VC600 drive and control integrated driver instructions

Directory

Chapter 1 Use and Description of software components

1.1 Description of devices

element	description	Internal start address of the drive	Drive internal end address	Common start address	common end address	Power-down save starting address	Power-down save end address
M	Auxiliary relay	0	511	512	3071	512	1535
C16 bit	Counter			0	199	100	199
C32 bit	High-speed counter			200	255	200	255
T	Timer			0	255	246	255
D	Data Register	0	2047	2048	7999	2048	3071
X	Input Relay			0	10		
Y	Output Relay			0	6		

Note: The following address is the PLC power-off saving parameter, which has been fixed internally and cannot be configured on the host computer

1.2 Address correspondence between devices and built-in PLC

X0-X9 correspond to DI1~DI9 of the drive
For example:
the function code of driver enable is InFn.01, and the bit address of the driver is:
40 (DI fixed offset address) + 01 (function code) = 41. Corresponding PLC M41. The driver is enabled as shown in the figure below.

Y1~Y9 correspond to DO1~DO6 of the drive
For example:
the function code in driver enable is OutFn.01, and the driver bit address is:

140 (DO fixed offset address) + 01 (function code) = 141. Corresponding PLC M141.

Note: DO bit addresses are read only.

M0~M512 correspond to the bit address 0~511 of the driver.

1.3 Parameter manipulation

D0~D2047 correspond to the parameters P00.00~P20.47 of the drive.

1.3.1 16-bit parameter read and write.

For example: set the drive parameter P04.03=888. As follows:

1.3.2 32-bit parameter read and write.

For example: set the drive parameter P13.10=500000. As follows:

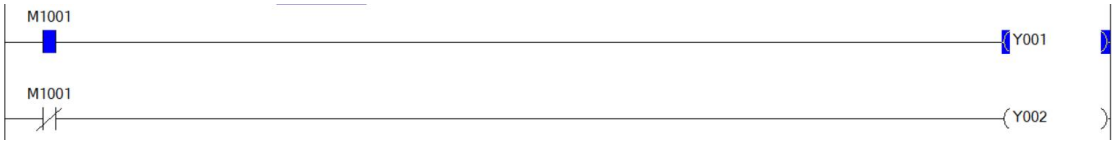
Chapter 2 Basic Commands

2.1 LD instruction

command parameter

Name	Input parameters	Output parameters	contact type
LD	-	-	A contact (normally open contact)
LDI	-	-	B contact (normally closed contact)

Graphical example:



Instruction Description:

Description
The logic operation of the normally open contact starts: the LD instruction is used to connect the contacts of the bus. The logic operation of the normally closed contact starts: the LDI instruction is used to connect the contacts of the bus.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1001	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	*	*

2.2 OUT instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
OUT[1]	-	-	The device programmed with the OUT instruction performs ON/OFF according to the status of the drive contact

Graphical example:

Instruction Description:

Description
Start coil: The OUT command is the coil drive command for output relays, auxiliary relays, status, timers, and counters, and is not used for input relays.

Available soft component:

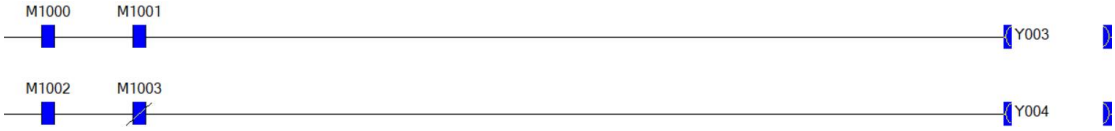
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	Y003	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	*	*

2.3 AND/ANI instruction

Instruction parameters

Name	Input parameters	Output parameters	type of data
AND	-	-	A contact (normally open contact)
ANI	-	-	B contact (normally closed contact)

Graphical example:



Instruction Description:

Description
Series connection of normally open contacts: The AND instruction performs a series connection of one contact. Series connection of normally closed contacts: The ANI instruction performs the series connection of 1 contact.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1001	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	*	*	

2.4 OR/OR Iinstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
OR	-	-	A contact (normally open contact)
ORI	-	-	B contact (normally closed contact)

Graphical example:

Instruction Description:

Description
Parallel connection of normally open contacts: OR is used as a parallel connection of 1 contact. Parallel connection of normally closed contacts: ORI is used as a parallel connection of 1 contact.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1000	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	*	*

2.5 LDP linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
LDP[1]	-	-	A contact (pulse rising edge detection)

Graphical example:

Instruction Description:

Description
Rising edge detection operation start: The LDP instruction is a contact instruction that performs rising edge detection, and is turned on for one operation cycle only at the rising edge of the specified bit device (when it changes from OFF to ON).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1000	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	*	*	

2.6 ORP linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
ORP[1]	-	-	A contact (pulse rising edge detection)

Graphical example:

Instruction Description:

Description
Rising edge detection parallel connection: The ORP instruction is a contact instruction that performs rising edge detection, and it is turned on for one operation cycle only when the specified bit device is on the rising edge (when it changes from OFF to ON).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1003	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	*	*	

2.7 ANDP linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
ANDF	-	-	A contact (pulse rising edge detection)

Graphical example:

Instruction Description:

Description
Rising edge detection series connection: The ANDP instruction is a contact instruction that performs rising edge detection, and it is turned on for one operation cycle only when the specified bit device is on the rising edge (when OFF→ON changes).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1002	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	*	*

2.8 LDF linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
LDF	-	-	A contact (pulse falling edge detection)

Graphical example:

Instruction Description:

Description
Falling edge detection operation start: The LDF instruction is a contact instruction that performs falling edge detection, and is turned on for one operation cycle only when the designated bit device falls on the falling edge (when it changes from ON to OFF).

Available soft component:

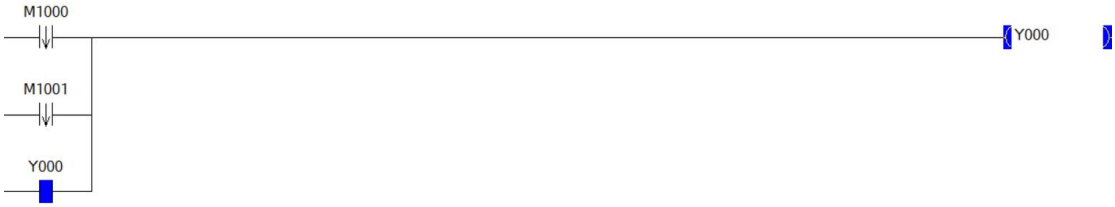
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1000	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	*	*

2.9 ORF linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
ORF[1]	-	-	A contact (pulse falling edge detection)

Graphical example:



Instruction Description:

Description
Falling edge detection parallel connection: The ORF instruction is a contact instruction that performs falling edge detection, and is turned on for one operation cycle only when the designated bit device falls on the falling edge (when it changes from ON to OFF).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1001	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	*	*	

2.10 ANDF linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
ANDF[1]	-	-	A contact (pulse falling edge detection)

Graphical example:

Instruction Description:

Description
Falling edge detection series connection: The ANDF instruction is a contact instruction that performs falling edge detection, and it is turned on for one

operation cycle only when the falling edge of the specified bit device (when ON →OFF changes).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1001	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	*	*

2.11 ORB linstruction

Instruction parameters

Name	Input parameters	Output parameters	Contact type
ORB	-	-	-

Graphical example:

Instruction Description:

Description
When connecting series circuit blocks in parallel, use the LD and LDI instructions at the start of the branch, and use the ORB instruction at the end of the branch.

2.12 INV linstruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
INV	-	-	The instruction to reverse the operation result

Graphical example:

Instruction Description:

Description
Inversion of the operation result: The INV instruction is an instruction to invert the operation result before the INV instruction, and it is not necessary to specify the element number

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C

2.13 MEP linstruction

Instruction parameters

Name	Input parameters	Output parameters	Instruction description
MEP	-	-	Turns ON (on state) at the rising edge (OFF →ON) of the operation result before the MEP instruction. When the operation result before the MEP instruction is not a rising edge, it is OFF (non-conduction state).

Graphical example:

Instruction Description:

Description
Operation Hold: Once the input condition is ON, the input condition will continue to operate even if it is turned OFF. Once the input condition is ON, the output will remain inactive even if it is turned OFF.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C

2.14 PLF linstruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
PLF	-	-	Differential output for falling edge

Graphical example:

Instruction Description:

Description
Differential output of falling edge: After using the PLF instruction, the object elements Y, M only run within one operation cycle after the drive input is turned OFF.

Available soft component:

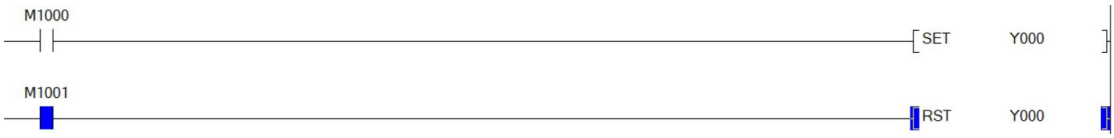
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	M1001	Bit	-	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

2.15 SET/RST Iinstruction

Instruction parameters

Name	Input parameters	Output parameters	Instruction description
SET	-	-	Action Hold: Once an input condition is ON, the input condition will continue to operate even if it is turned OFF. Once the input condition is turned on, the output will remain inactive even if it is turned off.
RST	-	-	Release action hold, clear current value and register: Once the input condition is ON, even if it is OFF, the input condition will continue to operate. Once the input condition is ON, the output will remain inactive even if it is turned OFF.

Graphical example:



Instruction Description:

Description
Action Hold: Once an input condition is ON, the input condition will continue to operate even if it is turned OFF. Once the input condition is turned on, the output will remain inactive even if it is turned off. Release action hold, clear current value and register: Once the input

condition is ON, the input condition will continue to operate even if it is OFF. Once the input condition is ON, the output will remain inactive even if it is turned OFF.

Available soft component:

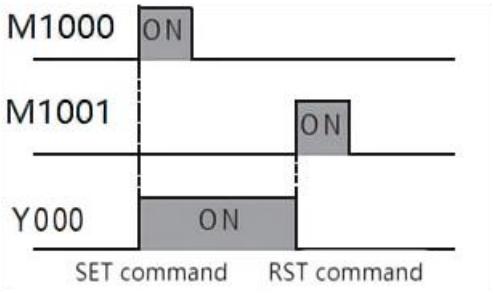
SET

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	Y000	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-

RST

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	Y000	bit /BIN16	-	*	*	*	*	-	*	*	*	-	-	-	-	-	-	-	*	*

Timing diagram:



Chapter 3 Program Flow

3.1 FNC 00 -CJ/CJP jump instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
CJ	-	-	Continuous jump when user program instruction is ON
CJP	-	-	When the user program instruction is ON, it jumps only once

Graphical example:

Instruction Description:

Description
Conditional jump: CJ, CJP instructions are instructions that make the part of the sequence program not executed. If there is this instruction, the operation cycle can be shortened and double coils can be used.

Available soft component:

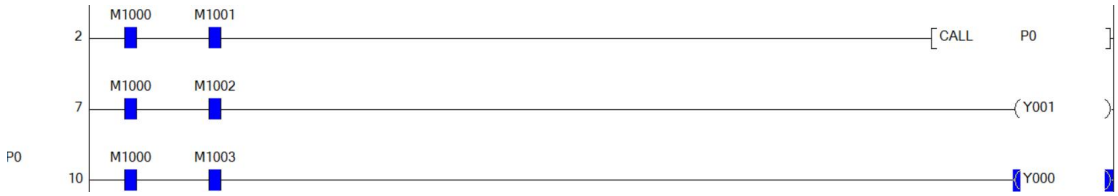
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
P	P0	name	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-

3.2 CALL/CALLP Subroutine call instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
CALL	-	-	Jumps are executed continuously when the user program instruction is ON
CALLP	-	-	Pulse execution jumps when user program command is ON

Graphical example:



Instruction Description:

Description
Subprogram call: When the input condition is ON, execute the CALL instruction and jump to the specified label step.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
P	P0	name	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-

Chapter 4 Transmission and Comparison

4.1CMP/CMPP - 16-bit compare instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
CMP	100	ON	Consecutively execute the comparison of the contents of code 1 and code 2
CMPP	100	ON	Pulse execution compares the content of code 1 and code 2

Graphical example:

Instruction Description:

Description
Compare: Compare the contents of source code 1 and source code 2, and perform corresponding actions according to the same size target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K100	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	C20	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	M1000	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–

4.2 DCMP/DCMPP - 32-bit compare instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DCMP	65540	ON	Consecutively execute the comparison of the contents of code

			1 and code 2
DCMPP	65540	ON	Pulse execution compares the content of code 1 and code 2

Graphical example:

Instruction Description:

Description
Compare: Compare the contents of source code 1 and source code 2, and perform corresponding actions according to their size and target.

Available soft component:

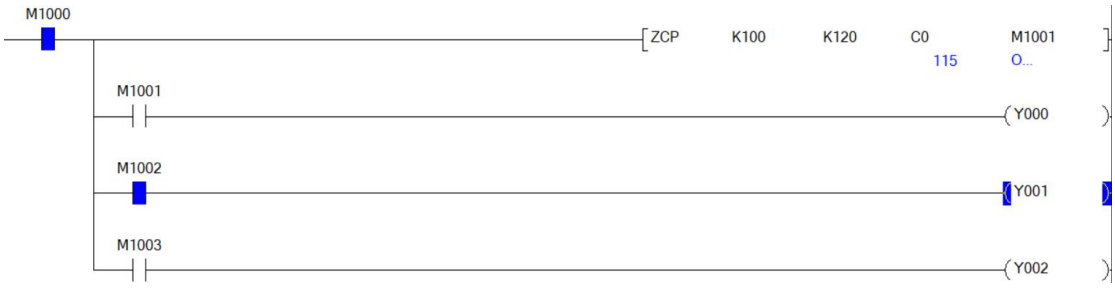
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K65540	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	C0	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	M1000	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-

4.3 ZCP/ZCPP - 16-bit interval comparison instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
ZCP	115	ON	The set value is continuously compared with the interval.
ZCPP	115	ON	Set value and interval pulse execution comparison size

Graphical example:



Instruction Description:

Description
Area comparison: Comparison of the size of the set values of 2 points.

Available soft component:

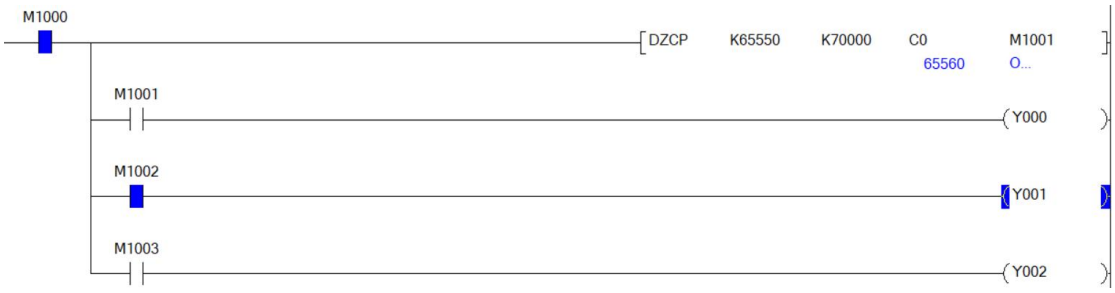
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K100	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	K120	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	C0	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	M1001	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–

4.4 DZCP/DZCPP linstruction -32-bit range comparison instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DZCP	115	ON	The set value is continuously compared with the interval.
DZCPP	115	ON	Set value and interval pulse execution comparison size

Graphical example:



Instruction Description:

Description
Area comparison: The size comparison of the set value of 2 points.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K65550	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
S	K70000	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
S	C0	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
D	M1001	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–

4.5 MOV/MOVP - 16-bit transfer instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
MOV	100	100	Consecutive execution transfers the content to the target address
MOVP	100	100	Pulse execution transfers the content to the target address

Graphical example:

Instruction Description:

Description
Transfer: Transfer the contents of the source code to the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K100	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D0	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

4.6 DMOV/DMOVP - 32-bit transfer instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DMOV	65590	65590	Consecutive execution transfers the content to the target address
DMOVP	65590	65590	Pulse execution transfers the content to the target address

Graphical example:

Instruction Description:

Description
Transfer: Transfer the contents of the source code to the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D3002	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

4.7 CML/CMLP -16-bit reverse transfer instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
CML	5000	-5001	Continuous execution inverts each bit of the data and transfers it to the destination address
CMLP	5000	-5001	Pulse execution inverts the data bits and transfers them to the target address

Graphical example:

Instruction Description:

Description
Reverse transmission: invert each bit of source code data (0 → 1,1 → 0) and then transmit it to the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D3001	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

4.8 DCML/DCMLP - 32-bit reverse transfer instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DCML	66633	-66634	Continuous execution inverts each bit of data and transfers it to the target address
DCMLP	66633	-66634	Pulse execution inverts each bit of data and transfers it to the target address

Graphical example:

Instruction Description:

Description
Reverse transfer: Invert each bit of the source code data (0→1, 1→0) and transfer it to the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D3002	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

4.9 BMOV/BMOVP batch transfer instructions

Instruction parameters

Name	Input parameters	Output parameters	instruction
BMOV	5000	5000	Continuous execution to transfer data in batches to the target address
BMOVP	5000	5000	Pulse execution transfers data in batches to the target address

Graphical example:

Instruction Description:

Description
Bulk transfer: batch transfer of n-point data starting with the element specified by the source code to n-point elements starting with the element specified by the target. (Transfer within possible range when the element number is out of range.)

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
D	D5000	BIN16	-	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
n	K10	BIN16	-	-	-	-	*	-	-	-	-	-	-	-	*	*	-	-	-	-

4.10 FMOV/FMOVP – 16-bit multipoint transmit command

Instruction parameters

Name	Input parameters	Output parameters	instruction
FMOV	2000	2000	Continuously executes the transfer of contents to the target address and n devices after the target address
FMOVP	2000	2000	Pulse execution transfers the content to the target address and n devices after the target address

Graphical example:

Instruction Description:

Description
Multipoint transfer: The content of the element specified by the source code is transferred to the element at n points starting from the element specified by the target. The contents of elements at n points are all the same. The target's element number is sent within the possible range when the element number is out of range.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D5000	BIN16	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
n	K10	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–

4.11 DFMOV/DFMOVP – 32-bit multipoint transmit command

Instruction parameters

Name	Input parameters	Output parameters	instruction
DFMOV	65590	65590	Continuously executes the transfer of contents to the target address and n devices after the target address
DFMOVP	65590	65590	Pulse execution transfers the content to the target address and n device addresses after the target address

Graphical example:

Instruction Description:

Description
Multipoint transfer: The content of the element specified by the source code is transferred to the element at n points starting from the element specified by the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
D	D5000	BIN32	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
n	K10	BIN32	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–

4.12 XCH/XCHP – 16-bit swap instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	Output parameters	instruction
XCH	3555	6000	6000	3555	Consecutive executions swap data in the two addresses
XCHP	3555	6000	6000	3555	Pulse execution will exchange data in two addresses

Graphical example:

Instruction Description:

Description
Replace: Perform data replacement between targets. When using continuous effective instructions, it is necessary to pay attention to data replacement in each operation cycle.

Available soft component:

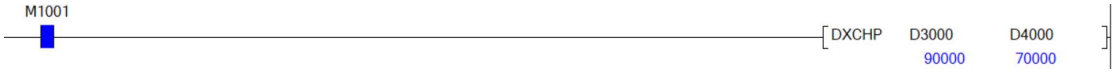
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*
D	D4000	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

4.13 DXCH/DXCHP – 16-bit swap instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	Output parameters	instruction
XCH	70000	90000	90000	70000	Sequential execution swaps data in two addresses
XCHP	70000	90000	90000	70000	Pulse execution swaps data in two addresses

Graphical example:



Instruction Description:

Description
Replace: Perform data replacement between targets. When using continuous effective instructions, it is necessary to pay attention to data replacement in each operation cycle.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*
D	D4000	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*

4.14 BCD/BCDP transform - 16 bit

Instruction parameters

Name	Input parameters	Output parameters	instruction
BCD	256	598	Convert decimal number to BCD code in continuous execution
BCDP	256	598	Pulse execution converts decimal number to BCD code

Graphical example:

Instruction Description:

Description
BCD conversion: source code (BIN) → target (BCD) conversion transfer instruction.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*
D	D4000	BCD4	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

4.15 BCD/DBCDP transform - 32bit

Instruction parameters

Name	Input parameters	Output parameters	instruction
DBCD	65579	598	Convert decimal number to BCD code in continuous execution
DBCDP	65579	598	Pulse execution converts decimal number to BCD code

Graphical example:

Instruction Description:

Description
BCD conversion: source code (BIN) → target (BCD) conversion transfer instruction.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*
D	D4000	BCD8	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

4.16 BIN/BINP transform – 32bit

Instruction parameters

Name	Input parameters	Output parameters	instruction
BIN	256	100	Convert decimal number to binary in continuous execution
BINP	256	100	Pulse execution to convert decimal number to binary

Graphical example:

Instruction Description:

Description
BIN conversion: source code (BCD) → target (BIN) conversion transfer instruction.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BCD4	*	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*
D	D4000	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

4.17 DBIN/DBINP transform – 32bit

Instruction parameters

Name	Input parameters	Output parameters	instruction
DBIN	68943	100	Convert decimal number to binary in continuous execution
DBINP	68943	100	Pulse execution to convert decimal number to binary

Graphical example:

Instruction Description:

Description
BIN conversion: source code (BCD) → target (BIN) conversion transfer instruction.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BCD8	*	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*
D	D4000	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*

Chapter 5 Four Logical Operations

5.1 ADD/ADDP – 16-bit Add Instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
ADD	30	50	80	Consecutively perform the addition of two numbers and transfer them to the destination address
ADDP	30	50	80	The pulse executes the addition of two numbers and transfers to the target address

Graphical example:

Instruction Description:

Description
BIN addition operation: perform binary addition operation on two source code data, and then transmit to the destination.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K30	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	K50	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D5	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

5.2 DADD/DADDP - 32-bit Add Instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DADD	65590	50	65640	Consecutively perform the addition of two numbers and transfer them to the destination address
DADDP	65590	50	65640	The pulse executes the addition of two numbers and transfers to the target address

Graphical example:

Instruction Description:

Description
BIN addition operation: perform binary addition operation on two source code data, and then transmit to the destination.

Available soft component:

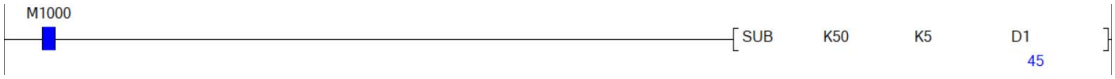
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K65590	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
S	K50	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
D	D5	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*

5.3 SUB/SUBP - 16-bit subtract instructions

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
SUB	50	5	45	Continuously perform subtraction of two numbers and transfer them to the target address
SUBP	50	5	45	The pulse performs the subtraction of two numbers and transfers to the destination address

Graphical example:



Instruction Description:

Description
BIN subtraction operation: perform an algebraic subtraction operation from the content specified by source code 1 to the element content specified by source code 2, and store the result into the element specified by the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K50	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	K5	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D1	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

5.4 DSUB/DSUBP - 32-bit subtract instructions

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DSUB	65590	50	65540	Continuously perform subtraction of two numbers and transfer to the destination address
DSUBP	65590	50	65540	The pulse performs the subtraction of two numbers and transfers to the destination address

Graphical example:

Instruction Description:

Description
BIN subtraction operation: perform an algebraic subtraction operation from the content specified by source code 1 to the element content specified by source code 2, and store the result into the element specified by the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K65590	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	K50	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D1	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

5.5 MUL/MULP – 16-bit multiply instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
MUL	50	50	2500	Continuously perform the multiplication of two numbers and transfer them to the target address
MULP	50	50	2500	The pulse performs the multiplication of two numbers and transmits it to the destination address

Graphical example:

Instruction Description:

Description
BIN multiplication operation: The product of the contents of the elements specified by each source code is stored as 32-bit data to the element (lower order side) and its (upper order side) specified by the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K50	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	K50	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D1	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

5.6 DMUL/DMULP – 32-bit multiply instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DMUL	10000	50	500000	Consecutively perform the multiplication of two numbers and transfer them to the destination address
DMULP	10000	50	500000	The pulse performs the multiplication of two numbers and transmits it to the destination address

Graphical example:

Instruction Description:

Description
BIN multiplication operation: The product of the contents of the elements specified by each source code is stored as 32-bit data in the element (lower order side) and its (upper order side) specified by the destination.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K10000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	K50	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D1	BIN64	-	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*

5.7 DIV/DIVP -16-bit divide instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DIV	9999	50	199	Continuously perform division of two numbers and transfer to the destination address
DIVP	9999	50	199	The pulse performs the division of two numbers and transfers to the destination address

Graphical example:

Instruction Description:

Description
BIN division operation: The content of the element specified by source code 1 is used as the dividend, the content of the element specified by source code 2 is used as the divisor, and the quotient and remainder are stored in the element specified by the target and its next numbered element.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K9999	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	K50	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D1	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

5.8 DDIV/DDIVP -32-bit divide instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DDIV	999999	50	19999	After successively performing the division of two numbers, transfer to the destination address
DDIVP	999999	50	19999	The pulse performs the division of two numbers and transfers to the destination address

Graphical example:

Instruction Description:

Description
BIN division operation: The content of the element specified by source code 1 is used as the dividend, the content of the element specified by source code 2 is used as the divisor, and the quotient and remainder are stored in the element specified by the target and its next numbered element.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K999999	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	K50	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D1	BIN64	-	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*

5.9 WAND/WANDP – 16-bit logical AND instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
WAND	3	50	2	Continuously perform the bitwise AND of two numbers and send them to the target address
WANDP	4	50	2	Pulse performs a bitwise AND of two numbers to the destination address

Graphical example:

Instruction Description:

Description
Logical AND: The corresponding logical AND operation can be performed on each bit.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K3	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	K50	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D1	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

5.10 DAND/DANDP – 32-bit logical AND instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DAND	65540	4564	2	Continuously perform bitwise AND of two numbers and send them to the destination address
DANDP	65540	4564	2	Pulse performs a bitwise AND of two numbers and sends them to the target address

Graphical example:

Instruction Description:

Description
Logical AND: The corresponding logical AND operation can be performed on each bit.

Available soft component:

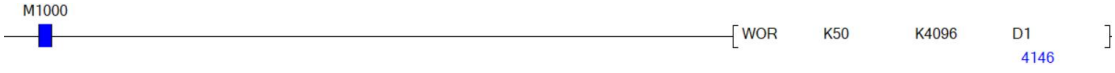
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K65540	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
S	K4564	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
D	D1	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*

5.11 WOR/WORP - 16-bit logical OR instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
WOR	50	4096	4146	Execute bitwise OR of two numbers consecutively and send them to the target address
WORP	50	4096	4146	Pulse performs a bitwise OR of two numbers and sends them to the destination address

Graphical example:



Instruction Description:

Description
Logical OR: The corresponding logical OR operation can be performed on each bit.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K50	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	K4096	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D1	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

5.12 DOR/DORP - 32-bit logical OR instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DOR	65590	4096	196612146	Execute the bitwise OR of two numbers consecutively and send them to the destination address
DORP	65590	4096	196612146	Pulse performs a bitwise OR of two numbers and sends them to the destination address

Graphical example:

Instruction Description:

Description
Logical OR: The corresponding logical OR operation can be performed on each bit.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K65590	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
S	K4096	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
D	D1	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*

5.13 WXOR/WXORP – 16-bit logical XOR instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
WXOR	1655	1376	791	Continuously execute the bitwise XOR of two numbers and send them to the target address
WXORP	1655	1376	791	The pulse performs the bitwise XOR of two numbers and sends them to the destination address

Graphical example:

Instruction Description:

Description
XOR : The corresponding XOR operation can be performed on each bit.

Available soft component:

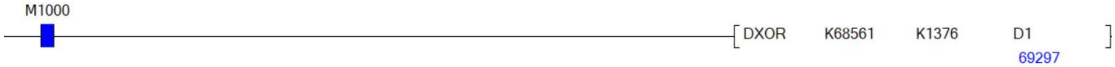
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K1655	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	K1376	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D1	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

5.14 DXOR/DXORP – 32-bit logical XOR instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DXOR	68561	1376	791	Continuously perform bitwise XOR of two numbers and send them to the target address
DXORP	68561	1376	791	The pulse performs the bitwise XOR of two numbers and sends them to the destination address

Graphical example:



Instruction Description:

Description
XOR: The corresponding XOR operation can be performed on each bit.

Available soft component:

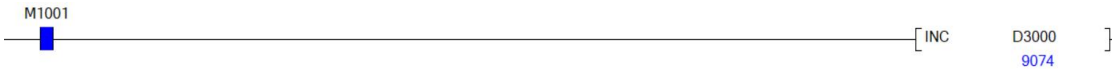
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K68561	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	K1376	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D1	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

5.15 INC/INCP – 16-bit increment instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
INC	0	9074	Continuously execute data increment by 1
INCP	0	9074	Pulse execution data auto increment by 1

Graphical example:



Instruction Description:

Description
BIN increment: Each time the input condition is ON, the content of the element specified by the target is incremented by 1.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

5.16 DINC/DINCP – 32-bit increment instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DINC	68972	68973	Continuously execute data increment by 1
DINCP	68972	68973	Pulse execution data auto increment by 1

Graphical example:

Instruction Description:

Description
BIN increment: Each time the input condition is ON, the content of the element specified by the target is incremented by 1.

Available soft component:

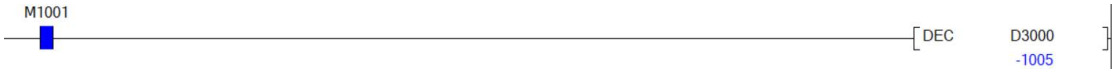
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

5.17 DEC/DECP – 16-bit decrement instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DEC	0	-1005	Continuously execute data decrement by 1
DECP	0	-1005	Pulse execution data is decremented by 1

Graphical example:



Instruction Description:

Description
BIN decrement: The content of the element specified by the target is decremented by 1 each time the input condition is ON.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

5.18 DDEC/DDECP – 32-bit decrement instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DDEC	65535	65534	Continuously execute data decrement by 1
DDECP	65535	65534	Pulse execution data is decremented by 1

Graphical example:

Instruction Description:

Description
BIN decrement: The content of the element specified by the target is decremented by 1 each time the input condition is ON.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*

5.19 NEG/NEGP – 16-bit complement instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
NEG	1165	-1165	Two's complement of data is executed continuously
NEGP	1165	-1165	Two's complement of pulse execution data

Graphical example:

Instruction Description:

Description
Complement number: After inverting each bit of the content of the element specified by the target (0→1, 1→0), the result obtained by adding 1 to the value is stored in the original element.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

5.20 NEG/NEGP – 16-bit complement instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DNEG	65536	-65536	Two's complement of consecutive execution data
DNEGP	65536	-65536	Two's complement of pulse execution data

Graphical example:

Instruction Description:

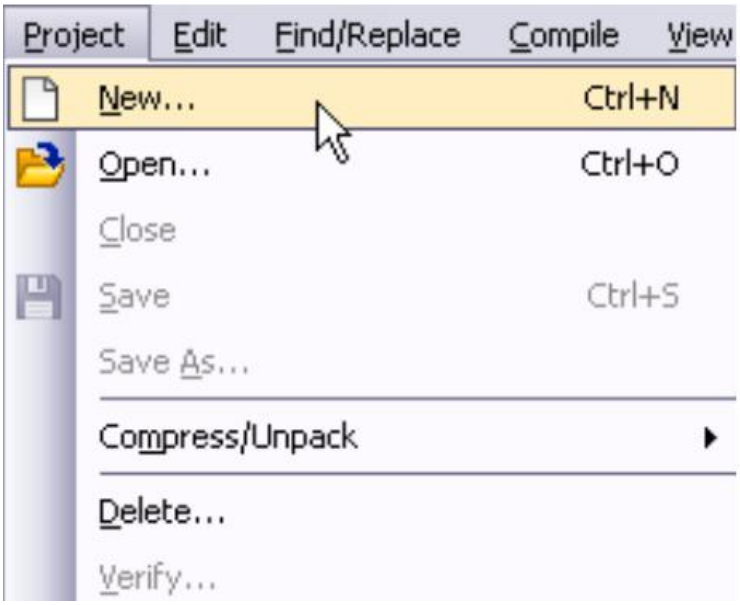
Description
Complement number: After inverting each bit of the content of the element specified by the target (0→1, 1→0), the result obtained by adding 1 to the value is stored in the original element.

Available soft component:

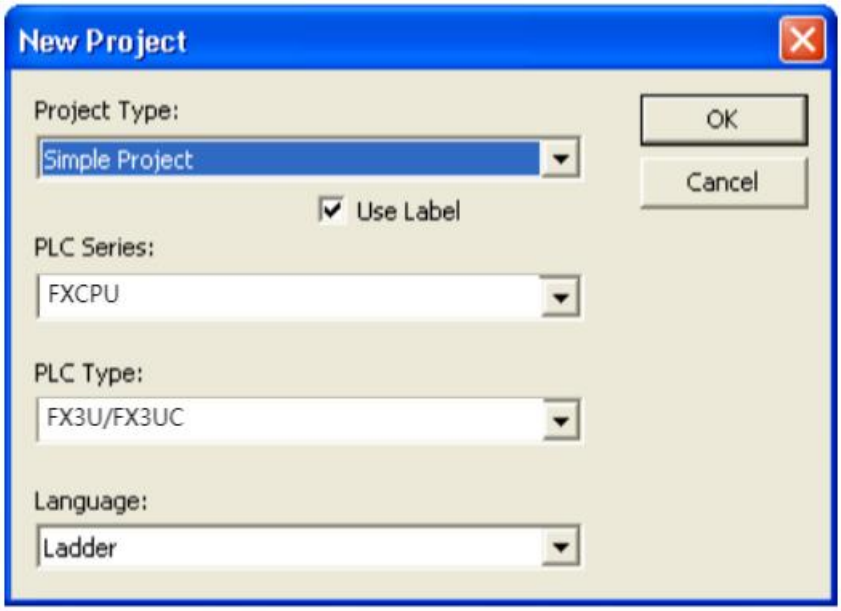
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

Chapter 6 Project Creation and Configuration

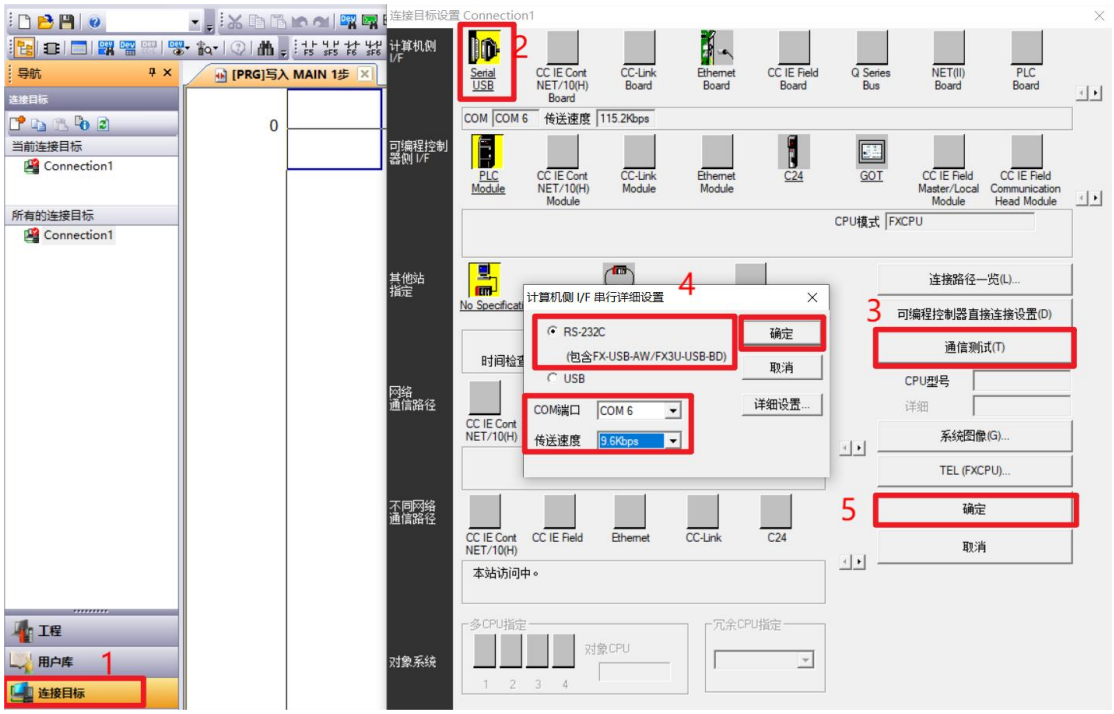
1. Click Project and select New Project



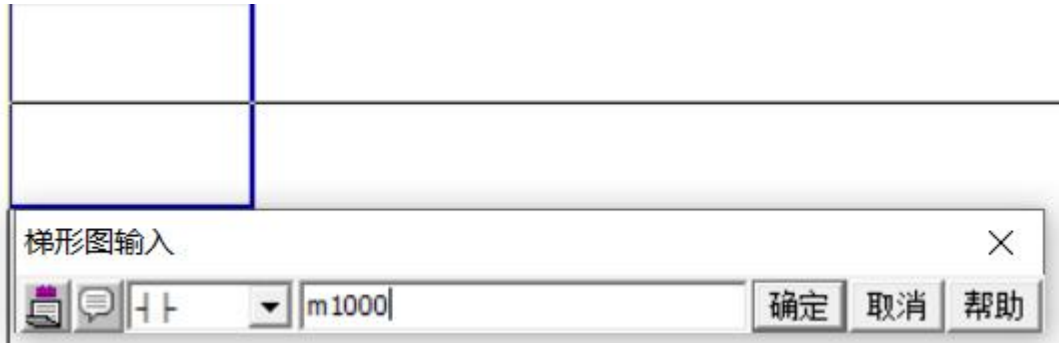
2. Select simple project as project type, select FXCPU as PLC series, select ladder diagram as program language, click OK and press



3. Click the connection target, double-click "Serial USB", select the RS-232C mode, configure the COM port number and the baud rate to 9.6Kbps, press the OK button, and click the communication test. After the communication test is successful, click the OK button, and the project is created.

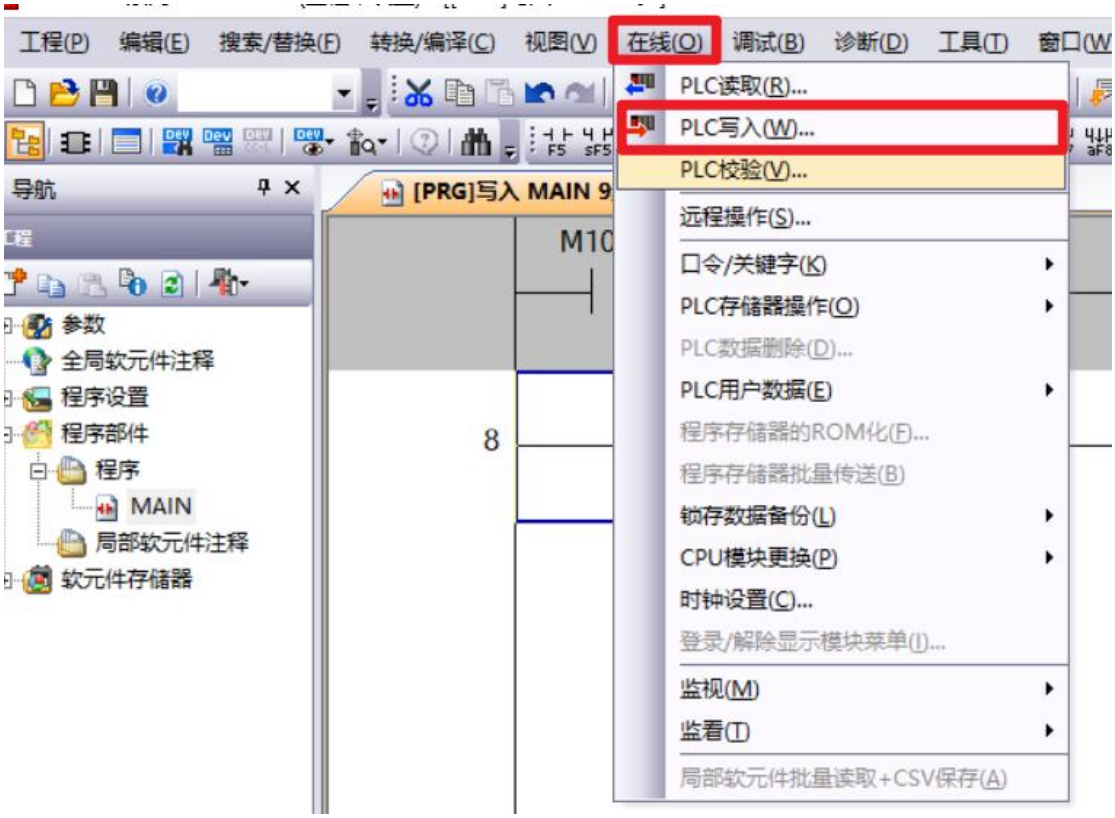


4. Enter the number of the software component and click OK.



5. transformation

6. Download to PLC


工程(P)
编辑(E)
搜索/替换(F)
转换/编译(C)
视图(V)
在线(O)
调试(B)
诊断(D)
工具(T)
窗口(W
PLC读取(R)...
PLC写入(W)...
PLC校验(V)...
远程操作(S)...
口令/关键字(K)
PLC存储器操作(O)
PLC数据删除(D)...
PLC用户数据(E)
程序存储器的ROM化(F)...
程序存储器批量传送(B)
锁存数据备份(L)
CPU模块更换(P)
时钟设置(C)...
登录/解除显示模块菜单(I)...
监视(M)
监看(T)
局部软元件批量读取+CSV保存(A)
导航
[PRG]写入 MAIN 9
参数
全局软元件注释
程序设置
程序部件
程序
MAIN
局部软元件注释
软元件存储器


读取(U)
写入(W)
校验(V)
删除(D)
CPU模块
执行对象数据的有无(无 / 有)
标题
编辑中的数据
参数+程序(P)
全选(A)
取消全选(N)
模块名/数据名
标题
对象
详细
更新时间
对象存储器
容量
我的第一个程序
PLC数据
程序存储器/软元...
程序(程序文件)
MAIN
2014/12/03 14:12:53
11 步
参数
PLC参数/网络参数
2014/12/03 14:12:52
全局软元件注释
COMMENT
详细
2014/12/03 14:15:32
软元件存储器
详细
MAIN
2014/12/03 14:12:55
必须设置(未设置 / 已设置)
必要时设置(未设置 / 已设置)
程序大小
11 步
8,000 步
更新为最新的信息(R)
关联功能(F)▲
执行(E)
关闭

Then wait until the PLC program is downloaded.

Chapter 7 Circular and Shift

7.1 ROR/RORP -16-bit rotate right instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
ROR	100	25	Continuously execute data right shift n bits
RORP	100	25	Pulse execution data is shifted right by n bits

Graphical example:

Instruction Description:

Description
Rotate Right: An instruction to rotate each bit of 16- or 32-bit data to the right.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*
n	K2	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.2 DROR/DRORP - 32-bit rotate right instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DROR	68976	17244	Continuously execute data right shift n bits
DRORP	68976	17244	Pulse execution data right shifted n bits

Graphical example:

Instruction Description:

Description
Rotate Right: An instruction to rotate each bit of 16- or 32-bit data to the right.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*
n	K2	BIN32	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

7.3 ROL/ROLP – 16-bit Rotate Left Instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
ROL	1234	4936	Continuously perform data left shift n bits
ROLP	1234	4936	Pulse execution data left shifted by n bits

Graphical example:

Instruction Description:

Description
Rotate Left: An instruction to rotate each bit of 16- or 32-bit data to the left.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*
n	K2	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

7.4 DROL/DROLP – 32-bit Rotate Left Instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
ROL	65535	262140	Continuously perform data left shift n bits
ROLP	65535	262140	Pulse execution data left shifted by n bits

Graphical example:

Instruction Description:

Description
Rotate Left: An instruction to rotate each bit of 16- or 32-bit data to the left.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*
n	K2	BIN32	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.5 RCR/RCRP – 16-bit Rotate Right with Carry

Instruction parameters

Name	Input parameters	Output parameters	instruction
RCR	2646	661	Continuously execute data right shift n bits
RCRP	2646	661	Pulse execution data is shifted right by n bits

Graphical example:

Instruction Description:

Description
Rotate Right with Carry: An instruction that rotates each bit of 16- or 32-bit data to the right.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*
n	K2	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.6 DRCR/DRCRP – 32-bit Rotation Right with Carry

Instruction parameters

Name	Input parameters	Output parameters	instruction
DRCR	65560	16390	Continuously perform data right shift by n bits

DRCRP	65560	16390	Pulse execution data is shifted right by n bits

Graphical example:

Instruction Description:

Description
Rotate Right with Carry: An instruction that rotates each bit of 16- or 32-bit data to the right.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*
n	K2	BIN32	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.7 RCL/RCLP – 16-bit Rotate Left with Carry

Instruction parameters

Name	Input parameters	Output parameters	instruction
RCL	1654	6616	Continuously execute data left shift n bits
RCLP	1654	6616	Pulse execution data left shifted by n bits

Graphical example:

Instruction Description:

Description
Rotate Left with Carry: An instruction to rotate each bit of 16- or 32-bit data to the left.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*
n	K2	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.8 DRCL/DRCLP – 32-bit circular left shift with carry

Instruction parameters

Name	Input parameters	Output parameters	instruction
RCL	65536	262144	Continuously perform data left shift n bits
RCLP	65536	262144	Pulse execution data left shifted by n bits

Graphical example:

Instruction Description:

Description
Rotate Left with Carry: An instruction to rotate each bit of 16- or 32-bit data to the left.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN32	–	*	*	*	*	*	*	–	*	–	–	–	–	–	–	–	*	*
n	K2	BIN32	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

7.9 SFTR/SFTRP – Bit right shift instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
SFTR	1	-	Continuous execution transfers n bits of the first data to the second data
SFTRP	1	-	Pulse execution transfers n bits of the first data to the second data

Graphical example:

Instruction Description:

Description
Bit right shift: executes the bit software components of n1 bits (length of shift register), and shifts left by n2 bits (executes the shift of n2 bits each time the instruction is executed).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1001	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	K1	BIN16	-	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-
n	K1	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.10 SFTL/SFTLP – bit shift left instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
SFTL	1	-	Continuous execution transfers n bits of the first data to the second data
SFTLP	1	-	Pulse execution transfers n bits of the first data to the second data

Graphical example:

Instruction Description:

Description
Bit left shift: executes the bit software components of n1 bits (the length of the shift register), and shifts it to the left by n2 bits (the shift of n2 bits is performed each time the instruction is executed).

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1001	Bit	*	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	K1	BIN16	-	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-
n	K1	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.11 WSFR/WSFRP -word right shift instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
SFTL	2000	100	1565	Continuously execute the software of n1 word length to right shift n2

				words
SFTLP	2000	100	1565	Pulse execution shifts a word of n1 word length to the right by n2 words

Graphical example:

Instruction Description:

Description
Word right shift: Execute the command of right shift n2 words for word devices of n1 words in word units. (n2<=n1<=512)

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
D	D4000	BIN16	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
n	K16	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–
n	K4	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

7.12 WSFL/WSFLP -word right shift instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
WSFL	2000	100	2000	Continuously execute the software to shift the word of n1 word length to the left by n2 words
WSFLP	2000	100	2000	Pulse execution software shifts n1 words to the left by n2 words

Graphical example:

Instruction Description:

Description
Word left shift: Execute the command of left shift n2 words for word devices of n1 words in word units. (n2<=n1<=512)

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
D	D4000	BIN16	-	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
n	K16	BIN16	-	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-
n	K4	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

7.13 SFWR/SFWRP -shift write instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
SFWR	2356	8	7	Continuous execution writes data to the target address + 1
SFWRP	2356	8	7	Pulse execution writes data to target address + 1

Graphical example:

Instruction Description:

Description
Shift Write: A data write instruction used to control first-in, first-out.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D4000	BIN16	-	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
n	K8	BIN16	-	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-

7.14 SFRD/SFRDP -shift readout

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
SFWR	2356	8	4464	Continuous execution reads data into the target address
SFWRP	2356	8	4464	Pulse execution reads data into the target address

Graphical example:

Instruction Description:

Description
Shift Read: A data read instruction used to control first-in, first-out.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
D	D4000	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*
n	K8	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–

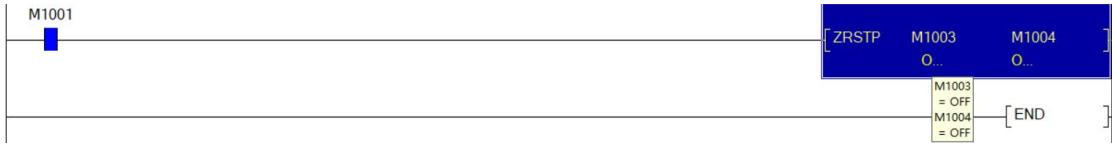
Chapter 8 Data Processing

8.1 ZRST/ZRSTP – Batch reset instructions

Instruction parameters

Name	Input parameters	Output parameters	instruction
ZRST	-	-	Continuously perform reset software components
ZRSTP	-	-	Pulse execution reset software components

Graphical example:



Instruction Description:

Description
Batch reset: target 1 and target 2 of the software components, to perform a batch reset. (D1 number <= D2 number)

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	M1003	bit /BIN16	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
D	M1004	bit /BIN16	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*

8.2 DECO/DECOP -decode instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DECO	100	-	Continuous execution decodes data and embodies it in software components
DECOP	100	-	Pulse execution decodes data and embodies it in software components

Graphical example:

Instruction Description:

Description
Decoding: The lower n bits (n<=4) of the source code are decoded by the target. When n<=3, the high bits of the target are all 0.

Available soft component:

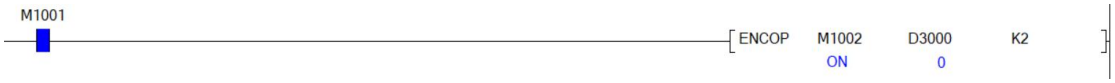
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	bit /BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	M1002	bit /BIN16	-	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
n	K2	BIN16	-	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-

8.3 ENCO/ENCOP -Coded instructions

Instruction parameters

Name	Input parameters	Output parameters	instruction
ENCO	132	-	Continuous execution encodes data and embodies it in software components
ENCOP	132	-	Pulse execution encodes data and embodies it in software components

Graphical example:



Instruction Description:

Description
Encoding: Encode the 2 n times the bearing data in the source code and store it in the target file. The last bit is ignored when the complex bit in the source code is 1. Or an operation error occurs when the source code is all 0s.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	M1002	bit /BIN16	*	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*
D	D3000	BIN16	-	-	-	-	*	*	*	*	*	-	-	-	-	-	-	-	*	*
n	K2	BIN16	-	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-

8.4 SUM/SUMP -16-bit ON number of digits

Instruction parameters

Name	Input parameters	Output parameters	instruction
SUM	21847	9	Continuously execute the ON command in the calculation data
SUMP	21847	9	ON instruction in the pulse execution calculation data

Graphical example:

Instruction Description:

Description
ON number of digits: store the number of 1s in the source code into the target

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K21847	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	C0	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

8.5 DSUM/DSUMP -32-bit ON number of digits

Instruction parameters

Name	Input parameters	Output parameters	instruction

DSUM	65597	6	Continuously execute the ON command in the calculation data
DSUMP	65597	6	ON instruction in the pulse execution calculation data

Graphical example:

Instruction Description:

Description
ON number of digits: store the number of 1s in the source code into the target

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K21847	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	C0	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*

8.6 BON/BONP – 16-bit NO bit judgment

Instruction parameters

Name	Input parameters	Output parameters	instruction
BON	50	-	Continuously execute n-position commands of judgment data
BONP	50	-	n-position command of pulse execution judgment data

Graphical example:

Instruction Description:

Description
ON bit judgment: When the nth bit in the source code is 1 (ON), the target is running.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K50	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	K1	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

8.7 DBON/DBONP – 32-bit NO bit judgment

Instruction parameters

Name	Input parameters	Output parameters	instruction
DBON	279314482	-	Continuously execute n-position commands of judgment data
DBONP	279314482	-	n-position command of pulse execution judgment data

Graphical example:

Instruction Description:

Description
ON bit judgment: When the nth bit in the source code is 1 (ON), the target is running.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K279314482	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	K1	BIN32	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

8.8 MEAN/MEANP – 16-bit average Instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
MEAN	500	250	Continuously execute the average value instruction of data
MEANP	500	250	Pulse execution to take the average value of data instruction

Graphical example:

Instruction Description:

Description
average value: Save the mean of the source code of n points (divide the algebraic sum by n) in the destination. Round off the remainder. When the element number overflows, treat n as a small value as much as possible within the allowable range.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	T1	BIN16	*	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
D	C5	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*
n	K2	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

8.9 DMEAN/DMEANP – 32-bit average value instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
MEAN	65597	32798	Continuously execute the average value instruction of data
MEANP	65597	32798	Pulse execution to take the average value of data instruction

Graphical example:

Instruction Description:

Description
average value: Save the mean of the source code of n points (divide the algebraic sum by n) in the destination. Round off the remainder. When the element number overflows, treat n as a small value as much as possible within the allowable range.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	-	-	-	-	-	-	-	-	-	*	*
D	D3002	BIN32	-	*	*	*	*	*	*	-	*	-	-	-	-	-	-	-	*	*
n	K2	BIN32	-	-	-	-	*	-	*	-	-	-	-	-	*	*	-	-	-	-

8.10 SQR/SQRP – 16-bit square root operation

Instruction parameters

Name	Input parameters	Output parameters	instruction
SQR	120	10	Continuously execute the square root instruction to fetch data

SQRP	120	32798	Pulse execution to fetch the square root instruction of data

Graphical example:

Instruction Description:

Description
BIN Square Root: An instruction that performs square root (root) operations.

Available soft component:

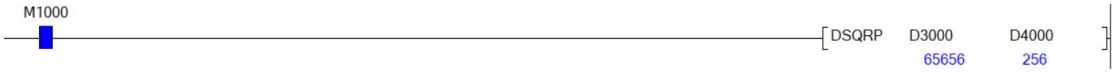
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	*	*	–	–	–	–
D	D4000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

8.11 DSQR/DSQRP – 32-bit square root operation

Instruction parameters

Name	Input parameters	Output parameters	instruction
SQR	65656	256	Continuously execute the square root instruction to fetch data
SQRP	65656	256	Pulse execution to fetch the square root instruction of data

Graphical example:



Instruction Description:

Description
BIN Square Root: An instruction that performs square root (root) operations.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	–	–	–	–	*	*	*	–	–	–	–	–	*	*	–	–	–	–
D	D4000	BIN32	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

8.12 FLT/FLTP – 16-bit binary floating point number conversion

Instruction parameters

Name	Input parameters	Output parameters	instruction
FLT	2000	2000.000	Converting data to floating point is executed continuously
FLTP	2000	2000.000	Pulse execution converts data to float

Graphical example:

Instruction Description:

Description
BIN integer→binary floating point conversion: conversion instruction between BIN integer value and binary floating point value.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

8.13 DFLT/DFLTP – 32-bit binary floating point number conversion

Instruction parameters

Name	Input parameters	Output parameters	instruction
DFLT	70000	70000.000	Converting data to floating point numbers is executed consecutively
DFLTP	70000	70000.000	Pulse execution converts data to floating point numbers

Graphical example:

Instruction Description:

Description
BIN integer→binary floating point conversion: conversion instruction between BIN integer value and binary floating point value.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–
D	D4000	BIN32	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

Chapter 9 High Speed Processing

9.1 REF/REFP -input and output refresh

Instruction parameters

Name	Input parameters	Output parameters	instruction
REF	-	-	Continuous execution refreshes data immediately
REF P	-	-	Pulse execution refreshes data immediately

Graphical example:

Instruction Description:

Description
Input and output refresh: The programmable controller adopts the input and output batch refresh mode, and the input terminal information is put into the input information memory before the 0-step operation.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	X200	Bit	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	K8	BIN16	-	-	-	-	-	-	-	-	-	-	-	*	*	-	-	-	-	-

9.2 REFF -Input refresh (with filter)

Instruction parameters

Name	Input parameters	Output parameters	instruction
REF	-	-	Continuous execution refreshes data immediately
REFF	-	-	Pulse execution refreshes data immediately

Graphical example:

Instruction Description:

Description
Filter adjustment: When the input condition is ON, it will be used as the

input filter nms, and the image memory of the input X000~X017 will be refreshed.

Available soft component:

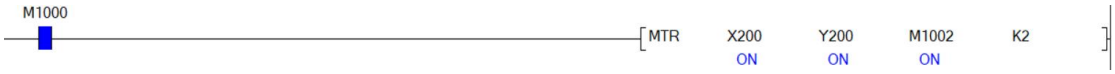
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
n	D3000	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

9.3 MTR matrix input

Instruction parameters

Name	Input parameters	Output parameters	instruction
MTR	2	2	Continuously execute input signal reading of 8*n points

Graphical example:



Instruction Description:

Description
Matrix input: A command to sequentially read input signals of 8 points and n columns using 8 points of input and n points of output.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	X200	Bit	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
D	Y200	Bit	–	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
D	M1002	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–
n	K2	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–

Chapter 10 Convenience Instructions

10.1 SER/SERP – 16-bit data retrieval

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
SER	200	200	1	Continuous execution to retrieve the same data, maximum value, minimum value
SERP	200	200	1	Pulse execution to retrieve the same data, maximum value, minimum value

Graphical example:

Instruction Description:

Description
Data search: Commands to search for the same data, search for the maximum value, and the minimum value.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	–	–	–	–	–	–	–	–	*	*
S	D3002	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D3004	BIN16	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
n	K1	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

10.2 DSER/DSERP – 32-bit data retrieval

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DSER	65590	65590	1	Continuous execution to retrieve the same data, maximum value, minimum value
DSERP	65590	65590	1	Pulse execution to retrieve the same data, maximum value, minimum value

Graphical example:

Instruction Description:

Description
Data search: Commands to search for the same data, search for the maximum value, and the minimum value.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	D3002	BIN32	*	*	*	*	*	*	*	–	*	–	–	–	*	*	–	–	*	*
D	D3004	BIN32	–	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
n	K1	BIN32	–	–	–	–	*	–	–	–	–	–	–	–	*	*	–	–	–	–

10.3 ABSD – 16-bit cam control absolute mode instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
ABSD	100	ON	The value of the retrieval counter is continuously compared with the standard value, and the output level

Graphical example:

Instruction Description:

Description
Barrel Sequence Absolute Mode: Instructions that create multiple output modes based on the current value of the counter.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	C0	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	*
D	M1001	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–
n	K1	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–

10.4 INCD -16-bit Cam Control Relative Instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
INCD	250	OFF	The value of the retrieval counter is continuously compared with the standard value, and the output level

Graphical example:

Instruction Description:

Description
Bucket Sequence Relative Mode: An instruction that uses a pair of counters to create multiple output modes.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	C0	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	*
D	M1002	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–
n	K1	BIN16	–	–	–	–	–	–	–	–	–	–	–	–	*	*	–	–	–	–

8.5 TTMR -teach timer

Instruction parameters

Name	Input parameters	Output parameters	instruction
TTMR	0	13	The time that the search button is pressed continuously

Graphical example:

Instruction Description:

n	Magnification	(D•)
K0	τ 0	(D•) ×1
K1	10 τ 0	(D•) ×10
K2	100 τ 0	(D•) ×100

Description
Teaching time: Measure the pressing time of the button (input condition) by the target + 1, multiply it by the magnification specified by n, and store it in

the target. This makes it possible to adjust the set time of the timer by pressing the button.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	–	–	–	–	–	–
n	K0	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

10.5 ALT/ALTP -Alternate output instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
ALT	-	-	Invert the software components when the continuous execution input is ON
ALTP	-	-	Invert the software components when the pulse execution input is ON

Graphical example:

Instruction Description:

Description
Alternate output: Each time the drive input changes from OFF→ON, the target is inverted.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
D	M1000	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–

10.6 RAMP -Ramp signal instruction

Instruction parameters

Name	Input parameters	Input parameters	Input parameters	Output parameters	instruction
RAMP	1000	20000	50	4483	Continuous execution is passed between the initial value and the target value according to the specified n times to get the changed data into the target address

Graphical example:

Instruction Description:

Description
Ramp signal: Write the specified initial value and target value into source code 1 and source code 2 in advance, and set the input condition to ON, the target content will gradually change from the value of source code 1 to the value of source code 2. This transition time is n scans.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	–	–	–	–	–	–
S	D4000	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	–	–	–	–	–	–
D	D5000	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	–	–	–	–	–	–
n	K50	BIN16	–	–	–	–	*	–	*	–	–	–	–	–	*	*	–	–	–	–

Chapter 11 External Device I/O

11.1 SEGD/SEGDP -7-segment decoder

Instruction parameters

Name	Input parameters	Output parameters	instruction
SEGD	2	91	Continuously execute the command to convert the data to light up the 7-digit digital tube
SEGDP	2	91	Pulse execution converts data into instructions to light up a 7-digit

			digital tube

Graphical example:

Instruction Description:

Description
SEG decoder: The data from 0 to F (hexadecimal) specified by the lower 4 bits of the source code is decoded into 7-segment display data and stored in the target. The upper 8 bits of the target are unchanged.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D4000	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

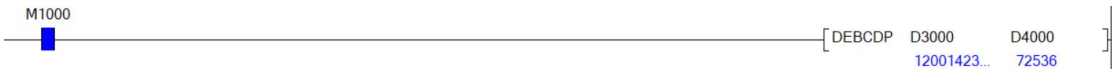
Chapter 12 Floating Point Operations

12.1 DEBCD/DEBCDP Binary floating-point number -> decimal floating-point number conversion instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DEBCD	70000.00	72536	Continuously execute the instructions to convert binary floating-point numbers to decimal floating-point numbers
DEBCDP	70000.00	72536	Pulse execution converts binary floating-point numbers to decimal floating-point numbers

Graphical example:



Instruction Description:

Description
Binary floating point → decimal floating point conversion: Convert the binary floating point value in the element specified by the source code to a decimal floating point value, and store it in the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.2 DEBIN/DEBINP Binary floating-point number -> decimal floating-point number conversion instruction

Instruction parameters

Name	Input parameters	Output parameters	instruction
DEBIN	7000.00	1200142336	Continuously execute the instructions to convert decimal floating-point numbers to binary floating-point numbers
DEBINP	7000.00	1200142336	Pulse execution converts decimal floating-point numbers to binary floating-point numbers

Graphical example:

Instruction Description:

Description
Decimal floating point → Binary floating point: Convert the decimal floating point value in the element specified in the source code to a binary floating point value and store it in the destination.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.3 DEADD/DEADDP Binary floating point number addition operation instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DEADD	65535.000	7000.001	72535.000	Executes the instruction to add two binary floating-point numbers consecutively
DEADDP	65535.000	7000.001	72535.000	Pulse execution adds two binary floating point numbers

Graphical example:

Instruction Description:

Description
Binary floating-point addition: Performs the addition of two binary floating-point values in the source code and stores them as binary floating-point values to the destination.

Available soft component:

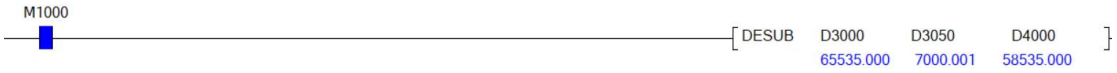
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
S	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.4 DESUB/DESUBP Binary floating point number subtraction instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DESUB	65535.000	7000.001	58535.000	Execute the instruction that subtracts two binary floating-point numbers in succession
DESUBP	65535.000	7000.001	58535.000	Pulse executes the instruction to subtract two binary floating point numbers

Graphical example:



Instruction Description:

Description
Binary floating-point subtraction operation: Calculate the binary floating-point value in the element specified by source code 2 from the binary floating-point value in the element specified by source code 1, and use the result as a binary floating-point value stored in the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
S	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.5 DEMUL/DEMULP Binary floating point multiplication instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DEMUL	18914.000	5.000	94570.000	Continuously execute the instruction to multiply two binary floating point numbers
DEMULP	18914.000	5.000	94570.000	Pulse execution to multiply two binary floating point numbers

Graphical example:

Instruction Description:

Description
Binary floating-point multiplication: Performs the addition of two binary floating-point values in the source code and stores them as binary floating-point values to the destination.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
S	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.6 DEDIV/DEDIVP Binary floating point number division operation instructions

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DEDIV	65599.000	5.000	13119.800	Continuously execute the instruction to divide two binary floating point numbers
DEDIVP	65599.000	5.000	13119.800	Pulse executes the instruction to divide two binary floating point numbers

Graphical example:

Instruction Description:

Description
Binary floating-point division operation: Divide the binary floating-point value in the element specified by source code 1 with the binary floating-point value in the element specified by source code 2, and use the result as a binary floating-point value stored into the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
S	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
D	D4000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.7 DESQR/DESQRP Binary floating point number extraction of square root operation instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
DESQR	65590.000	256.105	Continuously execute the extraction of square root operation instruction of floating-point numbers
DESQRP	65590.000	256.105	Pulse execution to find the extraction of square root of floating-point numbers

Graphical example:

Instruction Description:

Description
Binary floating-point extraction of square root: Perform the extraction of square root operation of the binary floating-point value in the element specified by the source code, and store it in the destination as a binary floating-point value.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	*	*	*	–	–	–
D	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.8 DENEG Binary floating point number sign flip instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
DENEG	65590.000	-65590.000	Continuously execute the instruction to reverse the sign of floating-point number

Graphical example:

Instruction Description:

Description
Floating point sign inversion: Invert the sign of the floating-point real number data of the software components specified by the target, and store it in the software components specified by the target.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
D	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.9 INT/INTP – 16-bit binary floating point number -> BIN integer conversion instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
INT	2000.000	2000	Continuously execute the instructions to convert a floating-point number to an integer
INTP	2000.000	2000	Pulse execution converts floating point numbers to integer instructions

Graphical example:

Instruction Description:

Description
Binary floating point → BIN integer conversion: Convert the binary floating point value in the element specified in the source code to a BIN integer, and store it in the target. In this case, the decimal point is rounded off.

Available soft component:

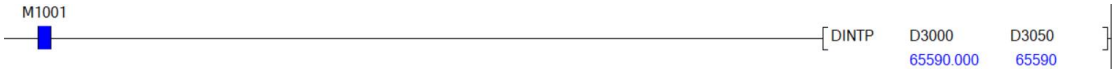
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–
D	D3050	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.10 DINT/DINTP – 32-bit binary floating point number -> BIN integer conversion instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
DINT	65590.000	65590	Continuously execute the instructions to convert a floating-point number to an integer
DINTP	65590.000	65590	Pulse execution converts floating point numbers to integer instructions

Graphical example:



Instruction Description:

Description
Binary floating point → BIN integer conversion: Convert the binary floating point value in the element specified in the source code to a BIN integer and store it in the destination. In this case, the decimal point is rounded off.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–
D	D3050	BIN32	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.11 DSIN/DSINP Binary floating point number SIN operation instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
DSIN	45.000	0.851	Continuously execute the SIN instruction to find the angle
DSINP	45.000	0.851	Pulse executes the SIN instruction to find the angle

Graphical example:

Instruction Description:

Description
Floating-point SIN operation: Find the SIN value of the angle (RAD) specified in the source code, and send it to the target instruction.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	*	–	–	–
D	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.12 DCOS/DCOSP Binary floating point number COS operation instructions

Instruction parameters

Name	Input parameters	Input parameters	instruction
DCOS	45.000	0.525	Continuously execute the COS instruction to find the angle
DCOSP	45.000	0.525	Pulse executes the COS instruction for finding the angle

Graphical example:

Instruction Description:

Description
Floating-point COS operation: finds the COS value of the angle (RAD)

specified in the source code, and transmits it to the target instruction.

Available soft component:

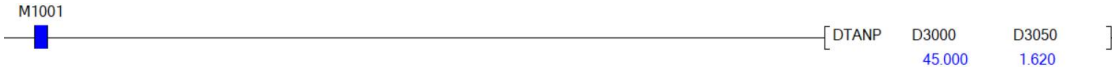
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	*	–	–	–
D	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

12.13 DTAN/DTANP Binary floating point number TAN operation instruction

Instruction parameters

name	Input parameters	Input parameters	instruction
DTAN	45.000	1.620	Continuously execute the TAN instruction to find the angle
DTAN	45.000	1.620	The pulse executes the TAN instruction for finding the angle

Graphical example:



Instruction Description:

Description
Floating-point TAN operation: finds the TAN value of the angle (RAD) specified in the source code, and transmits it to the target instruction.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	*	–	–	–
D	D3050	real	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–

Chapter 13 Data Processing 2

13.1 SWAP/SWAPP – 16-bit high and low byte swap operation instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
SWAP	1234	-11772	Continuously execute high and low byte swap instructions
SWAPP	1234	-11772	The pulse executes the high and low byte swap instruction

Graphical example:

Instruction Description:

Description
High and low byte conversion: When a 16-bit instruction is used, the low 8 bits and the high 8 bits are exchanged. In 32-bit instructions, the lower 8 bits and the upper 8 bits are exchanged respectively.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	-	*	*	*	*	*	*	*	*	-	-	-	-	-	-	-	*	*

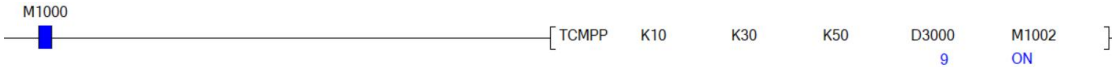
Chapter 14 Clock Operations

14.1 TCMP/TCMPP Clock Data Compare Instructions

Instruction parameters

Name	Input parameters	Output parameters	instruction
TCMP	9	ON	Continuous execution is compared to baseline time data
TCMPP	9	ON	Pulse execution is compared with base time data

Graphical example:



Instruction Description:

Description
Clock data comparison: The time of the source code [source code 1, source code 2, S3] is compared with the clock data of 3 points with the source code as the starting point, and the 3-point ON/OFF operation is performed from the target according to the size of the same.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K10	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	K30	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	K50	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	D3000	BIN16	-	-	-	-	*	*	*	-	-	-	-	-	-	-	-	-	*	*
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-

14.2 TZCP/TZCPP Clock Data Compare Instructions

Instruction parameters

Name	Input parameters	Output parameters	instruction
TZCP	7	ON	Comparing continuous execution with benchmark time interval data
TZCPP	7	ON	Pulse execution is compared with reference time interval data

Graphical example:

Instruction Description:

Description
Clock data area comparison: Compare the comparison range of the upper and lower 2 o'clock with the 3 o'clock time data starting from the source code, and set 3 o'clock from the destination to ON/OFF according to its size range.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	D3050	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	D4000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
D	M1001	Bit	–	*	*	*	–	–	–	–	–	–	–	–	–	–	–	–	–	–

14.3 TADD/TADDP Clock data addition operation instruction

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
TADD	9	12	21	Continuously execute the addition of two time data and store it in the target address
TADDP	9	12	21	Pulse execution adds two time data and stores it in the target address

Graphical example:

Instruction Description:

Description
Clock data addition operation: Add the time data stored in 3 points starting with the source code and the time data at 3 points starting with source code 2, and store the result in 3 points starting with the destination code.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	D3050	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
D	D4000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*

14.4 TSUB/TSUBP Clock Data Subtraction Instructions

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
TSUB	16	12	4	Continuously execute the subtraction of two time data and store them in the target address
TSUBP	16	12	4	The pulse execution subtracts the two time data and stores it in the target address

Graphical example:

Instruction Description:

Description
Subtraction of clock data: From the time data stored at 3 points starting from source code 1, subtract the time data starting from source code 2 at 3 points, and store the result to 3 points starting with destination code. Click.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
S	D3050	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*
D	D4000	BIN16	–	–	–	–	*	*	*	–	–	–	–	–	–	–	–	–	*	*

14.5 HOUR Chronometer

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
HOUR	3	3	ON	Continuous execution of cumulative monitoring of electric shock ON duration

Graphical example:

Instruction Description:

Description
Chronograph command: The command to add the time after the input contact is activated in units of one hour.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K3	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
D	D3000	BIN16	-	-	-	-	*	-	*	-	-	-	-	-	-	-	-	-	-	-
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-

14.6 DHOUR Chronometer

Instruction parameters

Name	Input parameters	Input parameters	Output parameters	instruction
DHOUR	3000	4000	ON	Continuous execution of cumulative monitoring of electric shock ON duration

Graphical example:

Instruction Description:

Description
Chronograph command: The command to add the time after the input contact is activated in units of one hour.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K3000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
D	D3000	BIN32	-	-	-	-	*	-	*	-	-	-	-	-	-	-	-	-	-	-
D	M1002	Bit	-	*	*	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Chapter 15 External Devices

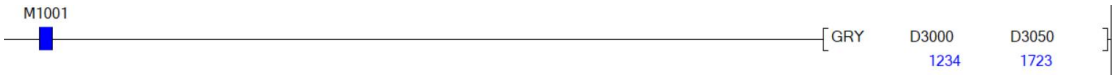
15.1 GRY/GRYP – 16-bit Gray code conversion

Instruction parameters

Name	Input parameters	Input parameters	instruction
GRY	1234	1723	Continuously execute the output instruction that converts the BIN value to gray code

GRYP	1234	1723	Pulse execution converts BIN value to gray code output instruction

Graphical example:



Instruction Description:

Description
Gray code conversion: Convert the BIN value to Gray code and transmit it. The data conversion speed depends on the scan time of the programmable controller.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D3050	Gray code	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

15.2 GBIN/GBINP – Inverse conversion of 16-bit Gray code

Instruction parameters

Name	Input parameters	Input parameters	instruction
GBIN	1723	1234	Continuously execute the instruction to convert Gray code value to BIN output
GBINP	1723	1234	Pulse execution converts Gray code value to BIN output command

Graphical example:

Instruction Description:

Description
Gray code inverse conversion: Convert gray code to BIN value and transmit the command. Can be used for absolute position detection with a Gray code encoder.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	Gray code	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
D	D3050	BIN16	–	*	*	*	*	*	*	*	*	–	–	–	–	–	–	–	*	*

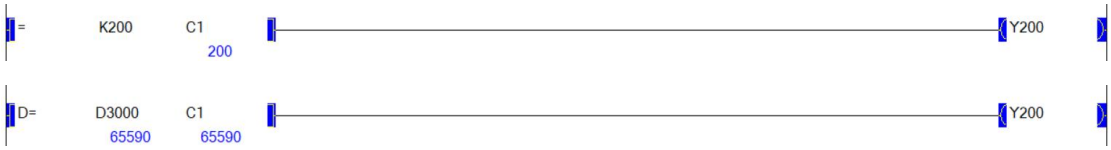
Chapter 16 Contact Instruction Comparison

16.1 LD=/LDD= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
LD=	200	-	Continuous execution turns ON the output that meets the condition
LDD=	65590	65590	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND<=: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. AND<= is a contact type comparison instruction that connects other contacts in series.

Available soft component:

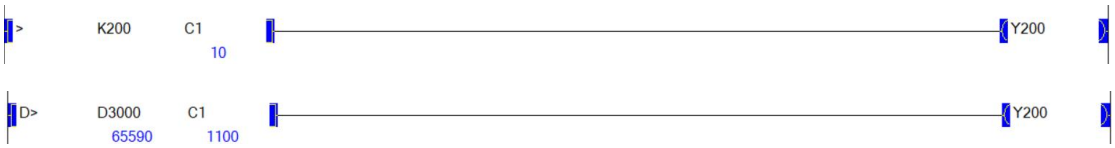
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.2 LD>/LDD> instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
LD>	-	10	Continuous execution turns ON the output that meets the condition
LDD>	65590	1100	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction LD>: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. LD> is a contact type comparison instruction connected to the bus.

Available soft component:

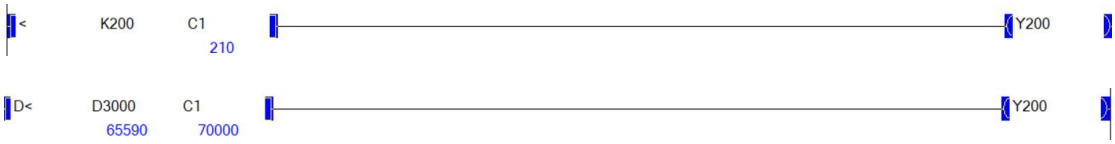
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*

16.3 LD</LDD< instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
LD<	-	210	Continuous execution turns ON the output that meets the condition
LDD<	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction LD<: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is executed according to the result. LD< is a contact type comparison instruction connected to the bus.

Available soft component:

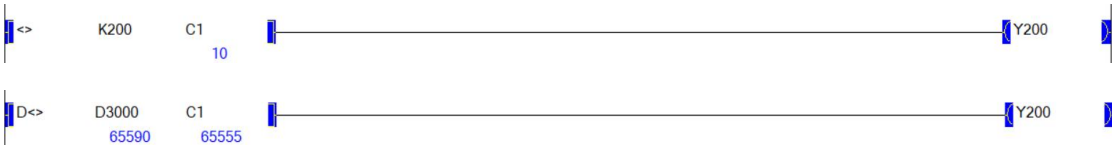
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.4 LD<>/LDD<> instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
LD<>	-	10	Continuous execution turns ON the output that meets the condition
LDD<>	65590	65555	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction LD<>: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. LD<> is a contact type comparison instruction connected to the bus.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

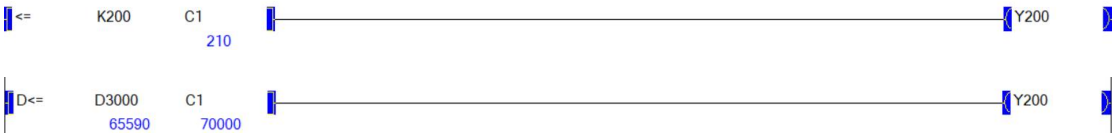
16.5 LD<=/LDD<= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
LD<=	-	210	Continuous execution turns ON the output that meets the condition

LDD<=	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction LD<=: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. LD<= is a contact type comparison instruction connected to the bus.

Available soft component:

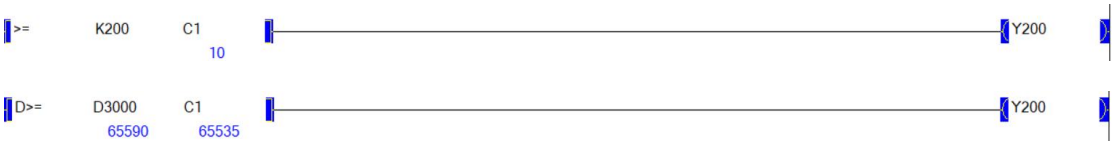
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.6 LD>=/LDD>= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
LD>=	-	10	Continuous execution turns ON the output that meets the condition
LDD>=	65590	655335	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction LD>=: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. LD>= is a contact type comparison instruction connected to the bus.

Available soft component:

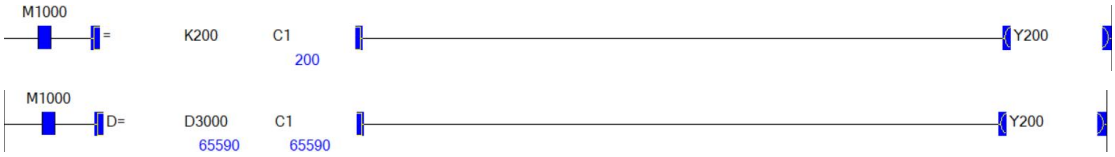
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.7 AND=/ANDD= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
AND =	-	200	Continuous execution turns ON the output that meets the condition
ANDD =	65590	655390	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND=: The contents of the source code are compared with BIN, and the sequence program operation in the latter stage is executed according to the result. AND= is a contact-type comparison instruction that connects other contacts in series.

Available soft component:

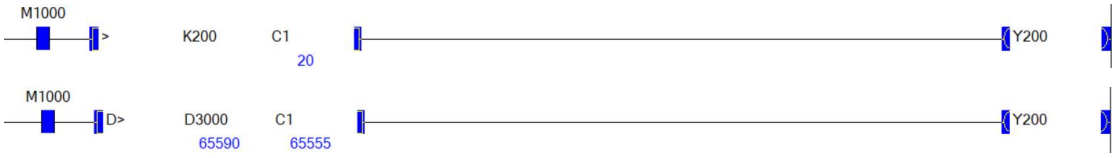
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.8 AND>/ANDD> instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
AND >	-	20	Continuous execution turns ON the output that meets the condition
ANDD >	65590	655355	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND>: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. AND> is a contact-type comparison instruction that connects other contacts in series.

Available soft component:

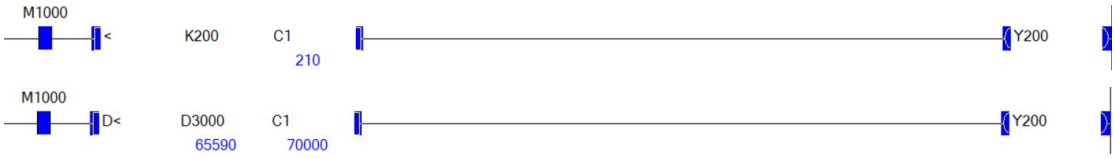
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.9 AND</ANDD< instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
AND <	-	210	Continuous execution turns ON the output that meets the condition
ANDD <	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND<: BIN comparison is performed on the contents of the source code, and the subsequent sequence program operation is executed according to the result. AND< is a contact-type comparison instruction that connects other contacts in series.

Available soft component:

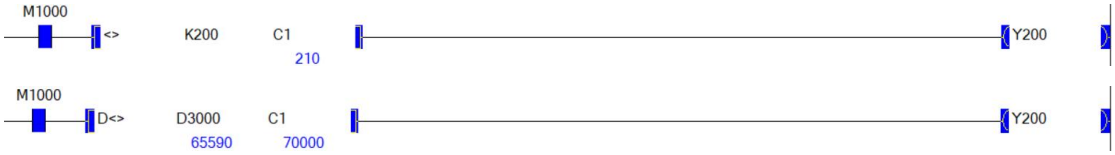
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.10 AND<>/ANDD<> instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
AND <>	-	210	Continuous execution turns ON the output that meets the condition
ANDD <>	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND<>: BIN comparison is performed on the contents of the source code, and the subsequent sequence program operation is executed according to the result. AND<> is a contact-type comparison command that connects other contacts in series.

Available soft component:

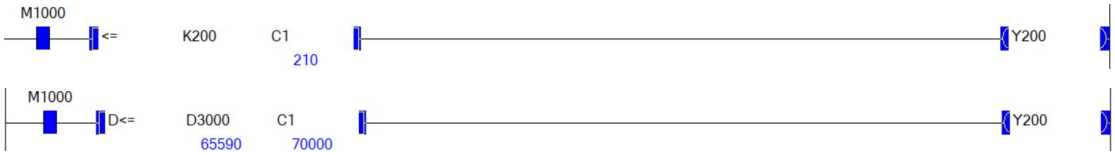
	Soft component input	data type	X	Y	M	S	D	U\	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.11 AND<=/ANDD<= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
AND <=	-	210	Continuous execution turns ON the output that meets the condition
ANDD <=	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND<=: BIN comparison is performed on the content of the source code, and the subsequent sequence program operation is performed according to the result. AND<= is a contact-type comparison command that connects other contacts in series.

Available soft component:

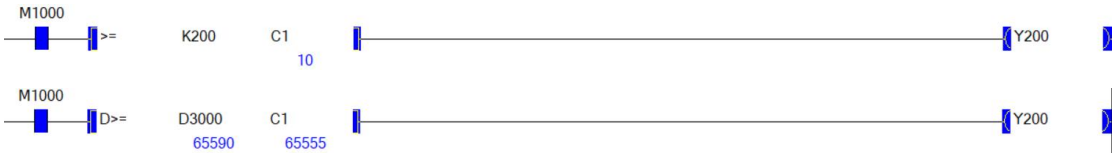
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.12 AND>=/ANDD>= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
AND >=	-	10	Continuous execution turns ON the output that meets the condition
ANDD >=	65590	65555	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND>=: BIN comparison is performed on the content of the source code, and the sequence control program operation in the latter stage is executed according to the result.

AND>= is a contact-type comparison command that connects other contacts in series.

Available soft component:

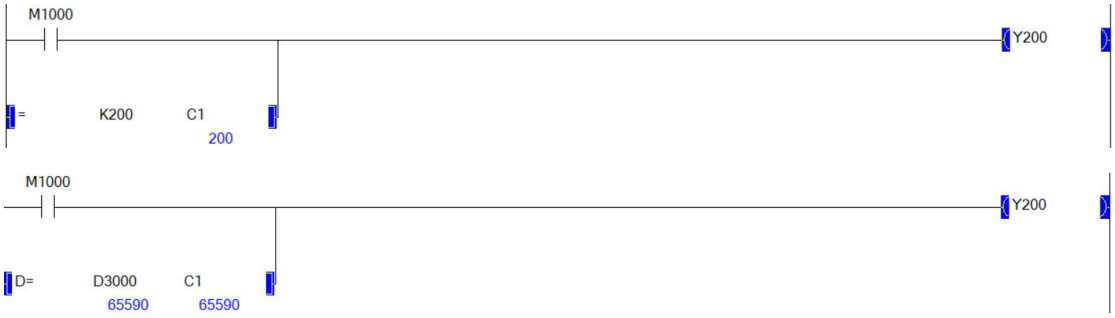
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.13 OR=/ORD= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
OR =	-	200	Continuous execution turns ON the output that meets the condition
ORD =	65590	65590	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction OR=: Compare the contents of the source code, and execute the sequence program operation in the latter stage according to the result. OR= is a contact-type comparison instruction for connecting other contacts in parallel.

Available soft component:

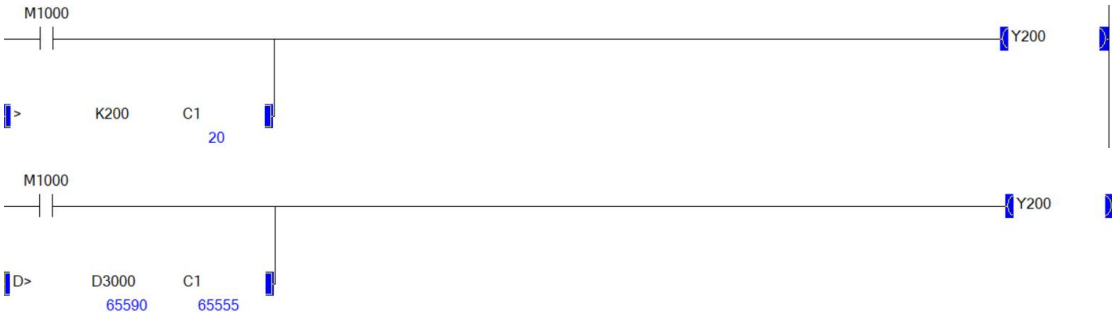
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	D3000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	C1	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*

16.14 OR>/ORD> instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
OR >	-	20	Continuous execution turns ON the output that meets the condition
ORD >	65590	65555	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction OR>: Compare the content of the source code, and execute the sequence program operation in the latter stage according to the result. OR> is a contact-type comparison command that connects other contacts in parallel.

Available soft component:

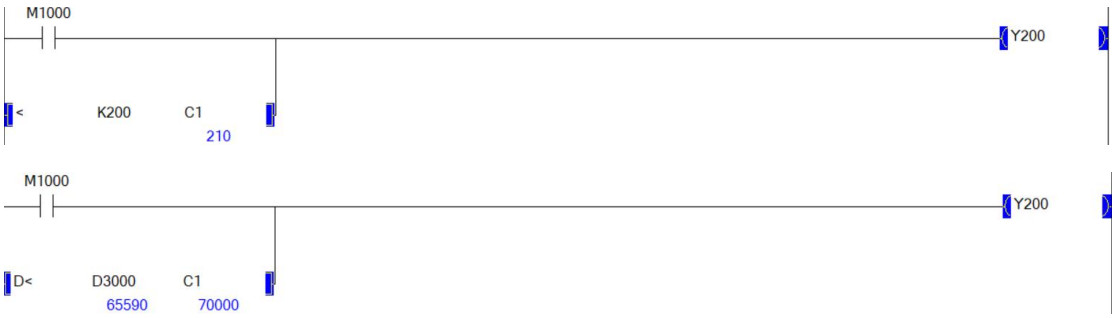
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	K200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	C1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.15 OR</ORD< instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
OR<	-	210	Continuous execution turns ON the output that meets the condition
ORD<	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND<: BIN comparison is performed on the contents of the source code, and the subsequent sequence program operation is executed according to the result. AND< is a contact-type comparison command that connects other contacts in series.

Available soft component:

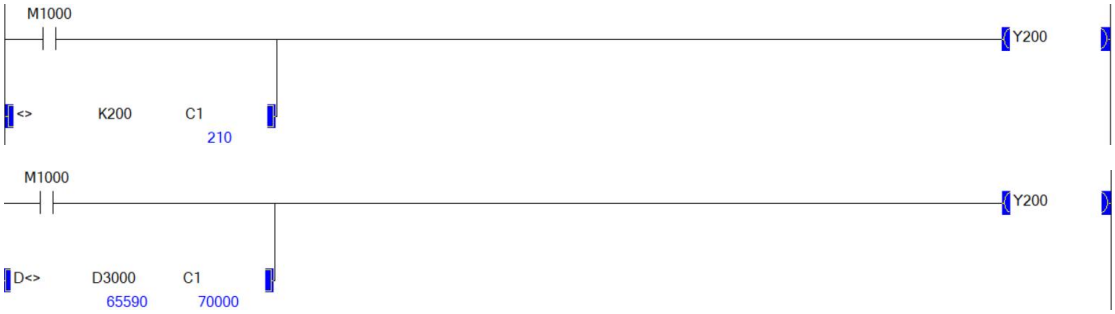
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	–	–	–	*	*	–	–	*	*

16.16 OR<>/ORD<> instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
OR<>	-	210	Continuous execution turns ON the output that meets the condition
ORD<>	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

Description
Contact type comparison instruction AND<>: BIN comparison is performed on the content of the source code, and the subsequent sequence

program operation is performed according to the result. AND<> is a contact-type comparison command that connects other contacts in series.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	d3000	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.17 OR<=/ORD<= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
OR<=	-	210	Continuous execution turns ON the output that meets the condition
ORD<=	65590	70000	Continuous execution turns ON the output that meets the condition

Graphical example:

Instruction Description:

Description
Contact type comparison instruction OR<=: Compare the contents of the source code, and execute the sequence program operation in the latter stage according to the result. OR<= is a contact-type comparison command for connecting other contacts in parallel.

Available soft component:

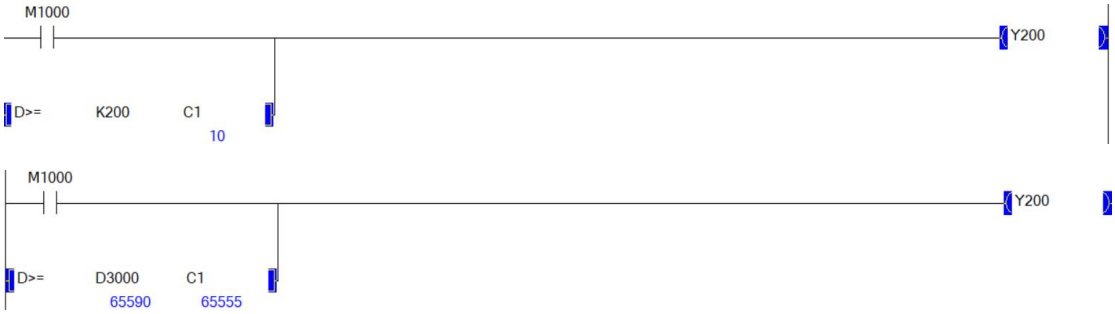
	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	k200	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*
S	c1	BIN16	*	*	*	*	*	*	*	*	*	-	-	-	*	*	-	-	*	*

16.18 OR>=/ORD>= instruction

Instruction parameters

Name	Input parameters	Input parameters	instruction
OR>=	-	10	Continuous execution turns ON the output that meets the condition
ORD>=	65590	65555	Continuous execution turns ON the output that meets the condition

Graphical example:



Instruction Description:

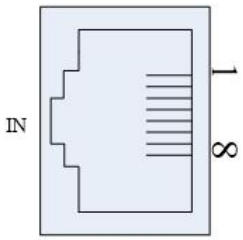
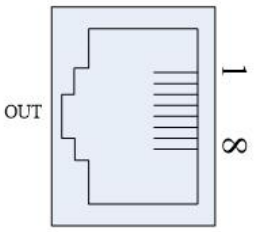
Description
Contact type comparison instruction OR>=: Compare the contents of the source code, and execute the sequence program operation in the latter stage according to the result. OR>= is a contact-type comparison command that connects other contacts in parallel.

Available soft component:

	Soft component input	data type	X	Y	M	S	D	U \	R	V	Z	P	I	N	K	H	E	$	T	C
S	d3000	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*
S	c1	BIN32	*	*	*	*	*	*	*	-	*	-	-	-	*	*	-	-	*	*

Chapter 17 CN1 Network Port Definition

Compared with VC210, VC600 servo adds a 232 serial port, and the definition of CN1 is as follows:

<table>
<tr><th>Position and Function</th><th>Terminal shape</th><th>Description</th></tr>
<tr><td>CN1 IN Port</td><td> 
</td><td>
<table>
<tr><th>No.</th><th>Define</th><th>Description</th></tr>
<tr><td>1</td><td>NC</td><td>NC</td></tr>
<tr><td>2</td><td>NC</td><td>NC</td></tr>
<tr><td>3</td><td>GND</td><td>power ground</td></tr>
<tr><td>4</td><td>SG+</td><td>RS485 signal is positive</td></tr>
<tr><td>5</td><td>SG-</td><td>RS485 signal is negative</td></tr>
<tr><td>6</td><td>TX</td><td>Send data</td></tr>
<tr><td>7</td><td>RX</td><td>Receive data</td></tr>
<tr><td>8</td><td>GND</td><td>Power ground</td></tr>
</table>
</td></tr>
<tr><td>CN1 OUT Port</td><td> 
</td><td>
<table>
<tr><th>No.</th><th>Define</th><th>Description</th></tr>
<tr><td>1</td><td>NC</td><td>NC</td></tr>
<tr><td>2</td><td>NC</td><td>NC</td></tr>
<tr><td>3</td><td>GND</td><td>power ground</td></tr>
<tr><td>4</td><td>SG+</td><td>RS485 signal is positive</td></tr>
<tr><td>5</td><td>SG-</td><td>RS485 signal is negative</td></tr>
<tr><td>6</td><td>NC</td><td>NC</td></tr>
<tr><td>7</td><td>NC</td><td>NC</td></tr>
<tr><td>8</td><td>GND</td><td>Power ground</td></tr>
</table>
</td></tr>
</table>

(1) It is necessary to connect the power ground of the controller (PLC) and the power ground of the servo drive

(2) When multiple drives are used in parallel with RS485 bus, please add a 120 Ω terminal resistance between the SG+ and SG- terminals of the farthest drive

Note: When wiring, please connect the GND terminal of the host device and the GND terminal of the servo drive together.

Release Notes

release date	Description	version
2022-04-28	The first edition of the instruction manual	1.01