
02 Description of each device

last modified by Stone Stone

on 2022/06/15 11:20

Table of Contents

User device	3
Input relay (X)	3
Output relay (Y)	3
Internal relay (M)	4
Status relay (S)	4
Timer (T)	4
Counter (C)	6
Long Counter (LC)	6
High-speed counter (HSC)	7
Data Register (D & R)	7
System device	8
Special Relay (SM)	8
Special Register (SD)	8
Index Register	8
Index register ([D])	8
Nesting	9
Nesting (N)	9
pointer	9
Pointer (P)	9
Constant	9
Decimal constant (K)	9
Hexadecimal constant (H)	10
Real number constant (E)	10
String constant	10
Power-down retention setting	10
Special use of device	11

Device list

Classification	Type	Device name	Sign	Range	Mark
User device	Bit	Input	X	0 to 1777	Octal number
	Bit	Output	Y	0 to 1777	Octal number
	Bit	Internal relay	M	0 to 7999	Decimal number
	Bit	Step relay	S	0 to 4095	Decimal number
	Bit/word	Timer	T	0 to 511	Decimal number
	Bit/word	Counter	C	0 to 255	Decimal number
	Bit/double word	Long counter	LC	0 to 255	Decimal number
	Bit/double word	High-speed counter	HSC	0 to 15	Decimal number
	Word	Data Register	D	0 to 7999	Decimal number
	Word	Data Register	R	0 to 29999	Decimal number
System software	Bit	Special	SM	0 to 4095	Decimal number
	Word	Special register	SD	0 to 4095	Decimal number
Index register	Word	Index register	[D]	0 to 7999	Decimal number
	Word	Index register	V	0 to 7	Decimal number
	Double word	Long index register	Z	0 to 7	Decimal number
Nested	Bit	Nested	N	0 to 7	Decimal number
Pointer	-	Pointer	P	0 to 4095	Decimal number
Constant	-	Decimal constant	K	-	Decimal number
	-	Hexadecimal constant	H	-	Hexadecimal number
	Single precision floating point	Real constant	E	-	-

User device

Input relay (X)

The input relay represents the original PLC external input signal status, and the external signal status is detected through the input X port. 0 represents the external signal is open, and 1 represents the external signal is closed.

Using the program instruction method, the state of the input relay cannot be modified, and its node signals (normally open, normally closed) can be used indefinitely in the user program.

The relay signal is identified by Signs such as X0, X1,.....X7, X10, X11..... and its serial number is numbered in octal.

When an expansion module is connected, the extended X point will also use the X point as the component of the input signal state, and the occupied X point is the starting position of the X point used by the PLC with 0 as the end of the X point, such as PLC Occupy 17 to 24 X points (X0 to X21, X0 to X27), at this time the X points of the expansion module will be stored starting from X30.

Output relay (Y)

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

Y relay numbers are identified by Signs such as Y0, Y1,...Y7, Y10, Y11, etc., and their serial numbers are numbered in octal format. Y relay components can be used indefinitely as promised by users .

In terms of hardware, according to the different output components, it can be divided into relay type, transistor type, solid state relay type, etc. If there are output expansion module ports, they are numbered in sequence starting from the main module.

Internal relay (M)

The auxiliary relay M element is used as an intermediate variable in the execution of the user program, just like the auxiliary relay in the actual electronic control system, used for the transmission of status information, and multiple M variables can also be combined into word variables. M variables and external ports There is no direct connection, but you can copy X to M through program statements, or copy M to Y to connect with the outside world. An M variable can be used unlimited times.

The auxiliary relay M is identified by Signs such as M0, M1....., M7999, and its serial number is numbered in decimal system.

Status relay (S)

The state relay S is used for the design and execution of the step program. The STL step instruction is used to control the transfer of the step state S, simplifying the programming design. If STL programming is not used, S can be used as an M variable. State S variables are identified by Signs such as S0, S1...S4095, and their serial numbers are numbered in decimal system.

Timer (T)

The timer T is equivalent to the time relay in the relay system and is used to complete the timing function. The timer is an addition expression. When the timer expires, the current value and the set value are the same value.

The measurement starts when the timer coil turns ON. When the current value of the timer is consistent with the set value, it will become the time limit, and the timer contact will turn ON. When the timer coil is turned OFF, the current value will become 0, and the timer contact will also become OFF.

The T value range of the timer is 0 to 32767.

When the timer coil (OUT T instruction) is executed, the timer coil is turned on/off, the current value is updated, and the contact is turned on/off.

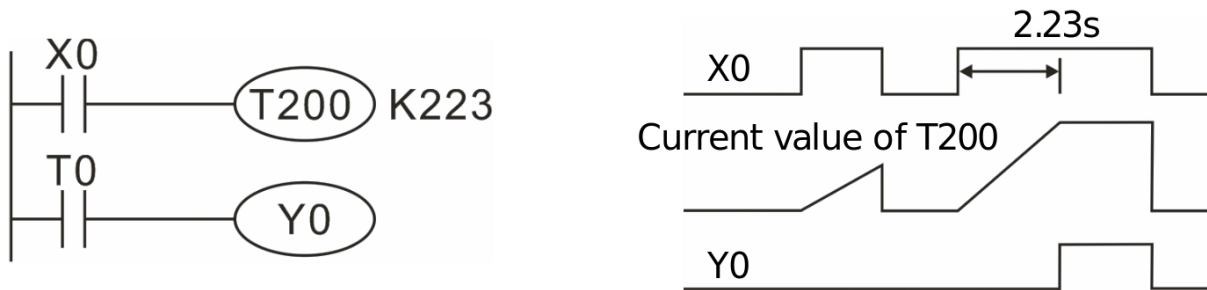
Diware number

T0ms timer
to
T191

T092ms subroutine timer (used in the subroutine, even if the subroutine is not called, it will still be updated)
to
T199

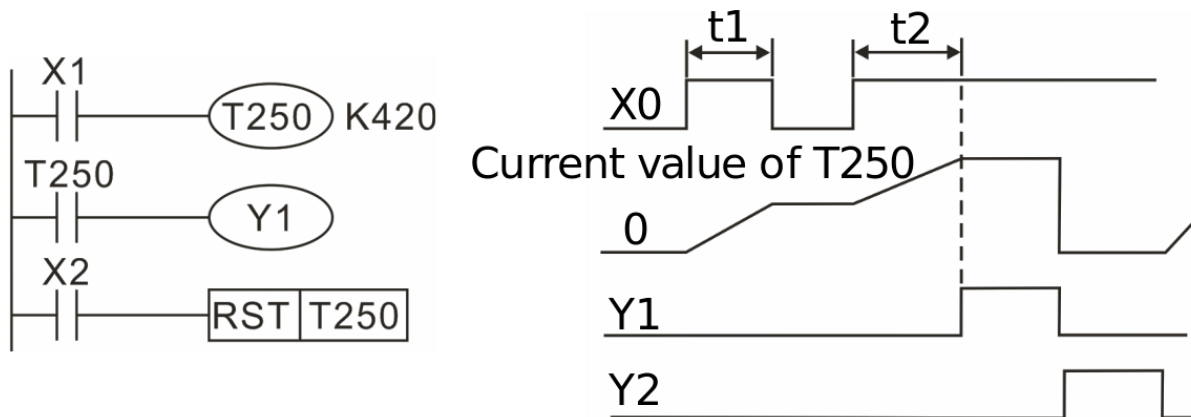
T200 timer
to
T245

(1) General-purpose timer (T0 to T245)



As shown in the figure above: when the normally open contact of X0 is turned on, the current value counter of T200 starts timing from zero and counts up the 10ms clock pulse. When the current value is equal to the set value 223, the timer's normally open contact is turned on and the normally closed contact is turned off, that is, the output contact of T200 will act after its coil is driven for 2.23s. After the normally open contact of X0 is disconnected, T200 is reset because the coil is de-energized. After reset, its normally open contact is disconnected, and the normally closed contact is connected, and the current value returns to zero.

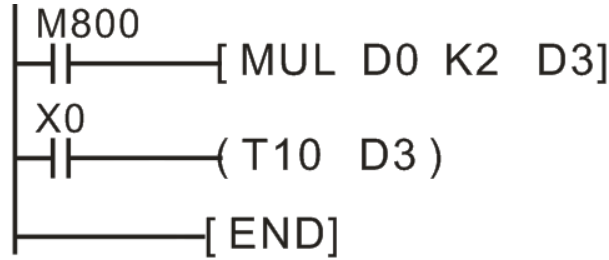
(2) Accumulative timer (T246 to T255)



When the X1 normally open contact in Figure b is turned on, the current value counter of T250 accumulates the 10ms clock pulse. When the normally open contact of X1 is disconnected or stopped, the counting stops, and the current value remains unchanged. When the normally open contact of X1 is turned on again, counting continues. When the accumulated time t_1+t_2 is 4.2s, the current value is equal to the set value of 420, the normally open contact of T250 is turned on and the normally closed contact is turned off. When the normally open contact of X2 is turned on, T250 will reset (because the coil of the accumulative timer will not reset when the power is off, you need to use the normally open contact of X2 and the reset instruction to force T250 to reset).

(3) Setting value

The timer time can use the constants (K, H) in the program memory as the set value, or can be specified indirectly by the content of the data register (D).



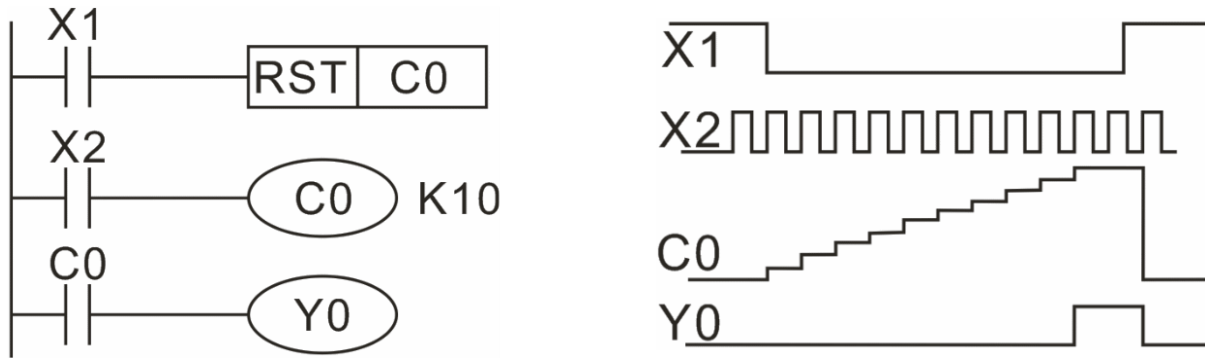
After PLC is powered on, multiplication is performed, $D3=D0*2$. Use the data of D3 as the timing time value of T10.

Counter (C)

The counter is used to complete the counting function. Each counter contains a coil, a contact, and a timer value register. Whenever the driving signal of the counter coil changes from OFF to ON, the counter reading value increases by 1, if the timer value reaches the preset time value, its contact action, a contact (NO contact) is closed, b contact (NC contact) is opened; if the timing value is cleared, the output a contact will be opened, and b contact (NC contact) will be closed. Some timers have features such as power-down retention, accumulation, etc., and maintain the value before power-down after power-on again.

The counters are identified by C0, C1, ..., C255, and the order is numbered in decimal.

The counter (C) is a 16-bit counter.



The setting value of the 16-bit up counter is 1 to 32767. As shown in the working process of the up counter in Figure c, after the normally open contact of X1 in the figure is turned on, C0 is reset, its corresponding bit storage unit is set to 0, the normally open contact of C0 is disconnected, and the normally closed contact Point is turned on, and its current counter value is set to 0 at the same time. X2 provides a counting input signal. When the reset input circuit of the counter is disconnected and the counting input circuit changes from disconnected to connected (that is, the rising edge of the counting pulse), the current value of counter C0 is increased by 1. After 10 count pulses, the current value of C0 is equal to the set value of 10, and its corresponding bit storage unit is set to 1, and the Y0 contact is turned on at this time. When counting pulses again, the current value does not change until the reset input signal is turned on, and the current value of the counter is set to 0.

Long Counter (LC)

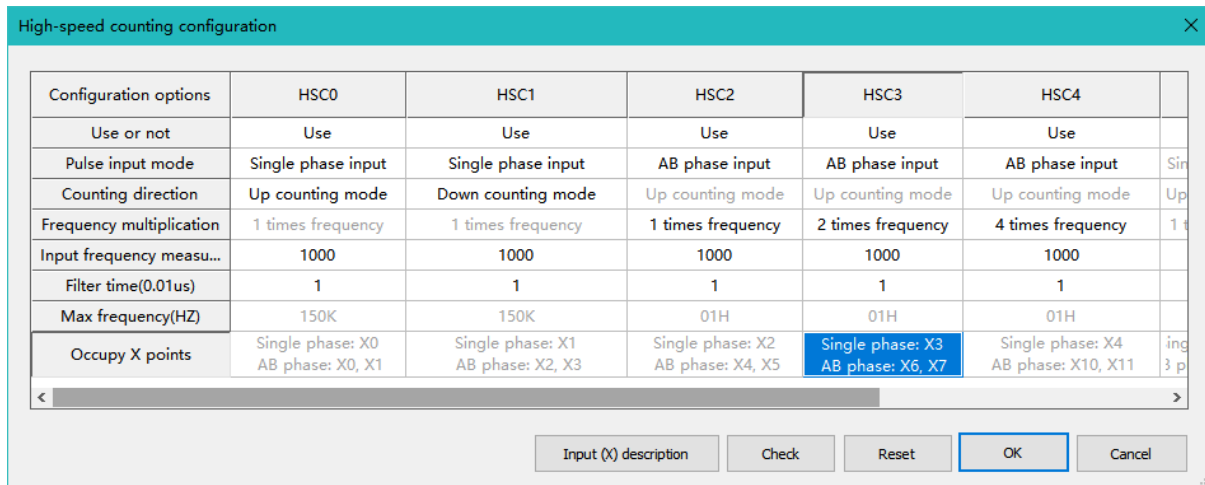
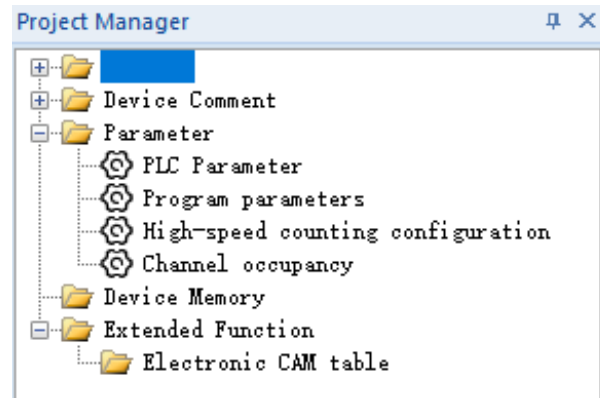
The long counter (LC) is basically the same as the counter (C), but compared to the counter (C), the long counter (LC) is a 32-bit register, and the range of values that can be counted is larger.

The long counter is identified by LC0, LC1, ..., LC255, and the sequence is numbered in decimal.

High-speed counter (HSC)

High-speed counter (HSC) is a device used for counting through external input of high-speed pulse signals. HSC is a 32-bit register.

The corresponding parameter configuration can be configured through: “project management” -> “parameters” -> “high-speed counter configuration”:



Data Register (D & R)

Registers are used for data calculation and storage, such as the calculation and calculation of timers, counters, and analog parameters. The width of each register is 16 bits. If 32bit instructions are used, the adjacent registers are automatically formed into 32bit registers for use, the lower address is the low byte, and the higher address is the high byte.

The address range of D register: D0 to D7999; the address range of R register: R0 to R29999.

The data involved in operations in most of our series PLC instructions are processed as signed numbers. For 16-bit registers, bit15 is the sign bit (0 represents a positive number, 1 represents a negative number); for a 32-bit register, the high byte bit15 It is the sign bit, and the value range is -32768 to 32767.

When 32-bit data needs to be processed, the two adjacent D registers can be formed into a 32-bit double word. For example, when accessing D100 in 32-bit format, use the high address D101 register as the high word and the high byte bit 15 as The sign bit of a double word can handle values from -2147483648 to 2 147483647.

System device

Special Relay (SM)

The special relay SM is an internal relay with a certain specification inside the programmable controller, so it cannot be used in the program like ordinary internal relays. It can be turned ON/OFF as needed to control the PLC.

For details, please refer to [Special relays \(SM\) list](#).

Special Register (SD)

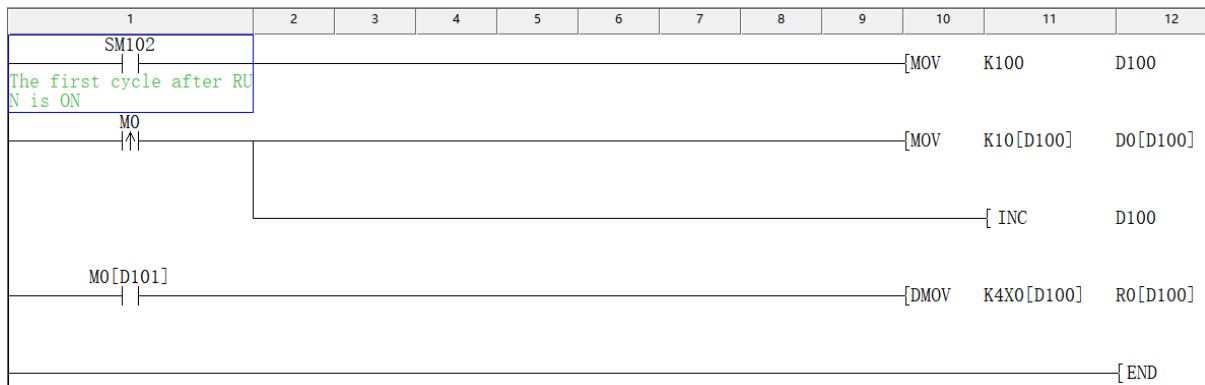
The special register SD is an internal register whose specifications are determined within the programmable controller, so it cannot be used in the program like a normal internal register, and the corresponding data can be written as needed to control the PLC.

For details, please refer to [Special register \(SD\) list](#).

Index Register

Index register ([D])

The index register is used to modify the index of the Devices. [D] The index register is actually the same as the data register D, ranging from D0 to D7999. The input method is as follows, just add [D] directly after the Devices:



The supported soft components for index modification are as follows:

Constant K, H plus index modification, such as D0 = 10, K10[D0] result = 10 + 10 = 20.

Constant E and character strings do not support index modification.

Add index modification to the data device, such as D0 = 10, the result of D10[D0] is the value of D20. Even if D10[D0] is used in a double word instruction, the double word value is the value of D20 (low word) and D21 (high word).

Bit device plus index modification, such as D0 = 10, the result of M0[D0] is the value of M10.

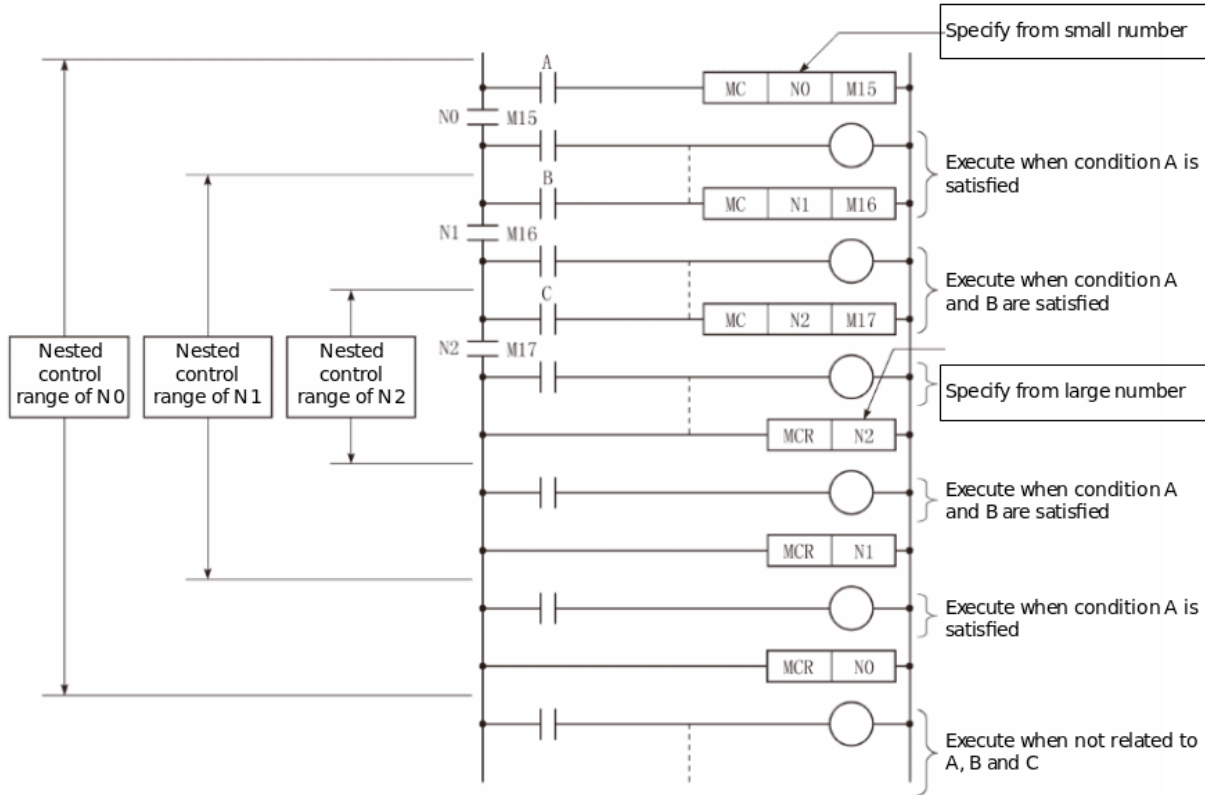
Bits are combined into words with index modification. For example, D0 = 10, K4M10[D0] first takes M10 offset by 10 addresses, and then combines them. The result is equivalent to K4M10.

Whether the index modification can be used depends on whether each instruction supports the format, you can check the "offset modification" in the description of the available device for each instruction.

Nesting

Nesting (N)

Nesting is a device used in master station control instructions (MC/MCR instructions)*1 to program operating conditions through a nested structure. Specify with a small number (order from N0 to N7) from the outside of the nested structure.



*1 is an instruction used to create an efficient ladder switching program by opening and closing the common bus of the Circuit program.

pointer

Pointer (P)

The pointer is the device used in the jump instruction (CJ instruction).

At present, the CALL instruction directly uses the subroutine name to call, and no longer uses the P pointer.

Constant

The constants are explained below.

Decimal constant (K)

“K” is a Sign that represents a decimal integer and is specified by K£ (for example: K123). It is mainly used to designate the set value of a timer or counter or the value in the operand of an application instruction. In 16bit

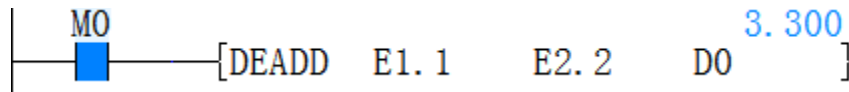
instructions, the value range of constant K is -32768 to 32767; in 32bit instructions, the value range of constant K is -247483648 to 2147483647.

Hexadecimal constant (H)

“H” is the Sign of hexadecimal number, specified by H□ (example: H123), mainly used to designate the value of the operand of the application instruction. The value range of the constant H is 0000 to FFFF; in the 32-bit instruction, the value range of the constant K is 0000, 0000 to FFFF, FFFF.

Real number constant (E)

“E” is the single-precision floating-point number representation Sign, specified by E□ (example: E1.23), mainly used to specify the value of the operand of the application instruction, the value range of the single-precision floating-point number E is $\pm 1.175495 \times 10^{-38}$ to $\pm 3.402823 \times 10^{+38}$ ($\pm 1.175495 \text{ E-}38$ to $\pm 3.402823 \text{ E+}38$) and 0 (7 effective digits).



(The address occupies D1 and D0)

String constant

The character string constant is the device that specifies the character string, and only supports the ASCII code character set, and any character string ends with a NULL character (00H). To use string devices, you must use double eye marks to modify the characters, as follows to convert the string to ASCII characters and fill in the device starting with D0:

Power-down retention setting

The user can freely configure the power-off storage range within the range of the Devices. The constant configuration is located in: “Project Management”→“Parameters”→“PLC Parameters”→“Device Latch”.

The screenshot shows the PLC Parameter dialog box with the Device Latch tab selected. The table below shows the configuration for various device latches.

	Symbol	Digit	Points	Device ra...	Enable...	Latch Start	Latch End	Latch sett...
Input	X	8	1024	0~1777				
Output	Y	8	1024	0~1777				
Supplemental	M	10	8000	0~7999	<input checked="" type="checkbox"/>	500	7999	0~7999
Status	S	10	4096	0~4095	<input checked="" type="checkbox"/>	500	4095	0~4095
Data	D	10	8000	0~7999	<input checked="" type="checkbox"/>	200	7999	0~7999
Data	R	10	30000	0~29999	<input checked="" type="checkbox"/>	0	29999	0~29999
Timer	T	10	512	0~511	<input checked="" type="checkbox"/>	200	511	0~511
Counter(16bit)	C	10	256	0~255	<input checked="" type="checkbox"/>	100	255	0~255
Counter(32bit)	LC	10	256	0~255	<input checked="" type="checkbox"/>	100	255	0~255
High-speed c...	HSC	10	16	0~15	<input checked="" type="checkbox"/>	0	15	0~15

Buttons: Check, Reset, OK, Cancel

#Note: The X and Y registers do not support the power-down save function.

Special use of device

(1) Use bits to form words

Format: KnB

K is a fixed character.

The value of n is 1 to 8, which means that $(n * 4)$ bits are combined into a word, such as K4M0 is a combination of M0 to M15.

B is the bit device number.

Example: Set a total of 32 bits M0 to M31 at the same time.

#Note: KnB type can also support index modification.

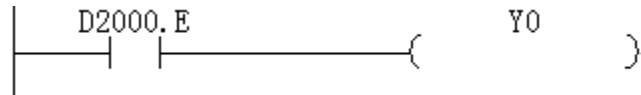
(2) Take the bit in the word

Format: D.b

D is the number of data device D (R is not available).

b is the bit number that needs to be taken, hexadecimal, and the value range is 0 to F.

Example: bit14 in D2000 is set and Y0 is output



#Note: D.b type can also support index modification.