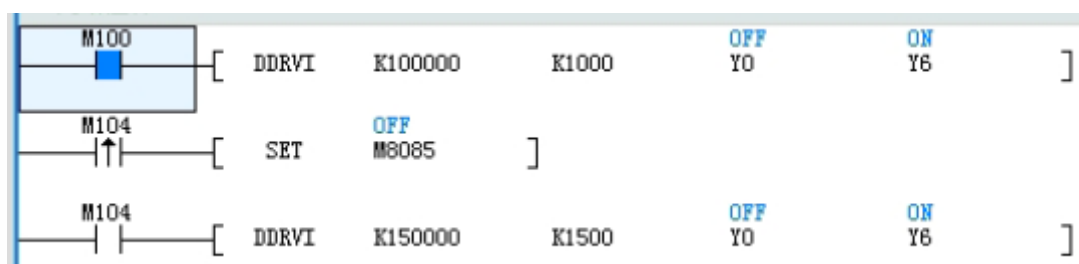
When the special bit (M8135~M8137) is ON, the total number of pulses can be changed at any time during the execution of the PLSR, DRVI, DRVA instructions (Note: the total number of pulses after the change must be greater than the number of pulses currently sent)

The acceleration time and deceleration time can be different, but the acceleration and deceleration time must be modified before the instruction is executed.

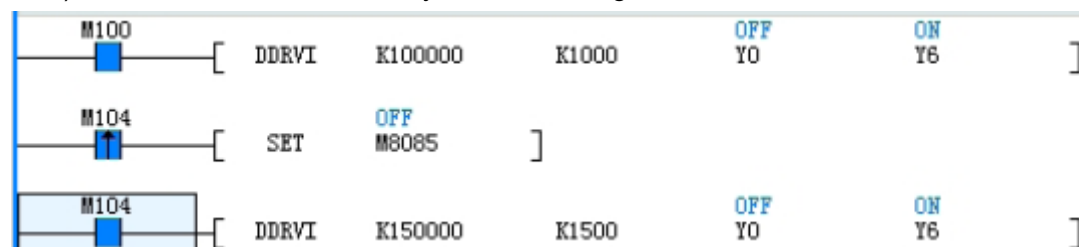


- b) By using special bits M8085~M8087 (corresponding to Y0~Y2 respectively) to be ON, the following functions can be realized:

When the drive special bit is ON, the next pulse output command can be started immediately, without the need to disable the previous power flow.



When M100 is activated and is in the process of sending pulses (as shown above), start M104, then the pulse command driven by M100 will be automatically cleared, and the pulse output command driven by M104 (as shown below) will be executed immediately without waiting for M100 to be disconnected.



- c) By using special bits M8090~M8092 (respectively corresponding to Y0~Y2) to be ON, the following functions can be realized:

A user interrupt can be executed after the pulse output is completed; the specific corresponding is as follows:

Port Number	Special Bit	User Interrupt
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

- 3) Use description of multi-user interrupt of high-speed counter (XP does not have this function)

In order to meet the requirement of supporting multiple high-speed free tasks when the high-speed counter is running, multi-user interrupt of the high-speed counter is provided (XP does not have this function) (maximum support 24, all of which are extended interrupt numbers), set and compare are defined in the form of a data



table:

Flag bit	Description
M8084	Enable the multi-user interrupt of the high-speed counter for ON (XP does not have this function)
D8084	High-speed counter serial number 235~255
D8085	The number of corresponding user interrupts, the maximum is 24, from I507 to I530
D8086	Corresponding to the serial number of multiple comparison points, it can only be a D component, and it is a double word width, for example, 200 is a double word starting from D200

Routines for storing comparison point data:

When using the high-speed counter C235, D8084=235; when using the 32-bit register at the beginning of D200 as the data comparison value, D8086=200; when 5 interrupt outputs are to be executed, D8085=5; finally M8084=ON, the function is valid



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

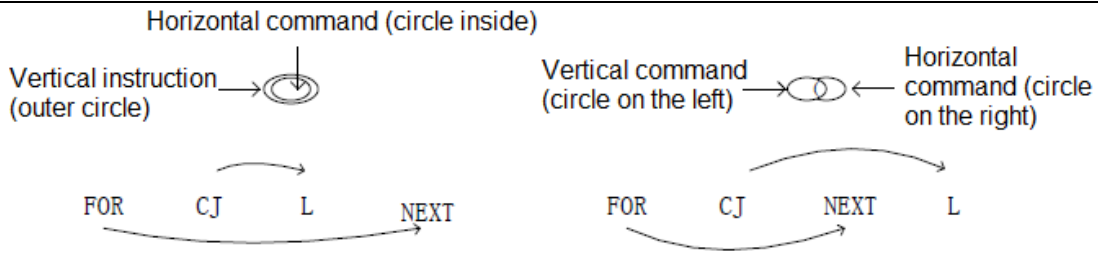
Each interrupt can be generated by the value of the high-speed counter and the value of the recording unit.

Note: When the table is executed, first execute it from top to bottom. Only after each cell of data is successfully compared will it continue to the next cell. After the last cell of the table is executed, M8133=ON, and the data comparison returns at this time. Go to the first cell of the table, it is recommended to clear the C235 data when returning to the first cell., so that the table will execute cyclically. This function is only effective when counting.

## 8.8 Interrelationship of program flow control instructions

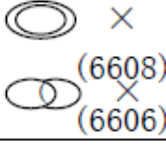
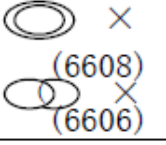
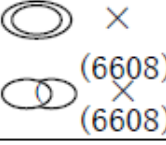
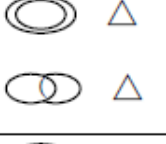

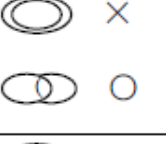
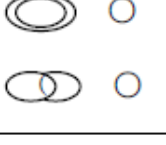
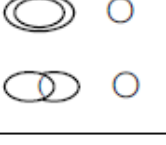
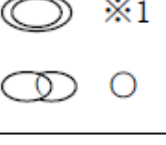
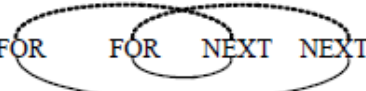
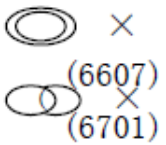
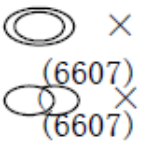
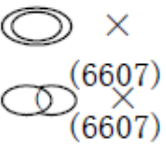
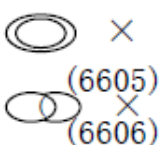
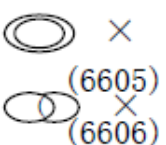
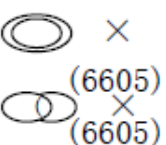
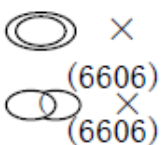
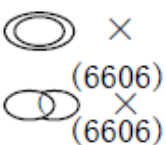
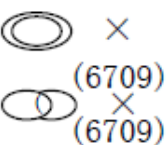
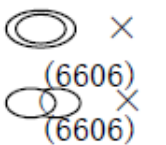
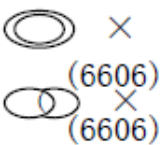
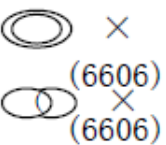
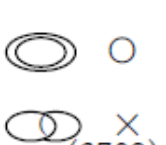
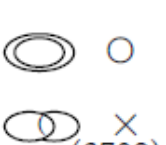
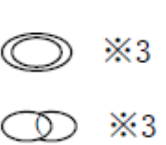
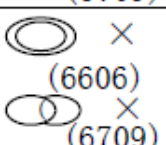
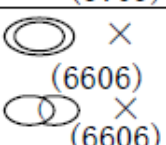
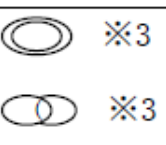
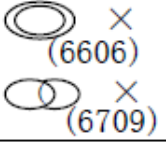
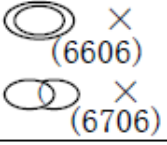
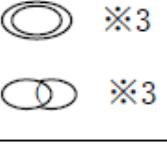
The relationship between various program flow control instructions is shown below.

In the following table,  means the including relation,  indicates that the interval before and after is repeated. In the AutoShop programming environment, subprograms and interrupt programs are written separately, so there is no P-SRET, I-IRET, FEND-END, O-FEND, O-END.



Horizontal \ Vertical	MC-MCR	CJ-L	EI-DI	FOR-NEXT	STL-RET
MC-MCR					
CJ-L					
EI-DI					
FOR-NEXT					
STL-RET					
P-SRET					
I-IRET					
FEND-END					
0-FEND					
0-END (NO-FEND)					

- Can be combined with each other without any problems.
- × Combinations that are forbidden to use, ( ) are the wrong serial numbers.
- △ Although it is not in the prohibited range, it may complicate the action. It is recommended to avoid the combination.

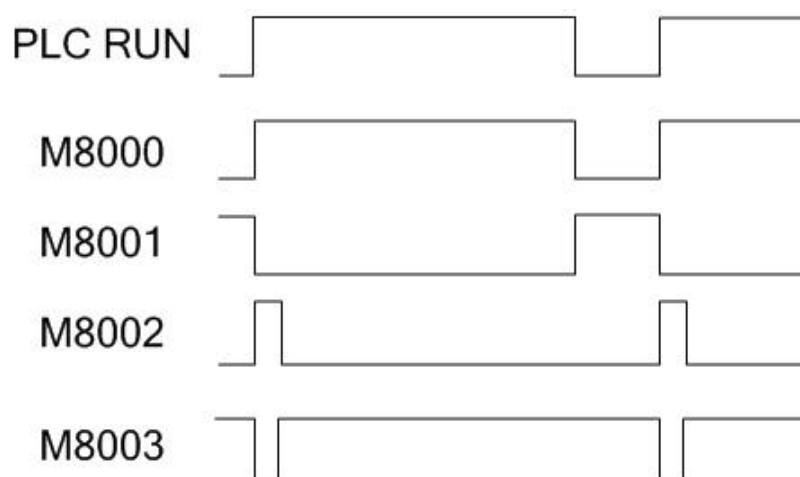
Horizontal Vertical	P-SRET	I-IRET	FEND-END	Remarks
MC-MCR				
CJ-L				※1 is the state of forgetting DI. Not a mistake.
EI-DI				※2 
FOR-NEXT				Act as shown by the solid line. ※3 The initial FEND and END are valid and are not the intended program. Not a mistake.
STL-RET				Except for some exceptions, instructions with containment relationships can be combined. But we must pay attention to the following special cases.
P-SRET				① MC-MCR cannot be used in FOR-NEXT, STL-RET, PSRET, and I-RET.
I-IRET				② STL-RET cannot be used in FOR-NEXT, P-SRET, and I-RET.
FEND-END				③ It is not possible to use I, IRET, SRET, FEND, END and other instructions for masking operations between MC-MCR, FOR-NEXT, PSRET, and I-RET instructions.
0-FEND				
0-END (NO-FEND)				

## 8.9 Function description of some special relays and registers

### System operating status

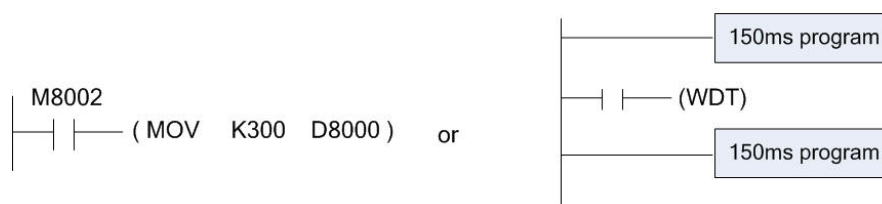
#### PLC running flag M8000-M8003

- 1) M8000 is a contact that remains ON (namely a normally open contact A for operation monitoring) when the switch is turned to RUN position. When the PLC is running, M8000 remains ON.
- 2) M8001 is a contact that remains OFF (namely a normally closed contact B for operation monitoring) when the switch is turned to RUN position. When the PLC is running, M8001 remains OFF.
- 3) M8002 is ON in the first scan cycle when the PLC is running and then remains OFF. The pulse width corresponds to one scan cycle. You can use this contact for initialization.
- 4) M8003 is OFF in the first scan cycle when the PLC is running and then remains ON; that is, negative pulses (turned off immediately the PLC runs) are started.



#### Watchdog timer: D8000

- 1) A watchdog timer is used to monitor the PLC scan time. When the scan time exceeds the specified threshold, the red error indicator remains ON and all outputs are OFF.
- 2) The default time is 200 ms. When instruction operations are excessively complex or the PLC is connected to many special modules, the scan time will be overlong. You can monitor D8010 to D8012 to check whether the scan time is more than the D8000 value. In this case, you can use an MOV instruction to change the watchdog timer value to 300 ms, or add a WDT instruction to PLC programs. When the CPU executes the WDT instruction, the embedded watchdog timer is zeroed so that the scan time will not exceed the preset value.



The maximum watchdog timer value is 32,767 (ms). However, if the value is too big, operation abnormality detection will be delayed. Therefore, if an operation is not so complex that the scan lasts for more than 200ms, set the watchdog timer threshold to less than 200ms.

PLC program version: D8001

For the board program version, if D8001 is 24115, 24 indicates H2U and 115 indicates V1.15, which means the device is an H2U PLC in V1.15.

PLC program capacity: D8002

The program capacity includes 4 KB, 8 KB, 16 KB, and 24 KB. The capacity of an H2U PLC is 24 KB.

Syntax check signal: M8004 and D8004

M8004: When any of error signals M8060 to M8067 (except M8062) is ON, M8004 is ON. It is used to monitor PLC system errors. D8004 indicates BCD values of M8060 to M8067, 0 by default.

Battery voltage detection: M8005, M8009, D8005 and M8030

- 1) When the D8005 value is less than the D8006 value (2.6 V by default), M8005 is output.
- 2) When the low battery voltage alarm is given (M8005 is ON), M8006 is ON (latched), and the PLC cannot be reset after restarting.
- 3) M8007 and M8008 act after 5 ms of AC de-energization. If the de-energization time is not greater than the D8008 value, the program continues; otherwise, the program is disabled, and M8000 is OFF.
- 4) The M8009 is ON upon a power failure (24 V) encountered by the expansion module, and D8009 records the module number.
- 5) When M8030 is ON, the low battery voltage alarm is shielded.

## System clock

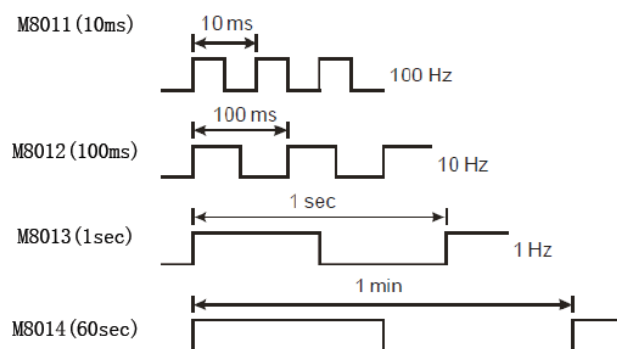
### Scan time monitoring: D8010 to D8012

The current value, minimum value, and maximum value of the program scan time are stored in D8010 to D8012.

- 1) D8010 indicates the current scan time.
- 2) D8011 indicates the minimum scan time.
- 3) D8012 indicates the maximum scan time.

### Cyclic free-run clock: M8011 to M8014

A PLC has the following four free-run clocks. When the PLC is powered on, the clocks act automatically. Free-run clocks act even if the PLC is stopped. Therefore, free-run clocks do not start in synchronization with the PLC.



### Real-time clock: D8013 to D8019 and M8015 to M8019

- 1) When M8015 is ON, the clock stops.
- 2) The clock is calibrated by  $\pm 30$ s each time M8017 is set to ON. When the D8013 value (seconds) is between 1 and 29, the value is zeroed while the value in minutes remains unchanged; when D8013 value is between 30 and 59, the value is zeroed while 1 is added to the value in minutes.
- 3) Normally the year is indicated by double digits (for example: the year of 2009 is indicated by 09). To indicate the year in four-digit format, execute the following statement in a scan cycle.

Address No.	Name	Action Function
M8015	Clock stop and time calibration	When it is ON, the clock stops. The time is written through the ON-to-OFF edge and then actions are performed again.
M8016	Time no longer displayed	When it is ON, the time is no longer displayed.

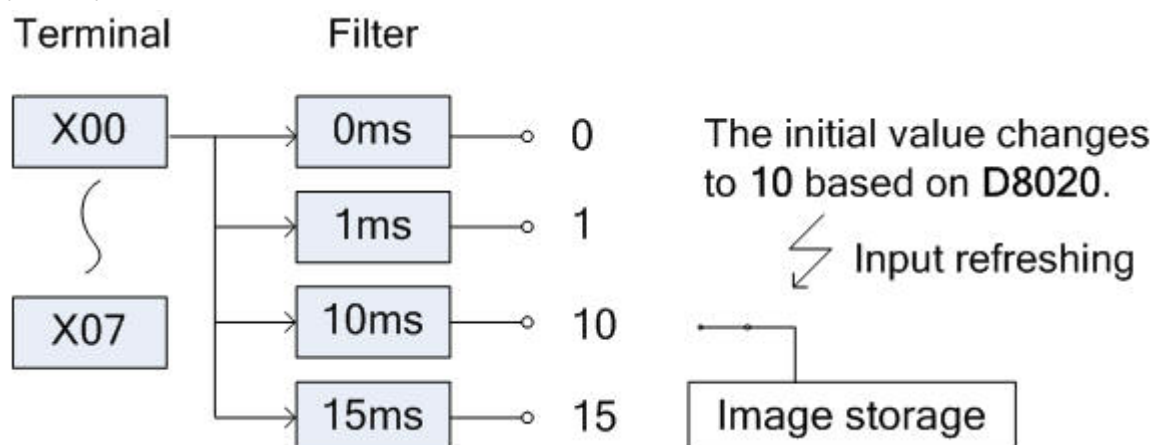
Address No.	Name	Action Function
M8017	30s calibration	The time is calibrated to the second through the ON-to-OFF edge. (0-29s: 0s; 30-59s: X minute)
M8018	Installation detection	It is ON when normal.
M8019	RTC error	During time calibration, it is ON when the special register value is out of range.

Address No.	Name	Range	Action Function
D8013	Second	0 to 59	It is used to write the initial value or read the current value.
D8014	Minute	0 to 59	
D8015	Hour	0 to 23	
D8016	Day	0 to 31	
D8017	Month	0 to 12	
D8018	Year	0 to 99	
D8019	Week	0 to 6	

### Instruction flag

Input filter adjustment: D8020

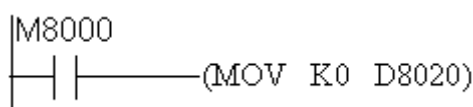
- 1) You can set the response time of pulse receiving for X0-X7 input ports through D8020. The response time ranges from 0 to 60ms and is 10ms by default.
- 2) When high-speed counters and interrupt insertion are enabled, the minimum filter time is applied to corresponding ports automatically, while the filter time for unrelated X0-X7 ports remains unchanged (D8020).



- 3) When the PLC power supply changes from OFF→ON, the content of D8020 automatically becomes 10
- 4) The filter constant becomes 0ms when the following programs are executed. However, C-R filter hardware is configured for the input ports. Even if you set the filter constant to 0, the reaction time will not be less than



10ms (the reaction time for X2-X5 ports of 40-point and 60-point PLCs is no less than 50ms).



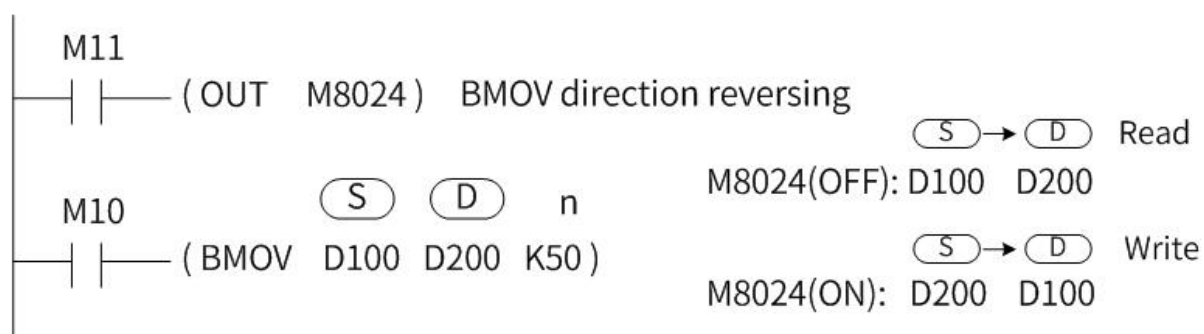
Operation flag: M8020 to M8022

The following lists actions of operation flags.

1. Zero flag: When the result value is 0, M8020 is ON.
2. Borrow flag: When the result value exceeds the minimum unit, M8021 is ON.
3. Carry flag: If the absolute value is out of range, M8022 is ON.

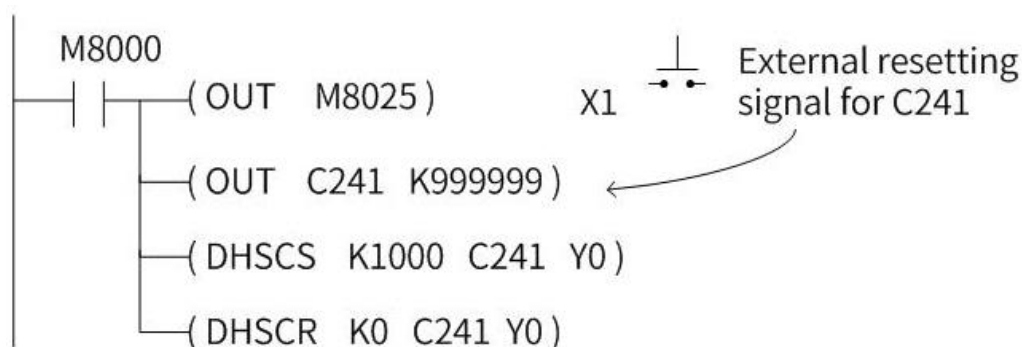
Instruction assignment commutation: M8024

Data can be transmitted in two directions by controlling the reverse flag M8024 of the BMOV instruction.



HSC instruction mode: M8025

High-speed counter output contacts and comparison outputs of FNC53 (D HSCS), FNC54 (D HSCR), and FNC55 (D HSZ) instructions change with the current value registers for counter inputs. Even if the current values (C235 to C255) are changed through the DMOV instruction, no comparison results will be output without counter inputs. As described above, a high-speed counter, for example, C241, has an external reset terminal (R), which executes instructions and outputs comparison results by resetting rising pulses of input signals. The following figure shows the external reset terminal used by C241 in external reset mode.



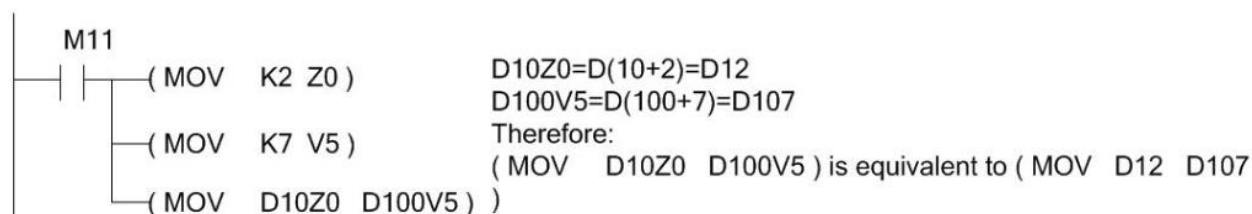
In the preceding example, when M8025 is ON, if the current C241 value is 2000, Y0 is ON. If the external reset button X1 is pressed, the current C241 value becomes 0, and Y0 is reset even if no counter inputs are generated on X0.

Execution complete flag: M8029

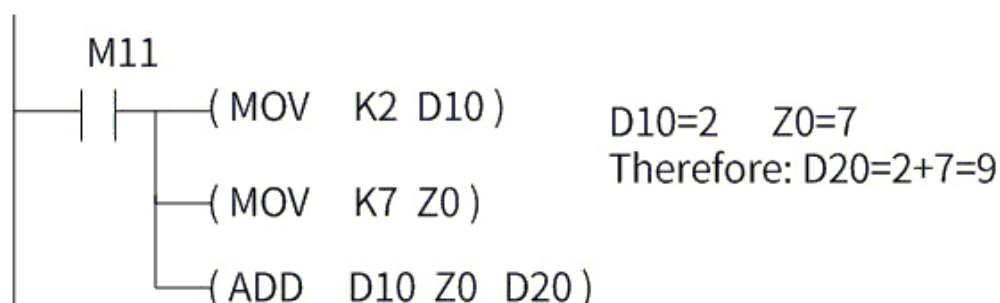
It is an instruction execution complete flag for high-speed pulse output, communication, MTR, HKY, DSW, SEGL, and PR.

Index register: D8028, D8029 and D8182 to D8195

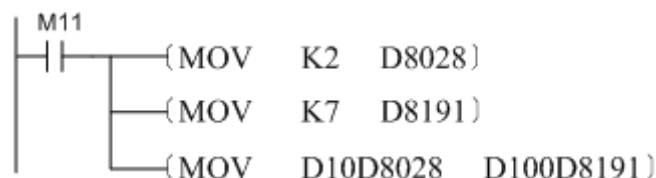
Variable-address registers not only support the functions of common data registers, but also support combined use with other element numbers or values in operands of application instructions. However, the soft element numbers for basic sequential control instructions (for example, LD, AND, and OUT) or step ladder diagram (LD) instructions cannot be used with index registers. Normally Z or V are used for index modification in programs.



Z and V can also be used as common registers.



There are special registers corresponding to Z and V. For example, the first program is equivalent to the following.



D8028 is Z0 register content; D8029 is V0 register content.  
 D8182 is Z1 register content; D8183 is V1 register content.  
 D8184 is Z2 register content; D8185 is V2 register content.  
 D8186 is Z3 register content; D8187 is V3 register content.  
 D8188 is Z4 register content; D8189 is V4 register content.  
 D8190 is Z5 register content; D8191 is V5 register content.  
 D8192 is Z6 register content; D8193 is V6 register content.  
 D8194 is Z7 register content; D8195 is V7 register content.

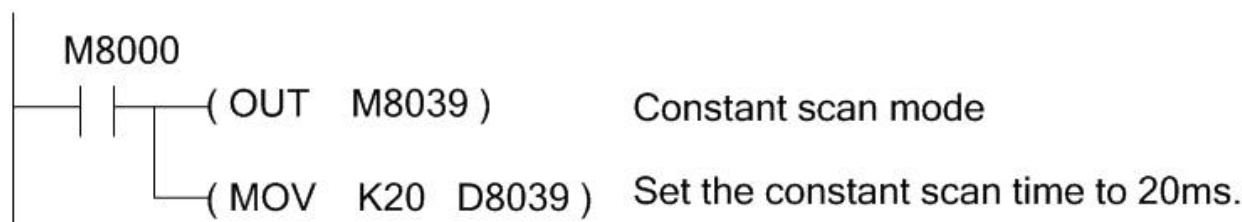
Bit status: M8031 to M8034

- 1) When M8031 is ON, all registers and relays not retentive at power failure are cleared.
  - Status for Y, general-purpose M, and general-purpose S contacts
  - Contact, and timing coils of general-purpose T
  - Contact, counting coils, and reset coils of general-purpose C
  - Current value registers of general-purpose D
  - Current value registers of general-purpose T
  - Current value registers of general-purpose C
- 2) When M8032 is ON, all registers and relays retentive at power failure are cleared.
  - Status of M and S contacts with support for the retentive at power failure feature
  - Contact and timing coils of cumulative timer T
  - Contact and counting coils of C with support for the retentive at power failure feature and high-speed counter C
  - Current value registers of D with support for the retentive at power failure feature
  - Current value registers of cumulative timer T
  - Current value registers of C with support for the retentive at power failure feature and high-speed counter C
- 3) When M8033 is ON, the device is stopped and all soft elements remain in the state before the device runs.
  - Status of all Y, M, and S contacts
  - Contact and timing coils of all T devices
  - Contact and counting coils of all C devices and high-speed counter C
  - Current value registers of all D, T, and C devices.
- 4) When M8034 is ON, all Y outputs are OFF.
- 5) You can set M8035 and M8036 ON through forced Run or Stop operation (in forced Run mode) to start the PLC.

6) You can set M8037 to ON (through forced Stop operation) to stop the PLC.

Constant scan mode: M8039 and D8039

Drive the auxiliary register M8039 and pre-write the target scan time (in the unit of 1ms) in the data register D8039 so that the PLC operation cycle will not be less than the target scan time. Even if an operation ends early, the system will not return to the origin until the scan time reaches the D8039 value. If the D8039 value is less than the actual program scan time, the actual program scan time prevails.



The RAMP, HKY, SEGL, ARWS, and PR instructions relating to the scan time must be used in fixed mode or used with timing insertion interrupts. In particular, the HKY instruction corresponds to 16-key input operation in 4x4 matrix form. The scan time must be greater than 20ms. The scan time indicated by D8010 to D8012 can be constant.

### Step ladder

Step: M8040 to M8049 and D8040 to D8049

- 1) After M8040 is driven, the status cannot be changed even if corresponding conditions are met. In addition, output operation continues when the device is stopped.
- 2) When step operation starts, M8041 is used as the IST instruction flag.
- 3) When pulses are started, M8042 is used as the IST instruction flag.
- 4) When zero return ends, M8043 is used as the IST instruction flag.
- 5) When the mechanical origin action is detected, M8044 is used as the IST instruction flag.
- 6) When output reset is disabled, M8045 is used as the IST instruction flag.
- 7) After M8047 is ON, when any of S0 to S999 is ON, M8046 is automatically connected as an action flag to avoid start in synchronization with other procedures or for process.
- 8) After M8049 is ON, when any of S900 to S999 is ON, M8048 is automatically connected as an action flag to avoid start in synchronization with other procedures or for process.
- 9) D8040 to D8047: The smallest 8 action address numbers of S0 to S999 are stored in D8040 to D8047.
- 10) When M8049 is ON, the smallest S900-S999 alarm address numbers are stored in D8049.

## Disable interrupt

Interrupt disabling: M8050 to M8059

After the interrupt function is disabled, no interrupts will be generated even if interrupt signals are transmitted. For example, when M8050 is ON, I00X will not be generated even if interrupt pulses are input on X0. Interrupts are defined in the following table.

Interrupt enable/disable setting				
M8050	Driver I00X interrupt disabling	Twelve X input interrupts correspond to rising and falling pulse interrupts of ports X0 to X5, respectively. 0 = rising pulse interrupt 1 = falling pulse interrupt	Each flag corresponds to control of one external interrupt. When an M flag is OFF, the corresponding X interrupt is enabled. When the M flag is ON, the X interrupt is disabled.	
M8051	Driver I10X interrupt disabling			
M8052	Driver I20X interrupt disabling			
M8053	Driver I30X interrupt disabling			
M8054	Driver I40X interrupt disabling			
M8055	Driver I50X interrupt disabling			
M8056	Driver I60X interrupt disabling			Timing Interrupt 0
M8057	Driver I70X interrupt disabling			Timing Interrupt 1
M8058	Driver I80X interrupt disabling			Timing Interrupt 2
M8059	Driver counter interrupt disabling	Six high-speed counter interrupts	When the flags are ON, interrupts I010 to I060 are disabled.	

However, the pulse capture function for X0 to X5 is not limited even if the interrupt function is disabled.

## Correction

Parallel connection protocol: M8070 to M873, M8162, and D8070

M8070 is set to a parallel master station and M8071 is set to a parallel slave station. M8070 and M8071 cannot be set simultaneously on one PLC; otherwise, the parallel connection protocol is invalid. If no other priority protocols exist, you can set D8126 to 50h so that it is a parallel master station or set D8126 to 5h so that it is a parallel slave station.

D8070: Sets the time for determination of communication errors. The default value is 500.

M8162: Indicates the high-speed parallel connection mode.

	Sent by Master Station (Received by Slave Station)	Sent by Slave Station (Received by Master Station)
Common mode M8162 = 0	M800 to M899 D490 to D499	M900 to M999 M50 to M509
High-speed mode M8162 = 1	D490 to D491	D500 to D501

Function application and setting are detailed in the parallel connection protocol for communication.

N:N protocol: D8076 to D8180, M8183 to M8191, and D8201 to D8218

Set one PLC as the N:N master station and multiple PLCs as N:N slave stations, and connect serial ports of all PLCs using N:N protocol. In this way, multiple PLCs can exchange data with each other.

D8176: Indicates the station number, ranging from 0 to 7, 0 indicating the master station.

D8177: Indicates the total number of slave stations, ranging from 1 to 7, used for the master station only.

D8178: Sets the refresh range (mode), ranging from 0 to 2, used for the master station only.

D8179: Sets the number of retries, used for the master station only.

D8180: Sets the communication timeout, in the unit of 10 ms, used for the master station only.

M8183 to M8190: Indicates the communication error flag, M8183 corresponding to Station 0 (master station), M8184 corresponding to Station 1, and M8190 corresponding to Station 7.

M8191: Transmits data.

D8201: Monitors the connection scan time.

D8202: Monitors the maximum connection time.

D8203: Monitors the number of master station communication errors, stopping on count of 10,000.

D8204 to D8210: Monitor the number of communication errors for slave stations 1 to 7 respectively, stopping on count of 10,000.

D8211: Indicates the master station communication error code.

D8212 to D8218: Indicate the communication error codes for slave stations 1 to 7, respectively. Error codes indicated by D8211 to D8218 are defined in the "Description of Errors".

Function application and setting are detailed in the N:N protocol for communication.

### Special function

Input port initialization: M8085 to M8089

The following functions can be enabled through PLSY, PLSR, DRVI, and DRVA instructions by setting special flags M8085 to M8089 (corresponding to Y0 to Y4, respectively) to ON.

When special flags are set to ON, the system immediately executes the next pulse output instruction without disabling previous instruction.

## Output complete interrupt: M8090 to M8094

The following functions can be enabled through PLSY, PLSR, DRVI, and DRVA instructions by setting special flags M8090 to M8094 (corresponding to Y0 to Y4, respectively) to ON.

A pulse output complete interrupt can be enabled, as shown in the following table.

Port Number	Special Bit	User Interrupt
Y0	M8090	I502
Y1	M8091	I503
Y2	M8092	I504
Y3	M8093	I505
Y4	M8094	I506

Function setting is detailed in the description of H2U special functions.

## Acceleration/deceleration time: M8135 to M8139, D8104 to D8108, and D8165 to D8169

The following functions can be enabled through PLSY, PLSR, DRVI, and DRVA instructions by setting special flags M8135 to M8139 (corresponding to Y0 to Y4, respectively) to ON.

You can change the number of output pulses (by increasing or decreasing pulses only during acceleration or when the speed is constant).

For PLSR, DRVI, and DRVA instructions, the deceleration time is defined by the following registers.

Port Number	Special Bit	Register
Y0	M8135	D8165
Y1	M8136	D8166
Y2	M8137	D8167
Y3	M8138	D8168
Y4	M8139	D8169

When M8135 is ON, Y0 deceleration time is determined by D8165, 100ms by default. The other coils are arranged in the same way.

For DRVI and DRVA instructions, the acceleration time is defined by the following registers.

Port Number	Special Bit	Register
Y0	M8135	D8104
Y1	M8136	D8105
Y2	M8137	D8106
Y3	M8138	D8107
Y4	M8139	D8108

When M8135 is ON, Y0 acceleration time is determined by D8104, 100 ms by default. The other coils

are arranged in the same way.

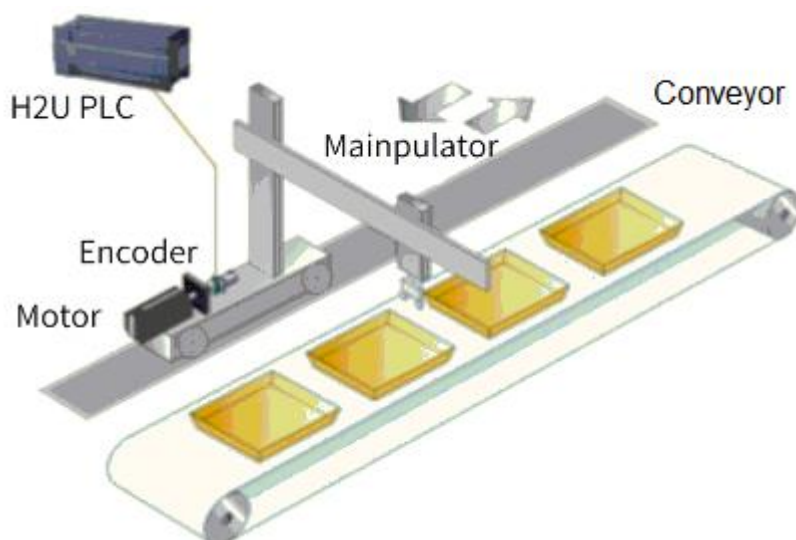
Function setting is detailed in the description of H2U special functions.

High-speed counter multi-user interrupt: M8084 and D8084 to D8086

To support multiple high-speed free tasks during running of a high-speed counter, you can enable any of the high-speed counters (C235 to C255) to generate multiple interrupts. This function is called the high-speed counter multi-user interrupt. A maximum of 24 interrupts are supported.

Flag	Description
M8084	If it is set to ON, the high-speed counter multi-user interrupt function is enabled.
D8084	Indicates high-speed counters C235 to C255.
D8085	Indicates the number of user interrupts. The maximum value is 24, ranging from I507 to I530.
D8086	Indicates the serial numbers of multiple comparative values and is only applicable to the D component. For example, 200 is a double word from head address D200.

For example, to enable vehicles to quickly execute loading/unloading tasks on different positions of the track, you can enable the high-speed counter multi-user interrupt function, which features quick response and easy operation.



Function setting is detailed in the description of H2U special functions.

Length counter and speed meter: M8100 to M8105

In an SPD instruction, if the function enabling flags M8100 to M8105 (corresponding to X0 to X5, respectively) are ON, the definition of "D" (SPD S1 S2 and D) is different from the original instruction.

When M8100 to M8105 are OFF:

D+0 indicates the number of pulses within S2, which is 16-bit data. D+1 indicates pulse counts within the period

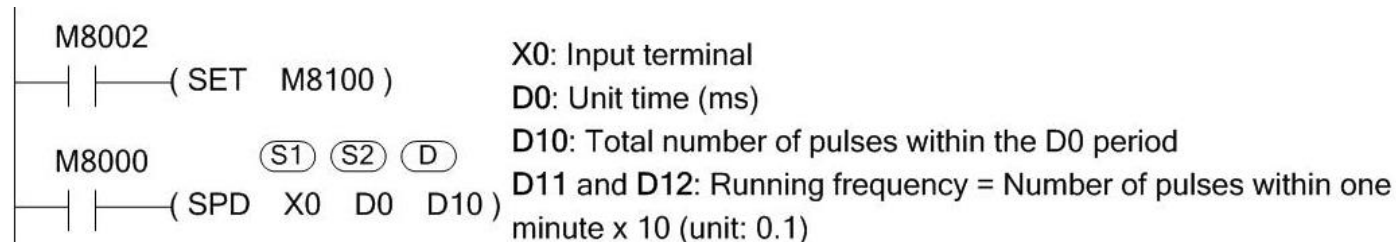


of time. D2+2 is used to measure the remaining time (ms).

When M8100 to M8105 are ON:

D+0 indicates the number of pulses within S2, which is 16-bit data. D+1 and D+2 indicate the number of pulse per minute, which is 32-bit data.

Example:



Function setting is detailed in the description of H2U special functions.

Output port clearing: M8140 and D8156 to D8160

When M8140 is ON, after the speed returns to the origin through the ZRN instruction, output ports corresponding to D8156 to D8160 are ON in one scan cycle for servo clearing.

D8156: Definition of Y0 clearing signals (by default, 5 = Y05)

D8157: Definition of Y1 clearing signals (by default, 6 = Y06)

D8158: Definition of Y2 clearing signals (by default, 7 = Y07)

D8159: Definition of Y3 clearing signals (by default, 8 = Y10)

D8160: Definition of Y4 clearing signals (by default, 9 = Y11)

**High-speed  
counter**

High-speed ring counter: M8099 and

After the special data register D8099 is driven on M8099, D8099 counts time in the unit of 0.1ms cumulatively.

When the D8099 value is greater than 32767, D8099 counts from 0. You can measure the pulse width in the unit of 1ms or 0.1ms by using a cumulative 1ms timer or special data register D8099 (high-speed ring counter).

## Communication

### COM0: D8110 to D8119

COM0 Protocol	D8116 Setting	Half-duplex/full-duplex mode	Communication format
Download protocol/HMI monitor protocol	01h	Depending on a jumper	Fixed to "7E1"
MODBUS-RTU slave/Qlink protocol	02h	Half-duplex	Depending on D8110
Other protocols (including RS instruction)	Not supported		

### COM1: M8120 to M8129 and D8120 to D8129

COM1 Protocol	D8126 Setting	Half-duplex/full-duplex mode	Communication format
RS command	00h	Determined by bit 10 of D8120	Determined by D8120
HMI monitor protocol	01h	Half-duplex	Fixed
Parallel slave station	50h	Half-duplex	
Parallel master station	05h	Half-duplex	
N:N protocol master station	40h	Half-duplex	
N:N protocol slave station	04h	Half-duplex	
Computer link protocol	06h	Half-duplex	
MODBUS-RTU slave/Qlink protocol	02h	Half-duplex	Determined by D8120
MODBUS-ASC slave station	03h	Half-duplex	
RS command	10h	Determined by bit 10 of D8120*	
Modbus-RTU Instruction	20h	Half-duplex	
MODBUS-ASC command	30h	Half-duplex	

\*Whether the RS instruction works in half-duplex or full-duplex mode is determined by Bit 10 of D8120.

1: Half-duplex RS-485 (standard ports to be used)

0: Full-duplex RS-232C/RS-422 (expansion board H2U-232-BD or H2U-422-BD to be used)

M8120	Reserved	D8120	Communication format, set on the user interface (UI), 0 by default
M8121	Sending (RS)	D8121	Station number, set on the UI, 1 by default
M8122	Flag of sending (RS) Execution state (Modbus)	D8122	Volume of residual data transmitted (for RS instruction only)
M8123	Flag of receipt (RS) Communication error flag (Modbus)	D8123	Volume of received data (for RS instruction only)
M8124	Receiving (RS)	D8124	Start of Text (STX) (for RS instruction only)

M8125	Reserved	D8125	End of Text (ETX) (for RS instruction only)
M8126	485BD expansion card is valid when ON	D8126	Communication protocol, set on the UI, 0 by default
M8127	Reserved	D8127	PC link protocol - head data address required
M8128	Reserved	D8128	PC link protocol - volume of sent data required
M8129	Timeout criterion	D8129	Communication timeout criterion, set on the UI, 10 (100ms) by default

Note: Please refer to "Chapter 5 Communication Function" for the M and D settings of COM2 and COM3.

#### Difference between COM0 and COM1 in function

COM0 is a standard RS-485 or RS-422 interface (an RS-485 interface is compatible with an RS-422 interface), which depends on a jumper. The interface terminal is an 8-hole mouse female connector.

COM0 422 supports program download and HMI monitoring protocol;

COM1 is an RS-485 interface, and the interface terminal is a wiring terminal.

The COM0 and COM1 RS-485 interface can serve as a master station or slave station. It supports the RS instruction, connection, and PC link protocol.

COM3 needs to expand the port through the expansion card.

#### Positioning

##### Positioning: M8145 to M8154 and D8136 to D8160

M8145: When you set M8145 to ON during pulse output, the process is immediately stopped on Y0. The PLSY and DRVI instructions will be executed to output pulses from scratch upon the next drive flow. The DRVA instruction will be executed to output remaining pulses upon the next drive flow.

M8146: When you set M8146 to ON during pulse output, the process is immediately stopped on Y1. The PLSY and DRVI instructions will be executed to output pulses from scratch upon the next drive flow. The DRVA instruction will be executed to output remaining pulses upon the next drive flow.

M8152: When you set M8152 to ON during pulse output, the process is immediately stopped on Y2. The PLSY and DRVI instructions will be executed to output pulses from scratch upon the next drive flow. The DRVA instruction will be executed to output remaining pulses upon the next drive flow.

M8153: When you set M8153 to ON during pulse output, the process is immediately stopped on Y3. The PLSY and DRVI instructions will be executed to output pulses from scratch upon the next drive flow. The DRVA instruction will be executed to output remaining pulses upon the next drive flow.

---

M8154: When you set M8154 to ON during pulse output, the process is immediately stopped on Y4. The PLSY and DRVI instructions will be executed to output pulses from scratch upon the next drive flow. The DRVA instruction will be executed to output remaining pulses upon the next drive flow.

M8147: When pulses are output on Y0, M8147 is ON; otherwise, M8147 is OFF. In this way, pulse output can be monitored for Y0.

M8148: When pulses are output on Y1, M8148 is ON; otherwise, M8148 is OFF. In this way, pulse output can be monitored for Y1.

M8149: When pulses are output on Y2, M8149 is ON; otherwise, M8149 is OFF. In this way, pulse output can be monitored for Y2.

M8150: When pulses are output on Y3, M8150 is ON; otherwise, M8150 is OFF. In this way, pulse output can be monitored for Y3.

M8151: When pulses are output on Y4, M8151 is ON; otherwise, M8151 is OFF. In this way, pulse output can be monitored for Y4.

D8136: Low word; D8137: High word

D8136 and D8137 are used as current value accumulation registers for Y0 and Y1 output positioning instructions. When DRVI and DRVA instructions are used, the current value is increased or decreased based on the rotation direction. In addition, as the same current value registers are used for PLSY and PLSR instructions that enable pulse output without direction signals, when the instructions are executed, the current value indicates the accumulated amount of output pulses.

D8140: Low word; D8141: High word

D8140 and D8141 are used as current value accumulation registers for the Y0 output positioning instruction. When DRVI and DRVA instructions are used, the current value is increased or decreased based on the rotation direction. In addition, as the same current value registers are used for PLSY and PLSR instructions that enable pulse output without direction signals, when the instructions are executed, the current value indicates the accumulated amount of output pulses.

D8142: Low word; D8143: High word

D8142 and D8143 are used as current value accumulation registers for the Y1 output positioning instruction. When DRVI and DRVA instructions are used, the current value is increased or decreased based on the rotation direction. In addition, as the same current value registers are used for PLSY and PLSR instructions that enable pulse output without direction signals, when the instructions are executed, the current value indicates the accumulated amount of output pulses.

D8150: Low word; D8151: High word

D8150 and D8151 are used as current value accumulation registers for the Y2 output positioning instruction. When DRVI and DRVA instructions are used, the current value is increased or decreased based on the rotation direction. In addition, as the same current value registers are used for PLSY and PLSR instructions that enable pulse output without direction signals, when the instructions are executed, the current value indicates the accumulated amount of output pulses.

D8152: Low word; D8153: High word

D8152 and D8153 are used as current value accumulation registers for the Y3 output positioning instruction. When DRVI and DRVA instructions are used, the current value is increased or decreased based on the rotation direction. In addition, as the same current value registers are used for PLSY and PLSR instructions that enable pulse output without direction signals, when the instructions are executed, the current value indicates the accumulated amount of output pulses.

D8154: Low word; D8155: High word

D8154 and D8155 are used as current value accumulation registers for the Y4 output positioning instruction. When DRVI and DRVA instructions are used, the current value is increased or decreased based on the rotation direction. In addition, as the same current value registers are used for PLSY and PLSR instructions that enable pulse output without direction signals, when the instructions are executed, the current value indicates the accumulated amount of output pulses.

D8145 indicates the base speed during execution of DRVI and DRVA instructions. For control of a step motor, you need to consider the resonance region and automatic start frequency of the step motor when setting the speed. The speed must be less than 10% of the maximum speed (D8147 and D8146). When the speed is out of range, it is automatically reduced to 10% of the maximum speed.

D8146: Low word; D8147: High word

D8146 and D8147 indicate the maximum speed during execution of DRVI and DRVA instructions. The frequency at which pulses are output on high-speed output ports must be less than the maximum speed. The frequency ranges from 10 to 100,000 Hz.

D8148 indicates the acceleration/deceleration time during execution of DRVI, DRVA, and PLSR instructions. The acceleration/deceleration time is the time it takes to reach the maximum speed (D8147 and D8146). Therefore, when the pulse output frequency is less than the maximum speed (D8147 and D8146), the actual acceleration/deceleration time will be shortened. The time ranges from 50 to 5,000ms.

M8135~M8139: (D8165~D8169)

M8135: Y0 port change pulse and deceleration time valid flag bit; by using this special bit, the number of pulses

output can be changed while the deceleration time can be individually modified. D8165 corresponds to the deceleration time of Y0.

M8136: Y1 port change pulse and deceleration time valid flag bit; by using this special bit, the number of pulses output can be changed while the deceleration time can be individually modified. D8166 corresponds to the deceleration time of Y0.

M8137: Y2 port change pulse and deceleration time valid flag bit; by using this special bit, the number of pulses output can be changed while the deceleration time can be individually modified. D8167 corresponds to the deceleration time of Y0.

M8138: Y3 port change pulse and deceleration time valid flag bit; by using this special bit, the number of pulses output can be changed while the deceleration time can be individually modified. D8168 corresponds to the deceleration time of Y0.

M8139: Y4 port change pulse and deceleration time valid flag bit; by using this special bit, the number of pulses output can be changed while the deceleration time can be individually modified. D8169 corresponds to the deceleration time of Y0.

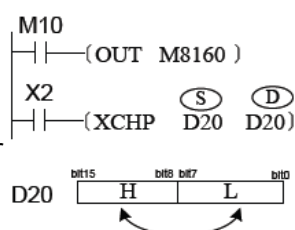
M8135~M8139 are only applicable to the following instructions: PLSY/PLSR/DRVI/DRVA.

## Data processing

### SWAP function of XCH: M8160

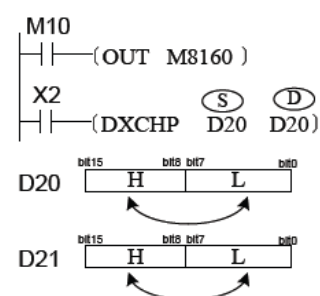
When the special variable M8160 is set to 1 and D, S share the same address, the upper eight bits and the lower eight bits are exchanged, which also applies to 32-bit instructions. That is, operations relating to the SWAP instruction is finished by executing the SWAP instruction.

In the figure on the left,  
the values of the upper 8 bits and  
the lower 8 bits of D20 are exchanged with each other



In the figure on the right,  
the values of the upper 8 bits and  
the lower 8 bits of D20 are exchanged.

The values of the upper 8 bits and  
the lower 8 bits of D21 are exchanged with each other,



**8-Bit and 16-bit modes: M8161**

The M8161 flag determines the variable width mode. When M8161 is OFF, the 16-bit mode is enabled. When M8161 is ON, the 8-bit mode is enabled so that the range of used variables is extended. ASC/RS/ASCII/HEX/CCD mode is used.

**Variable number of transmitted points: M8164 and D8164**

(FROM/TO) variable number of transmitted points: If M8164 is ON, when the FROM/TO instruction is executed, the data stored in special data register D8164 (specified for the number of transmitted points) are processed as the number of transmitted points n.

**Bit-word switchover: M8167**

When M8167 is set to ON, keys 0 to F are stored in hexadecimal format in D2 by enabling the HEY instruction.

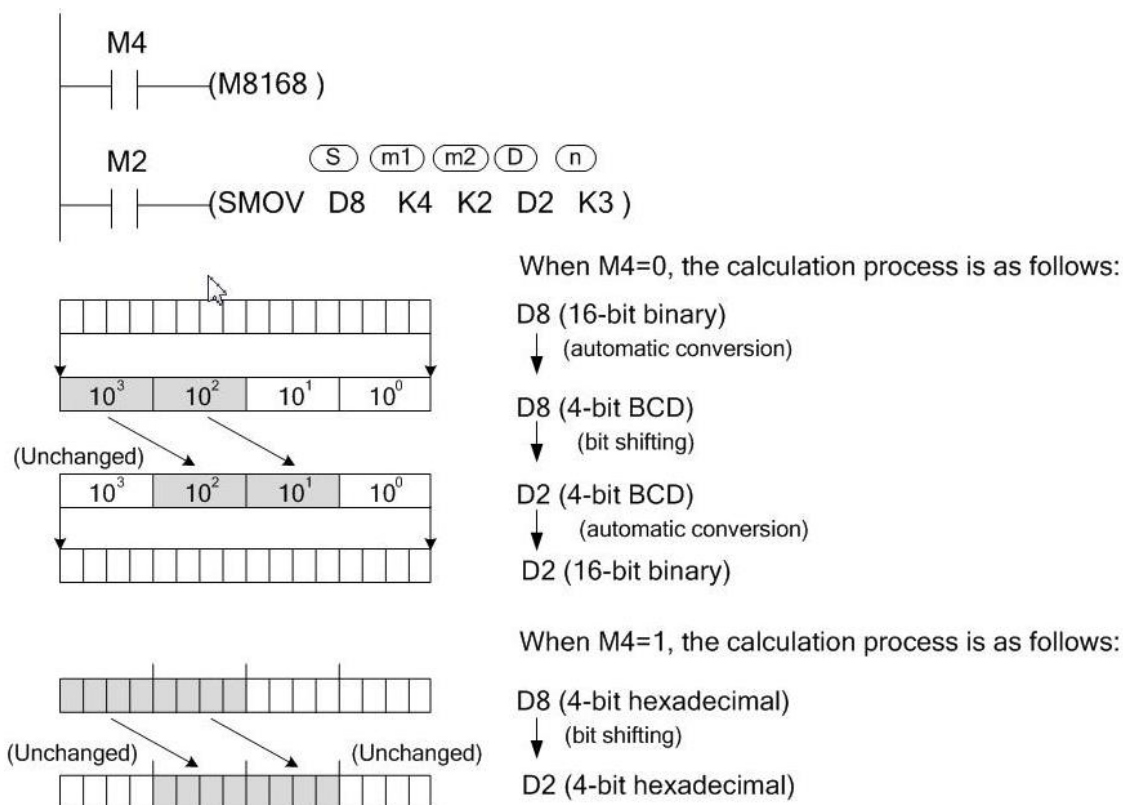
Example: When [123BF] is input, it is stored in BIN format in D2, which changes functions of A to F. For details, see the description of HEY instruction.

**SMOV decimal-hexadecimal switchover: M8168**

When M8168 is OFF, the BCD mode (decimal) is enabled. When M8168 is ON, the BIN mode is enabled, in which every four bits (hexadecimal) are transferred as one unit.

Assuming D8=K1234, D2=K5678, when M8168 is OFF (BCD mode), turn M2 ON, and the value of D2 becomes K5128;

When M8168 is ON (BIN mode), then D8=H04D2=K1234, D2=H162E=K5678, turn M2 ON, then D2=H104E=K4174.



## Pulse capture

Pulse capture: M8170 to M8175

M8170 X00 pulse capture

M8171 X01 pulse capture

M8172 X02 pulse capture

M8173 X03 pulse capture

M8174 X04 pulse capture

M8175 X05 pulse capture

To enable response to instant pulse signals on ports X0 to X5 without special requirements on the response time, you can enable pulse capture. The PLC saves rising pulse signals on ports X0 to X5 in units M8170 to M8175 as the basis for processing data in the main program and can be manually cleared after the response. After the "Enable Global Interrupts" EI instruction is executed, when X000 to X005 are switched from OFF to ON, special auxiliary registers M8170 to M8175 are set for interrupt processing. To enable inputs again, you need to reset elements through programs. The pulse capture action is unrelated to the actions of auxiliary registers for interrupt disabling M8050 to M8055; that is, pulse capture cannot be disabled by setting M8050 to M8055 to ON.



## High-speed input

### Table high-speed comparison mode: M8130, M8131, and D8130

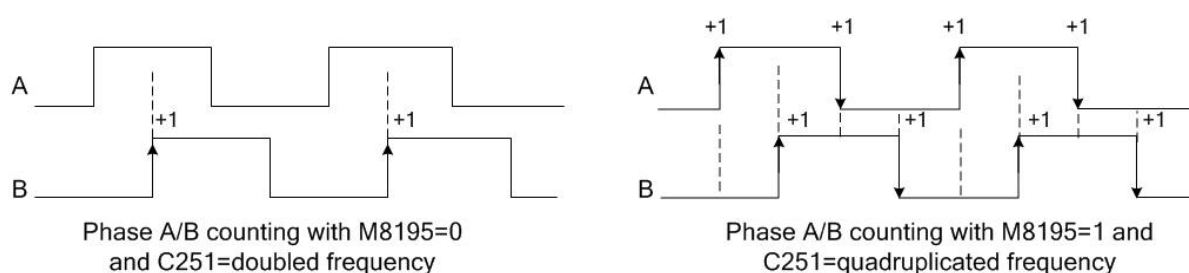
In table high-speed comparison mode, when data items in the first row are consistent with each other, the table counter D8130 value becomes 1 and the system starts the operation in the second row. After the operation in the last row is finished, actions of the flag M8131 are finished, and the operation in the first row is repeated.

Please refer to the specific application: HSZ instruction.

### Multiplication: M8195 to M8199

A/B phase high-speed counters T251 to T255 can count at a doubled or quadruplicated frequency, which is set by special registers M8195 to M8199. A/B phase high-speed counter signals occupy two pulse input ports and double the equivalent PLC pulses. If C251 to C255 count in A/B phase mode at a doubled frequency, the maximum high-speed input frequency is 50 kHz. If C251 to C255 count in A/B phase mode at a quadruplicated frequency, the software counting mode is enabled and the maximum high-speed input frequency is 25kHz.

When M8195 is OFF, C251 counts in A/B phase mode at a doubled frequency; when M8195 is ON, C251 counts in A/B phase mode at a quadruplicated frequency, as shown in the following figure.



Like C251,

When M8196 is OFF, C252 counts in A/B phase mode at a doubled frequency; when M8196 is ON, C252 counts in A/B phase mode at a quadruplicated frequency.

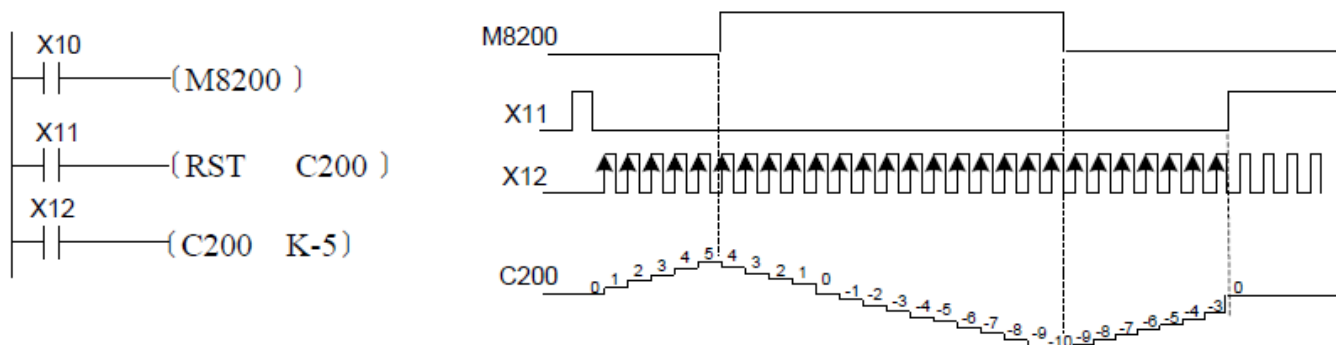
When M8197 is OFF, C253 counts in A/B phase mode at a doubled frequency; when M8197 is ON, C253 counts in A/B phase mode at a quadruplicated frequency.

When M8198 is OFF, C254 counts in A/B phase mode at a doubled frequency; when M8198 is ON, C254 counts in A/B phase mode at a quadruplicated frequency.

When M8199 is OFF, C255 counts in A/B phase mode at a doubled frequency; when M8199 is ON, C255 counts in A/B phase mode at a quadruplicated frequency.

## 32-bit counter increment/decrement: M8200 to M8234

For common 32-bit incremental/decremental counting, you can use the special auxiliary relays M8200 to M8234 to specify the counting direction. If M8XXX is ON, the counter CXXX counts decremental; if M8XXX is OFF, CXXX counts incremental.



When M8200 is OFF, C200 counts incremental; when M8200 is ON, C200 counts decremental.

Same as C200:

When M8201 is OFF, C201 counts incremental; when M8201 is ON, C201 counts decremental.

When M8202 is OFF, C202 counts incremental; when M8202 is ON, C202 counts decremental.

When M8203 is OFF, C203 counts incremental; when M8203 is ON, C203 counts decremental.

When M8204 is OFF, C204 counts incremental; when M8204 is ON, C204 counts decremental.

When M8205 is OFF, C205 counts incremental; when M8205 is ON, C205 counts decremental.

When M8206 is OFF, C206 counts incremental; when M8206 is ON, C206 counts decremental.

When M8207 is OFF, C207 counts incremental; when M8207 is ON, C207 counts decremental.

When M8208 is OFF, C208 counts incremental; when M8208 is ON, C208 counts decremental.

When M8209 is OFF, C209 counts incremental; when M8209 is ON, C209 counts decremental.

When M8210 is OFF, C210 counts incremental; when M8210 is ON, C210 counts decremental.

When M8211 is OFF, C211 counts incremental; when M8211 is ON, C211 counts decremental.

When M8212 is OFF, C212 counts incremental; when M8212 is ON, C212 counts decremental.

When M8213 is OFF, C213 counts incremental; when M8213 is ON, C213 counts decremental.

When M8214 is OFF, C214 counts incremental; when M8214 is ON, C214 counts decremental.

When M8215 is OFF, C215 counts incremental; when M8215 is ON, C215 counts decremental.

When M8216 is OFF, C216 counts incremental; when M8216 is ON, C216 counts decremental.

When M8217 is OFF, C217 counts incremental; when M8217 is ON, C217 counts decremental.

When M8218 is OFF, C218 counts incremental; when M8218 is ON, C218 counts decremental.

When M8219 is OFF, C219 counts incremental; when M8219 is ON, C219 counts decremental.

When M8220 is OFF, C220 counts incremental; when M8220 is ON, C220 counts decremental.

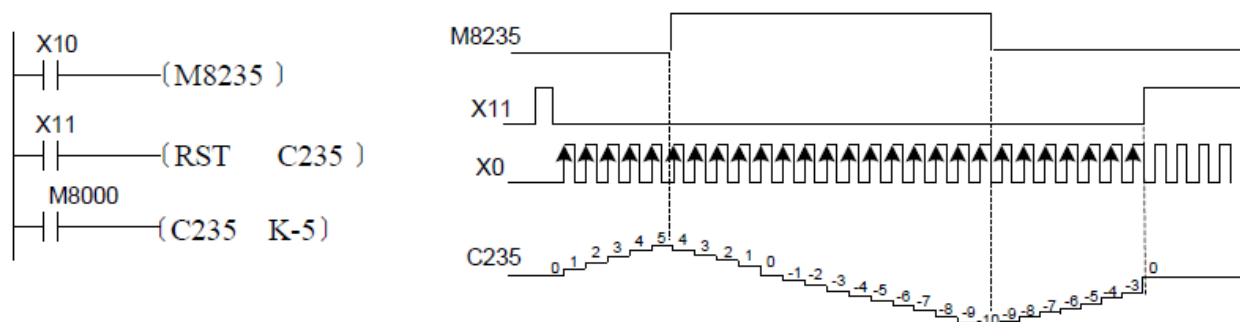
When M8221 is OFF, C221 counts incremental; when M8221 is ON, C221 counts decremental.

When M8222 is OFF, C222 counts incremental; when M8222 is ON, C222 counts decremental.

When M8223 is OFF, C223 counts incremental; when M8223 is ON, C223 counts decremental.  
 When M8224 is OFF, C224 counts incremental; when M8224 is ON, C224 counts decremental.  
 When M8225 is OFF, C225 counts incremental; when M8225 is ON, C225 counts decremental.  
 When M8226 is OFF, C226 counts incremental; when M8226 is ON, C226 counts decremental.  
 When M8227 is OFF, C227 counts incremental; when M8227 is ON, C227 counts decremental.  
 When M8228 is OFF, C228 counts incremental; when M8228 is ON, C228 counts decremental.  
 When M8229 is OFF, C229 counts incremental; when M8229 is ON, C229 counts decremental.  
 When M8230 is OFF, C230 counts incremental; when M8230 is ON, C230 counts decremental.  
 When M8231 is OFF, C231 counts incremental; when M8231 is ON, C231 counts decremental.  
 When M8232 is OFF, C232 counts incremental; when M8232 is ON, C232 counts decremental.  
 When M8233 is OFF, C233 counts incremental; when M8233 is ON, C233 counts decremental.  
 When M8234 is OFF, C234 counts incremental; when M8234 is ON, C234 counts decremental.

#### High-speed single-phase incremental/decremental counting: M8235 to M8245

For high-speed single-phase unidirectional counter inputs, only one count pulse signal input port is available. Special M registers determine the counting direction.



When M8235 is OFF, C235 counts incremental; when M8235 is ON, C235 counts decremental.  
 Same as C235:

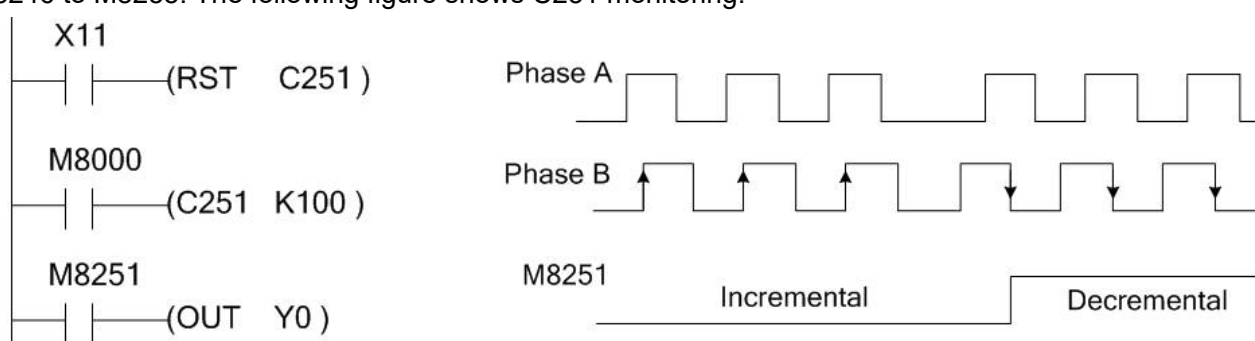
When M8236 is OFF, C236 counts incremental; when M8236 is ON, C236 counts decremental.  
 When M8237 is OFF, C237 counts incremental; when M8237 is ON, C237 counts decremental.  
 When M8238 is OFF, C238 counts incremental; when M8238 is ON, C238 counts decremental.  
 When M8239 is OFF, C239 counts incremental; when M8239 is ON, C239 counts decremental.  
 When M8240 is OFF, C240 counts incremental; when M8240 is ON, C240 counts decremental.  
 When M8241 is OFF, C241 counts incremental; when M8241 is ON, C241 counts decremental.  
 When M8242 is OFF, C242 counts incremental; when M8242 is ON, C242 counts decremental.  
 When M8243 is OFF, C243 counts incremental; when M8243 is ON, C243 counts decremental.  
 When M8244 is OFF, C244 counts incremental; when M8244 is ON, C244 counts decremental.  
 When M8245 is OFF, C245 counts incremental; when M8245 is ON, C245 counts decremental.  
 C241 to C245 support hardware reset inputs, and some of them support hardware start/stop inputs.

High-speed counter increment/decrement: M8246 to M8255

The system automatically controls the direction (increment/decrement) of high-speed single-phase bidirectional counting and high-speed A/B phase counting through external inputs.

For single-phase bidirectional counting, two count pulse signal input ports are available for incremental counter inputs and decremental counter inputs, respectively. Some counters have hardware reset and start/stop signal input ports. The A/B phase counting direction depends on the relationship between A phase and B phase: when A phase pulses are at high level, B phase rising pulses are counted incremental and falling pulses are counted decremental.

The counting direction (increment/decrement) can be monitored for C246 to C255 by reading the status of M8246 to M8255. The following figure shows C251 monitoring.



When C251 counts incremental, M8251 is OFF. When C251 counts decremental, M8251 is ON. You can judge whether C251 counts incremental or decremental by monitoring M8251.

When C246 counts incremental, M8246 is OFF. When C246 counts decremental, M8246 is ON. You can judge whether C246 counts incremental or decremental by monitoring M8246.

When C247 counts incremental, M8247 is OFF. When C247 counts decremental, M8247 is ON. You can judge whether C247 counts incremental or decremental by monitoring M8247.

When C248 counts incremental, M8248 is OFF. When C248 counts decremental, M8248 is ON. You can judge whether C248 counts incremental or decremental by monitoring M8248.

When C249 counts incremental, M8249 is OFF. When C249 counts decremental, M8249 is ON. You can judge whether C249 counts incremental or decremental by monitoring M8249.

When C250 counts incremental, M8250 is OFF. When C250 counts decremental, M8250 is ON. You can judge whether C250 counts incremental or decremental by monitoring M8250.

When C252 counts incremental, M8252 is OFF. When C252 counts decremental, M8252 is ON. You can judge whether C252 counts incremental or decremental by monitoring M8252.

When C253 counts incremental, M8253 is OFF. When C253 counts decremental, M8253 is ON. You can judge whether C253 counts incremental or decremental by monitoring M8253.

When C254 counts incremental, M8254 is OFF. When C254 counts decremental, M8254 is ON. You can judge whether C254 counts incremental or decremental by monitoring M8254.

When C255 counts incremental, M8255 is OFF. When C255 counts decremental, M8255 is ON. You can judge whether C255 counts incremental or decremental by monitoring M8255.



