01 Execution of the program

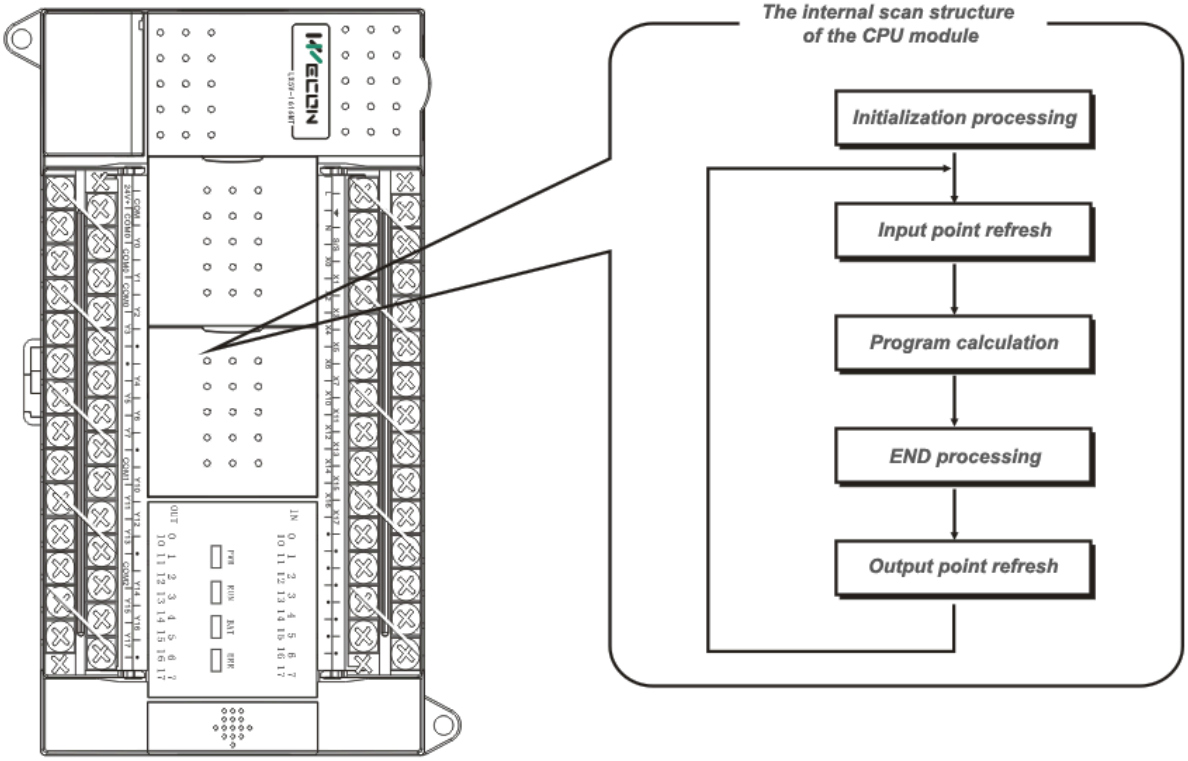
last modified by Joey

on 2022/06/14 17:12

Table of Contents

The composition of the scan

The scan configuration of the CPU module is as follows.



Initialization

The initialization based on the status of the CPU module is as follows.

√: execute. ×: not execute

Processing item	Status of the CPU module		
	When the power is ON	STOP	When STOP RUN
Initialization of input and output modules	√	×	×
CPU parameter check	√	×	×
Check of system parameters	√	×	×
Device initialization	√	×	√
Error clear	√	×	√

Input and output point refresh

Perform the following before starting program calculation.

Update the actual input point of the PLC to the input relay X.

The following is executed after the END instruction is executed.

Update the PLC output relay Y to the actual output point.

Key points

When performing a constant scan, the I/O refresh is performed after the waiting time of the constant scan.

Operation of the program

According to the program setting, the execution starts from step 0 of each program to the END instruction. This program is called the main program.

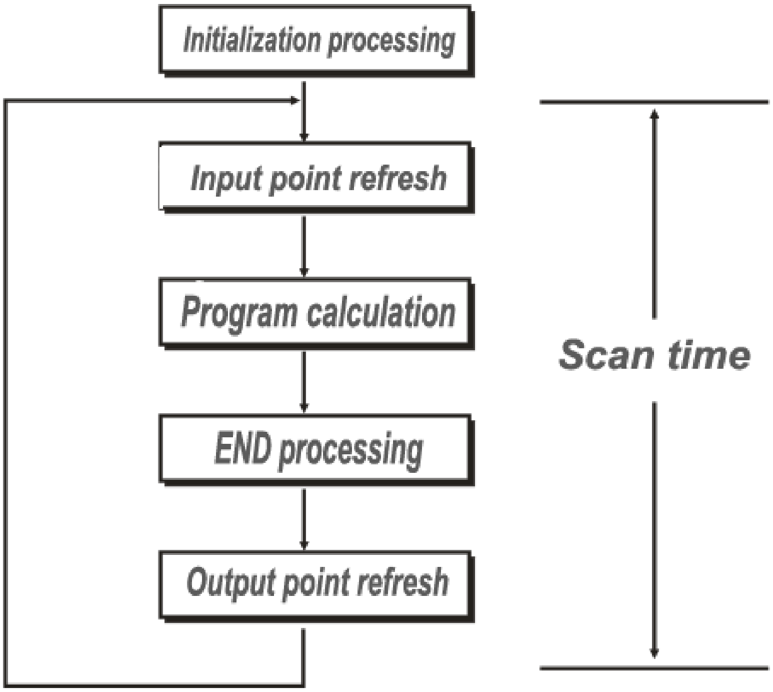
END processing

Perform the following processing.

- Completion processing of partial instructions
- Watchdog timer reset
- Communication processing
- Setting the value of special relay/special register (when the setting timing is END processing)

Scan time

The CPU module repeats the following processing, and the scan time is the total of the following processing and execution time.



The initial scan time indicates the time including this processing.

Initial scan time

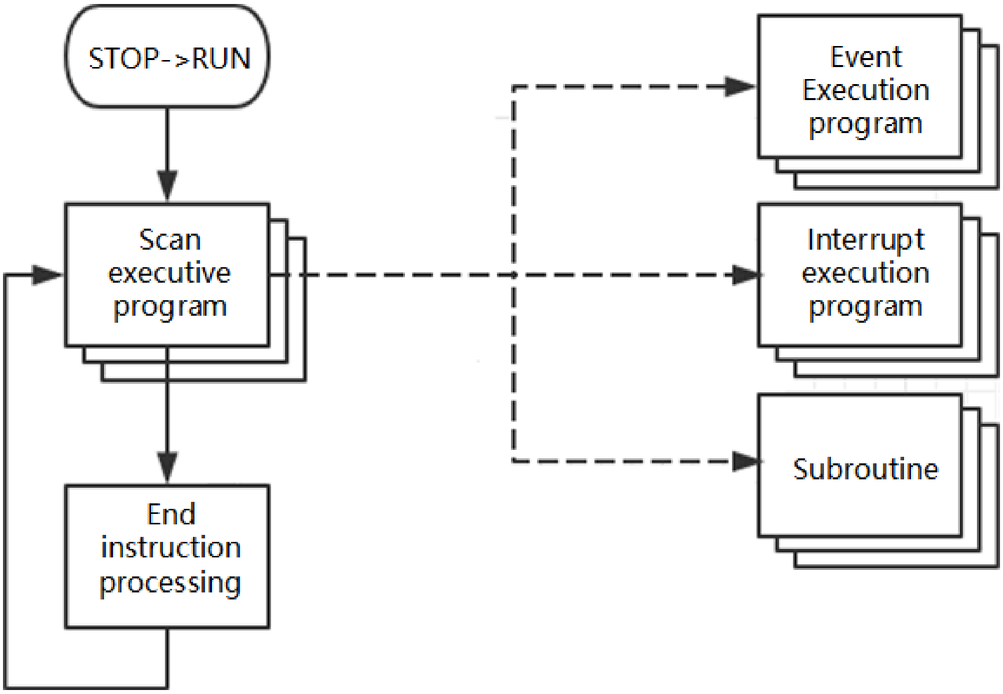
It is the first scan time of the CPU module in RUN.

Process as the following way:

The value stored in SD134 (initial scan time (ms unit)) and SD135 (initial scan time (s unit)).#

The flow of each procedure

When the CPU module changes to the RUN state, the programs are executed in sequence according to the program execution type and execution sequence settings.



Key points

When the execution types of the programs are the same, they are executed in the order set in the execution order.

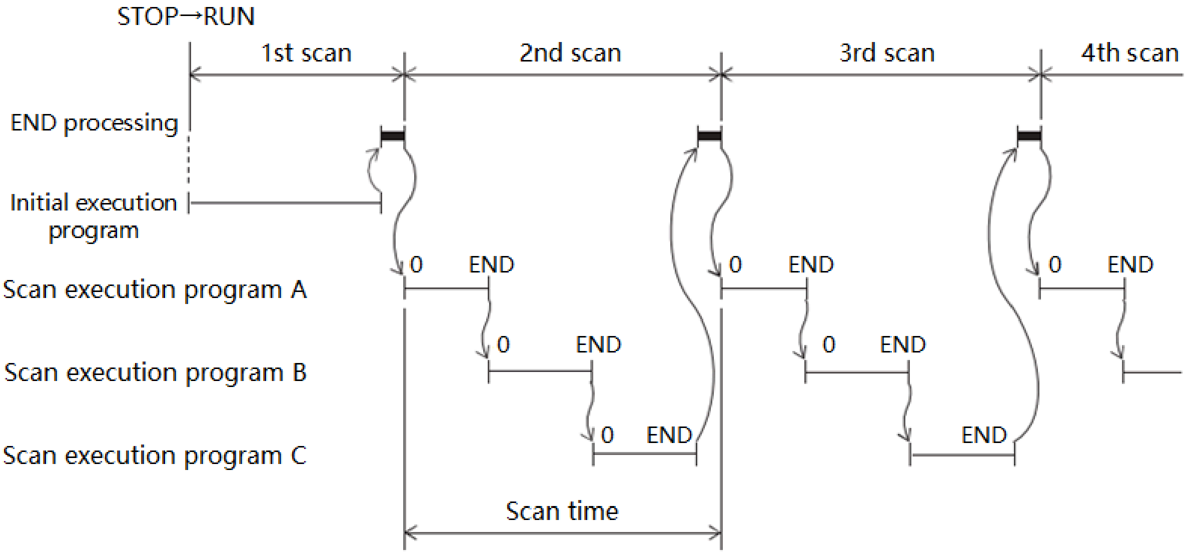
#Note:

When executing instructions that can be completed with multiple cycles (such as OUT T, RAMP, RS, etc.), they should be programmed in the scan program. If it is used in event execution type programs and mid-stage execution type programs, these instructions may not be executed in multiple scan cycles, causing actual results to be different from the ideal results. Therefore, unless events, interrupts and subroutines can be executed in each scan cycle. It is not recommended to use multi-cycle execution instructions in other situations.

Types of program execution

Scan execution program

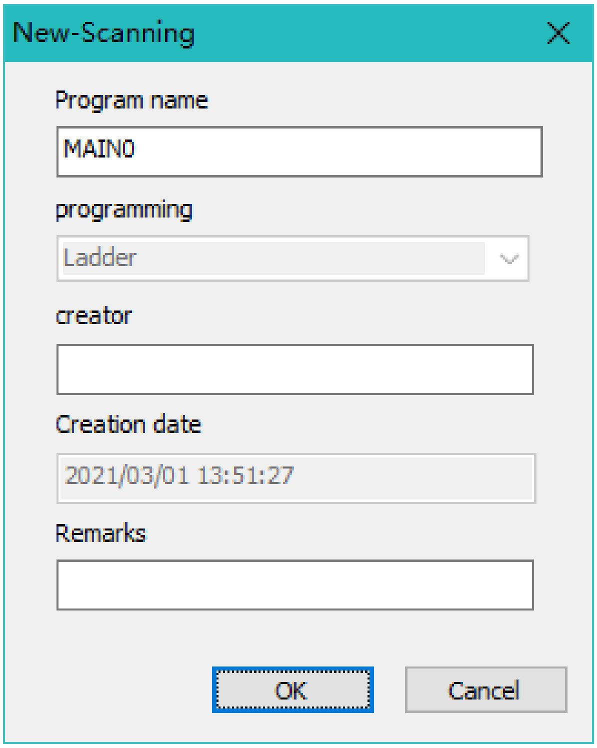
Each scan is executed only once from the next scan of the initial execution type program.



When multiple scan execution type programs are executed, the execution time of the scan execution type program is the time until all scan execution type programs are executed. In addition, before the execution of the scan execution type program is completed, if an interrupt program/event execution type program/subroutine is executed, the execution time will also be included.

Creation of multiple scanners

"Project Management"→ "Program"→ "Scan"→ Right click to create


New-Scanning

Program name: MAIN0

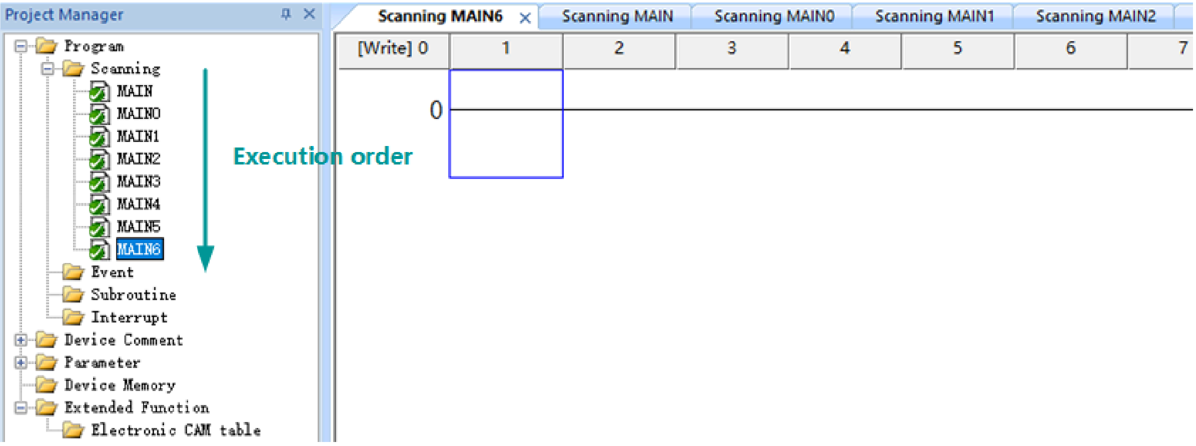
programming: Ladder

creator:

Creation date: 2021/03/01 13:51:27

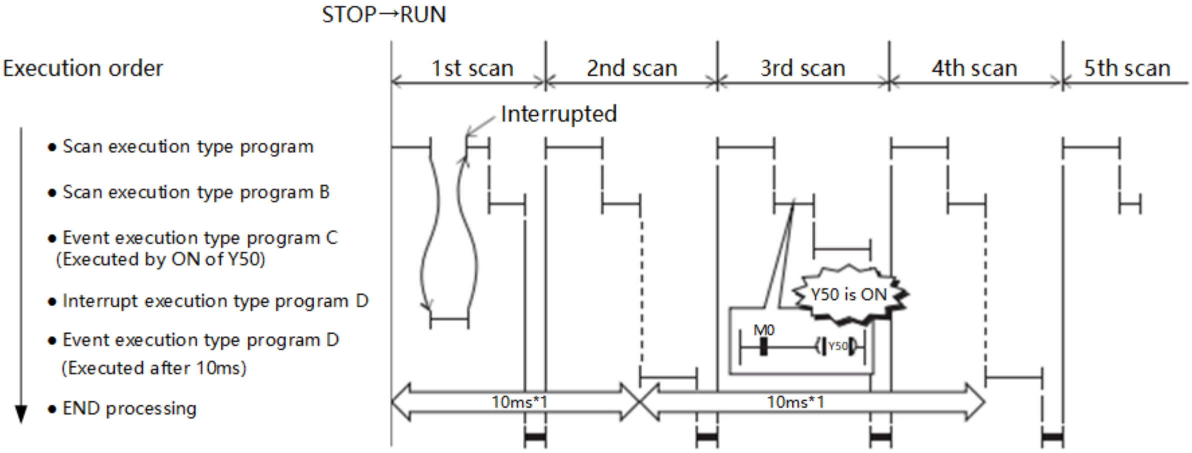
Remarks:

- Scan the program name: the program name requires to match case, and the program name cannot use the same name with device name (the device name does not match case).
- The input of /%$@&=to`^<>?:{}[],;!*.\\'" is not supported. It cannot exceed 64 characters. The default name is MAINx.
- The number of scan programs that can be built is limited to 100.
- Each scan program has been END ended, but only the last END instruction is completed to calculate a scan cycle.
- The execution sequence runs from top to bottom in the order of creation.



Event execution program

It uses the event specified by the user as a condition to trigger the program to start execution.


STOP→RUN
Execution order
1st scan
2nd scan
3rd scan
4th scan
5th scan
Interrupted
• Scan execution type program
• Scan execution type program B
• Event execution type program C
(Executed by ON of Y50)
• Interrupt execution type program D
• Event execution type program D
(Executed after 10ms)
• END processing
Y50 is ON
M0
Y50
10ms*1
10ms*1

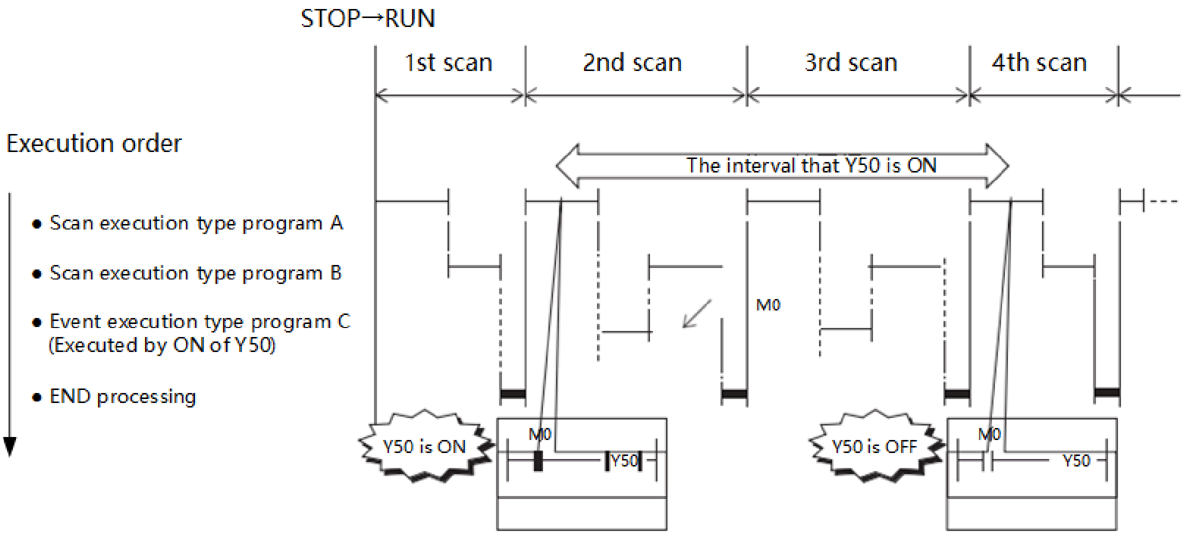
Trigger type

The trigger of event execution type program is as follows.

(1) ON event of bit data (TRUE)

- After the ON event is specified, if the contact that sets the trigger condition in the ON event is turned ON during the scan program, the ON event program will be executed in the scan program page*1 or before the END instruction is executed.
- The ON event program will only be executed once in a single scan cycle.
- After the ON event is executed, you can set whether to clear the current value of the output (Y) and timer (T) used in the program.

*1: Scan program paging: multiple scan programs are established, and each scan program is called a paging. After scan program A is executed, before scan program B is executed, it will be judged whether an event program needs to be executed.


STOP→RUN
1st scan
2nd scan
3rd scan
4th scan
Execution order
The interval that Y50 is ON
• Scan execution type program A
• Scan execution type program B
• Event execution type program C (Executed by ON of Y50)
• END processing
M0
Y50 is ON
M0
Y50
Y50 is OFF
M0
Y50

When it is the turn of the execution sequence of event execution type program C and Y50 is ON, the program is executed.

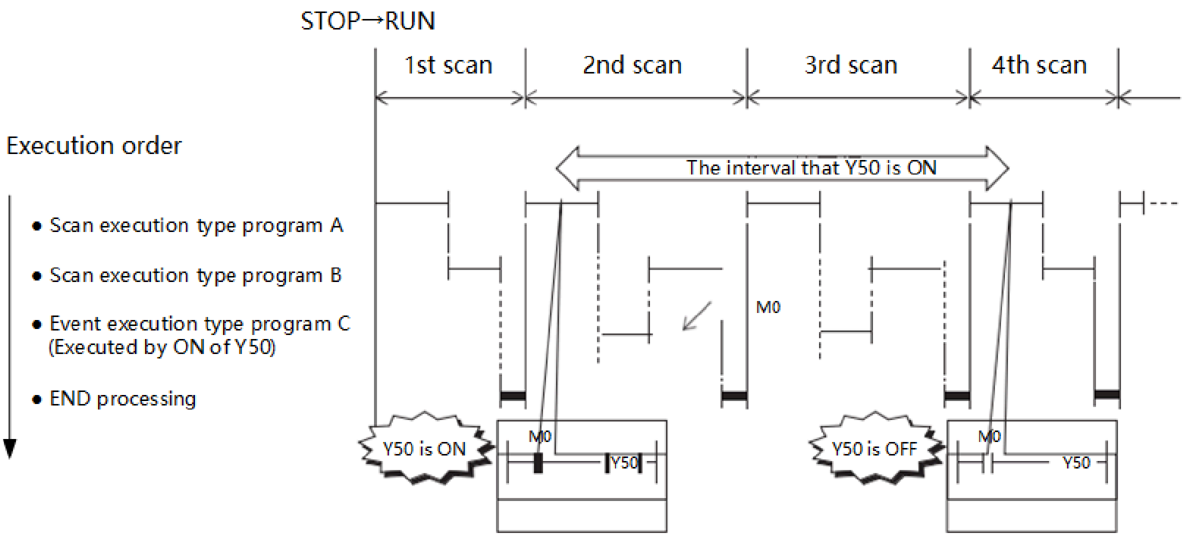
The devices that can be specified are as follows.

Project		Content
Device *1	Bit Device	X, Y, M, SM
	Bit specification of word device	D.b

*1 The indexed device cannot be specified.

(2) TIME event

After the program is to RUN state and the specified time has elapsed, event is executed one time when it comes to the execution sequence of the first corresponding program. For the second and subsequent executions, the time is re-measured from the start of the last event execution type program. After the specified time has elapsed, the program is executed repeatedly when it comes to the execution sequence of the first corresponding program. In addition, in the next scan after the corresponding program is executed, the current value of the output (Y) and timer (T) used in the corresponding program can be cleared. It can be used for programs that do not need to respond in a fixed period of time.


STOP→RUN
1st scan
2nd scan
3rd scan
4th scan
Execution order
The interval that Y50 is ON
• Scan execution type program A
• Scan execution type program B
• Event execution type program C (Executed by ON of Y50)
• END processing
M0
Y50 is ON
M0
Y50
Y50 is OFF
M0
Y50

After the specified time has elapsed, when it comes to the first execution sequence, the event execution type program C is executed.

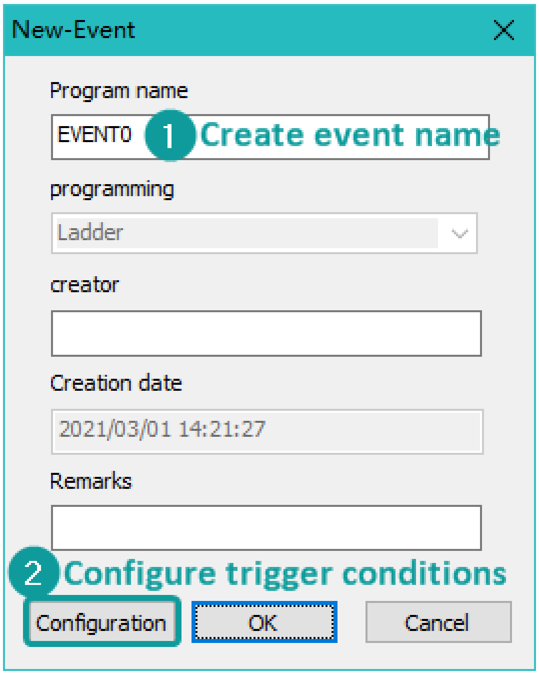
Key points

When set to clear the current value of output and timer, and the scan time is longer than the set value of elapsed time, the current value of output and timer will not be cleared.

Operational steps

1) New **event**

Project management→ Program→ Event → Right click to create.


New-Event

Program name: EVENT0

1 Create event name

programming: Ladder

creator:

Creation date: 2021/03/01 14:21:27

Remarks:

2 Configure trigger conditions

Event program name:

- The program name requires to match case, and the program name with the same name as the device cannot be used (the device name does not match case).

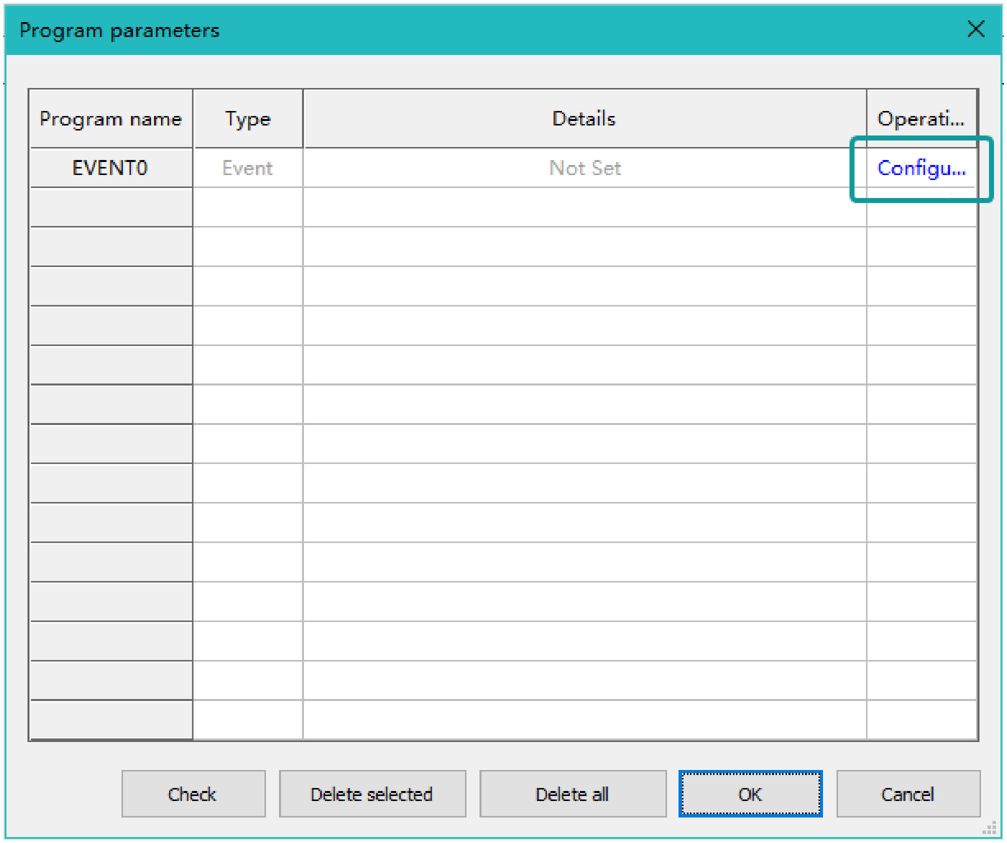
- The program name does not support /%$@&=to`^<>?:{}[],; !*.\\'" character input.
- The length of the program name cannot exceed 64 characters. The default name is EVENTx.

A maximum of 100 new event programs could be created.

2) Execution type

There are two ways to configure the event execution type:

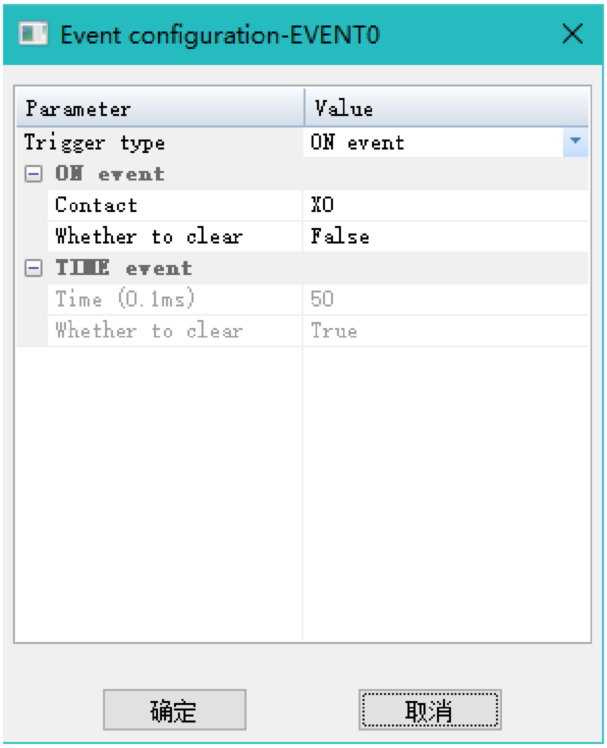
1. Configure when creating a new event program, as shown in the figure above.w
2. Project management→ Program→ Parameter → Program parameter→Configuration



Program name	Type	Details	Operati...
EVENT0	Event	Not Set	Configu...

Configuration instructions:

1) Configuration interface



Parameter	Value
Trigger type	ON event
⊟ **ON event**	
Contact	X0
Whether to clear	False
⊟ **TIME event**	
Time (0.1ms)	50
Whether to clear	True

2) Parameter content:

Project		**Content**	**Setting range**	**Default**
Execution type		Select event type	Not set/ON event/TIME event	Not set
ON event	Contact	The event type can be set when ON event is selected. Set the bit device as the trigger condition.	X/Y/M/SM/D.b	#
	Whether to clear	When the bit device of the trigger condition set by the ON event is turned OFF, whether to clear the current value of the output (Y) and timer (T) used in the execution program of the ON event in the next cycle.	True False	False
TIME event	Time	Set how long to execute the event program once.	1 to 2147483647 (100us unit)	#
	Whether to clear	When the TIME event is executed, if the event is not executed in the next scan cycle, select whether to clear the output (Y) used in the TIME event execution program and the current value of the timer (T).	True False	False

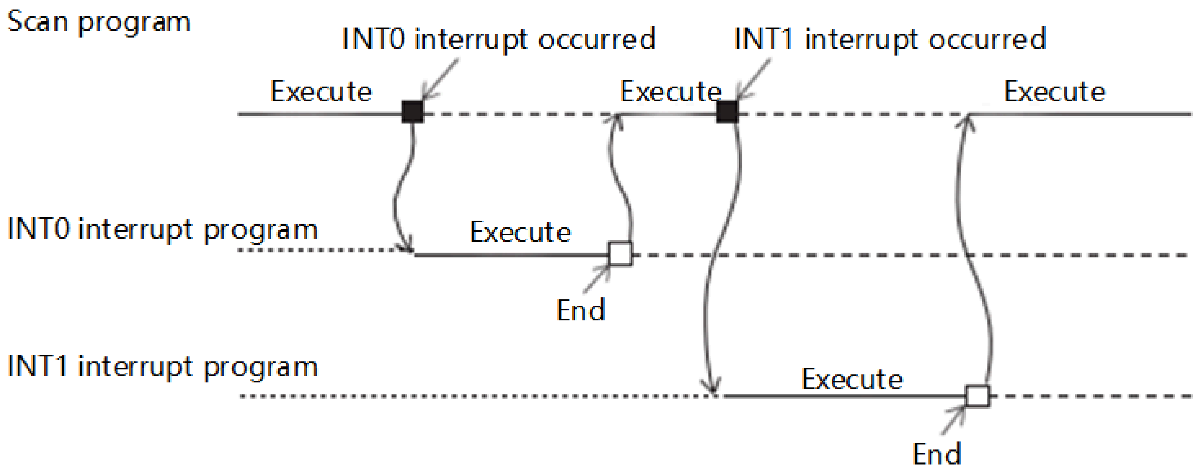
Key points

When "ON event" or "TIME event" is specified, if "Clear or not" is set to "Clear", the event program will not be executed in one scan cycle, and all the internal outputs (Y) and current value of timer (T) will all be cleared (except for the cumulative type and subroutine type T). If the time set by the TIME event is less than the scan period, it is equivalent to executing the TIME event every scan period. Even if the clear output is set, the output and timer data in the event program will not be cleared.

Interrupt execution program

In the process of executing the scan program, the program that can interrupt the priority execution of the scan program is called an interrupt execution program.

1. When an interrupt cause occurs, the interrupt program corresponding to the interrupt pointer number will be executed. However, the execution needs to be set to the interrupt enabled state by the EI instruction.



2. An interrupt name corresponds to an interrupt program, and the interrupt name cannot be repeated. Each interrupt has its own trigger condition and execution program, and each interrupt program ends with END.

3. Interrupt has the characteristic of interrupting the original execution program and executing the interrupt first, but it cannot interrupt the interrupt program being executed.

4. The interrupt program has the concept of priority. The smaller the priority value, the more priority the response. The priority setting range is 0 to 2.

The actions when an interruption cause occurs are as follows:

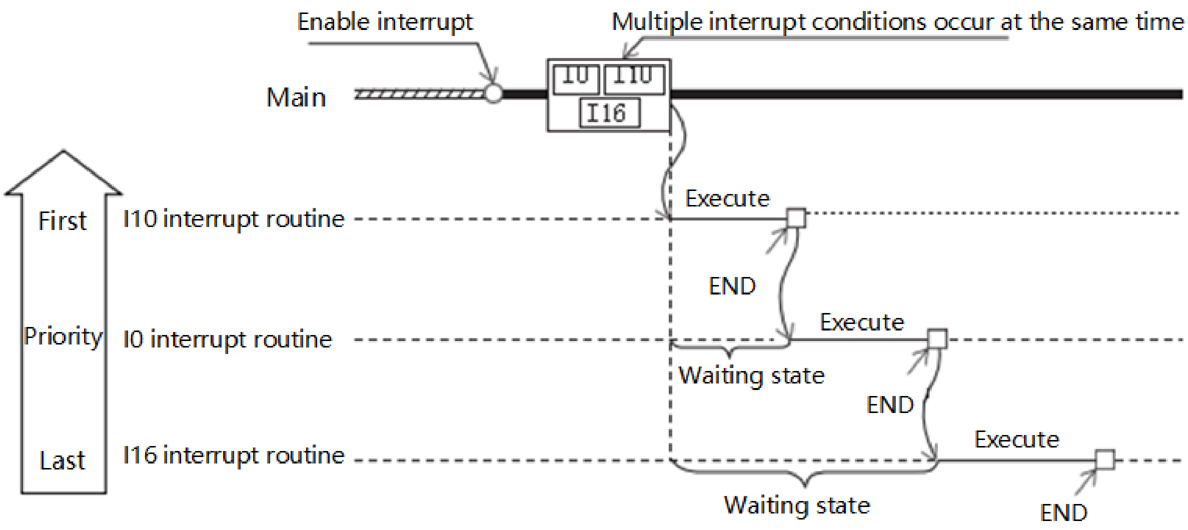
1. Interrupt prohibition (DI) when an interruption cause occurs.

If the interrupt execution condition is triggered in a program that is forbidden by DI, the interrupt will not be executed. Even if the subsequent program uses the EI instruction to allow interruption, the previously shielded interrupt program will not be executed. Only the interrupt execution condition is triggered again. The interrupt program will be executed.

2.When multiple interrupt causes occur simultaneously in the interrupt enabled state.

The interrupt program with higher priority will be executed sequentially. In addition, when multiple interrupts with the same priority occur at the same time, the actions are executed in the order of interrupt priority.

If three interrupt programs I0, I10, I16 are created, the priority of I0 is 1, the priority of I10 is 0, and the priority of I16 is 1. The execution logic is shown in the figure below: I10 has the smallest priority and is executed first; I0 and I16 have the same priority and are executed in the order of program establishment.



3. When an interrupt occurs during the waiting time when performing constant scan.

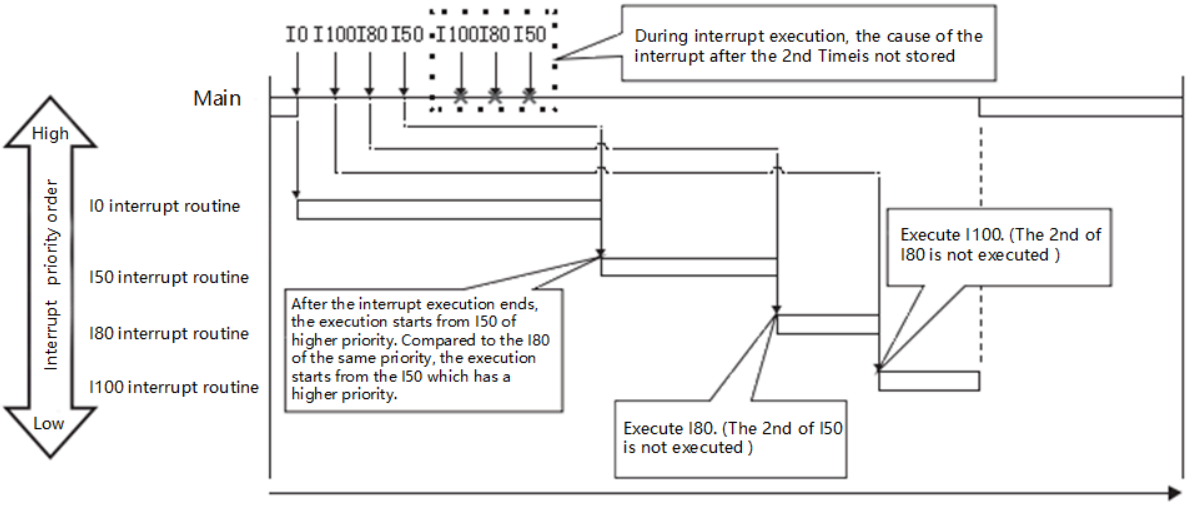
Execute the interrupt program for this interrupt.

4.When other interrupts occur during the execution of the interrupt program.

In the interrupt program (including the specification when the interrupt occurs in the event execution program), when other interrupts occur, the original interrupt execution program will not be interrupted. After the original interrupt execution program is completed, the new interrupt program is executed. After the execution is completed Then return to the scanning procedure.

5. During the execution of the interrupt program, when an interrupt cause with a low priority or the same priority occurs.

The interruption cause that occurred is stored, and after the interrupt program in execution ends, the interrupt program corresponding to the stored interruption cause is executed. Even if the same interruption cause occurs multiple times, the interruption cause is stored only once.



During interrupt execution, the cause of the interrupt after the 2nd Timeis not stored

After the interrupt execution ends, the execution starts from I50 of higher priority. Compared to the I80 of the same priority, the execution starts from the I50 which has a higher priority.

Execute I80. (The 2nd of I50 is not executed)

Execute I100. (The 2nd of I80 is not executed)

6. When the same interruption cause occurs during the execution of the interrupt program;

The interruption cause that occurred is stored, and after the interrupt program in execution ends, the interrupt program corresponding to the stored interruption cause is executed. Even if the same interruption cause occurs multiple times, the interruption cause is stored only once.

Interrupt trigger condition classification

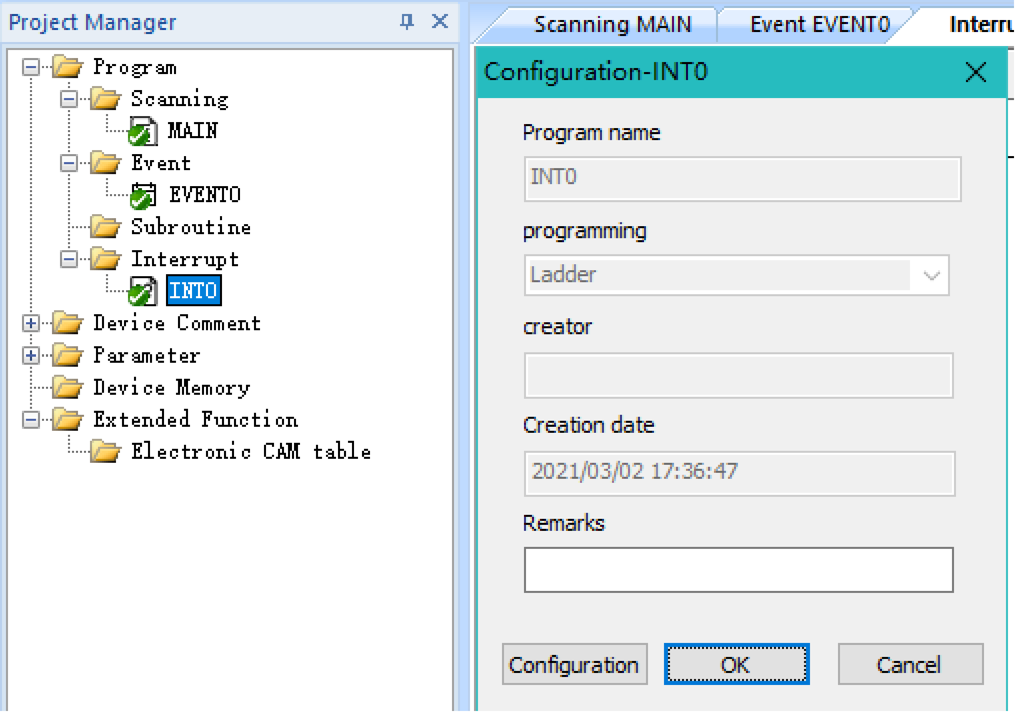
1. **External input (X) interrupt**

   1. Description of external input interrupt

1. The external input interrupt is triggered by the rising or falling edge of the fixed X point input.
2. Supports the rising and falling edge interrupts of a total of 8 input points of X0 to X7, and supports a total of 16 external input interrupts.
3. The same interrupt trigger condition cannot create multiple interrupt programs.
4. External input interrupt and high-speed counter cannot use the same X point.
5. You must use EI in the scan program to allow interrupts before the interrupt execution program will be executed.

   2. External input interrupt steps.

Project management→ Program→ Interrupt→ right click to create



Configuration-INT0

Program name: INT0

programming: Ladder

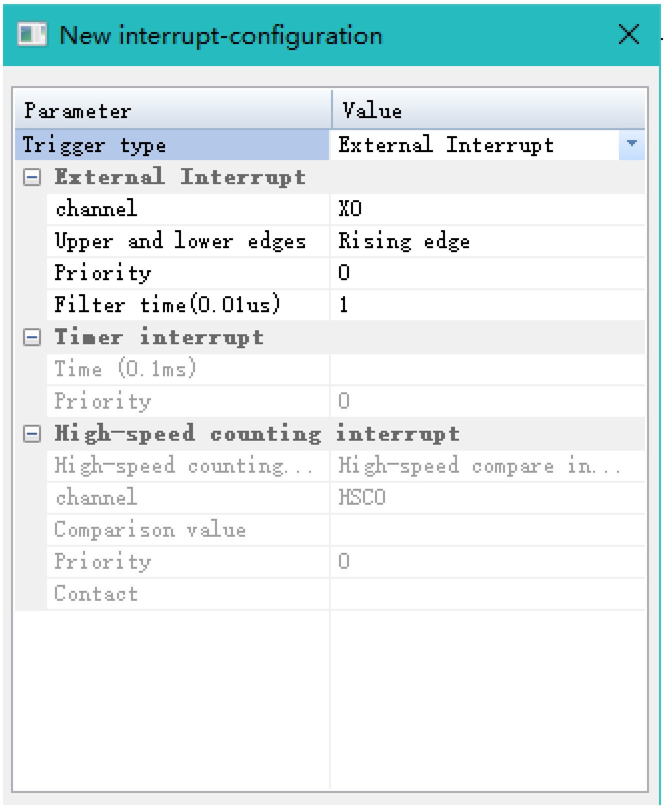
creator:

Creation date: 2021/03/02 17:36:47

Remarks:

1. The interrupt program name requires to match case, and the program name with the same name as the device cannot be used (the device name does not match case),
2. The interrupt program name does not support the input of /%$@&=to`^<>?:{}[],; !*.\\'" characters,
3. The length of the interrupt program name cannot exceed 64 characters and cannot be typed. The default name is INTx.

Click Configure, and select external interrupt for execution type, as shown in the figure below (it can also be configured in "program parameters" in "parameters" in project management):



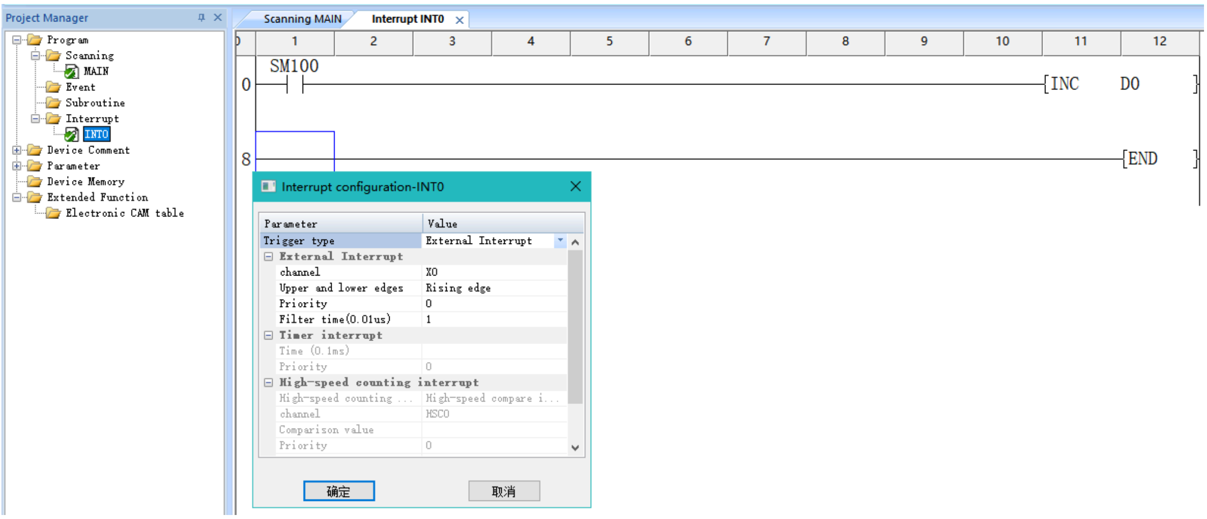
New interrupt-configuration

Parameter	Value
Trigger type	External Interrupt
⊟ External Interrupt	
channel	X0
Upper and lower edges	Rising edge
Priority	0
Filter time(0.01us)	1
⊟ Timer interrupt	
Time (0.1ms)	
Priority	0
⊟ High-speed counting interrupt	
High-speed counting...	High-speed compare in...
channel	HSC0
Comparison value	
Priority	0
Contact	

Project		Content	Setting range	Default
Execution type		Select the type of interrupt	Not set, External input interrupt, Timer event, high-speed counter interrupt	Not set
External input interrupt	channel	Select the channel for external input interrupt	X0 to X7	X0#
	Trigger edge type	Choose to trigger on rising edge or falling edge	Rising edge Falling edge	Rising edge
	priority	When multiple interrupts arrive at the same time, the order of priority execution, the smallest value is executed first	0 to 2	0
	Filter time (0.01us)	Set the filter time of X point, the unit is 0.01us. Note: X rising edge interrupt and X falling edge interrupt use the same X filter, so after the filter setting is changed in the X rising edge configuration, the X falling edge will also change. If the filter time is set to 1000, you must ensure that the high level and low level of the input signal are maintained for	0 to 1700	1

more than 10 us before
the interrupt can be
triggered.

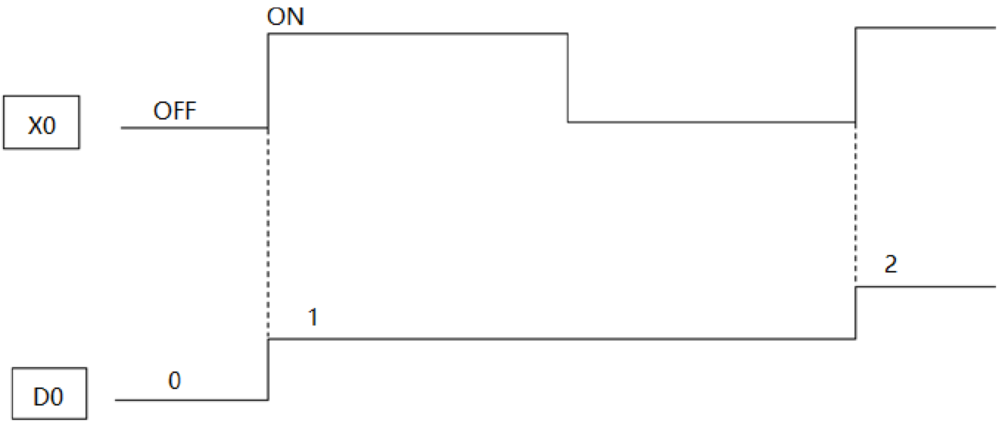
1. Write interrupt execution program



Interrupt configuration-INT0

Parameter	Value
Trigger type	External Interrupt
External Interrupt	
channel	X0
Upper and lower edges	Rising edge
Priority	0
Filter time(0.01us)	1
Timer interrupt	
Time (0.1ms)	
Priority	0
High-speed counting interrupt	
High-speed counting ...	High-speed compare i...
channel	HSC0
Comparison value	
Priority	0

Double-click the newly created interrupt program in the project management to start writing the interrupt execution program. As shown in the figure above, a newly created interrupt program is INT_X0_UP, and the trigger condition is configured to execute the interrupt program when the X0 rising edge is configured. If the EI instruction is used in the main program to allow interrupts, all programs in INT_X0_UP will be executed whenever X0 changes from OFF to ON , That is, D0 will increment once.



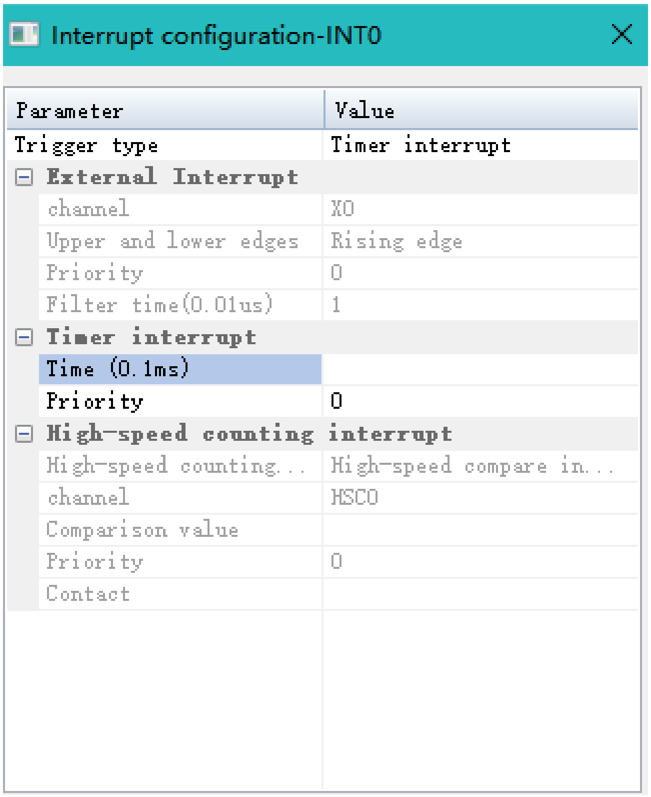
2.Timer interrupt

1. Timer interrupt description

1. Timer interrupt is based on the set time, execute the interrupt program every this time, the minimum time interval can reach 100us.
2. Up to 100 timer interrupt execution programs can be created.
3. Each timer interrupt program is independent of each other and does not affect each other.
4. Each timer interrupt program should be configured with priority. When triggered at the same time, it is executed in the order of priority, but when the priority is the same, it is executed in the order of the established program.
5. The interrupt execution program is executed only after EI is used in the scanner to allow the interrupt

2. Timer interrupt step

Project managementðProgramðInterruptðRight click to create. Enter the program name. The program name only supports the combination of English letters, numbers, and underscores, and must start with an English letter. The default is INTx. Click Configure and select Timer Interrupt as the execution type, as shown in the figure below (it can also be configured in "program parameters" in "parameters" in project management).

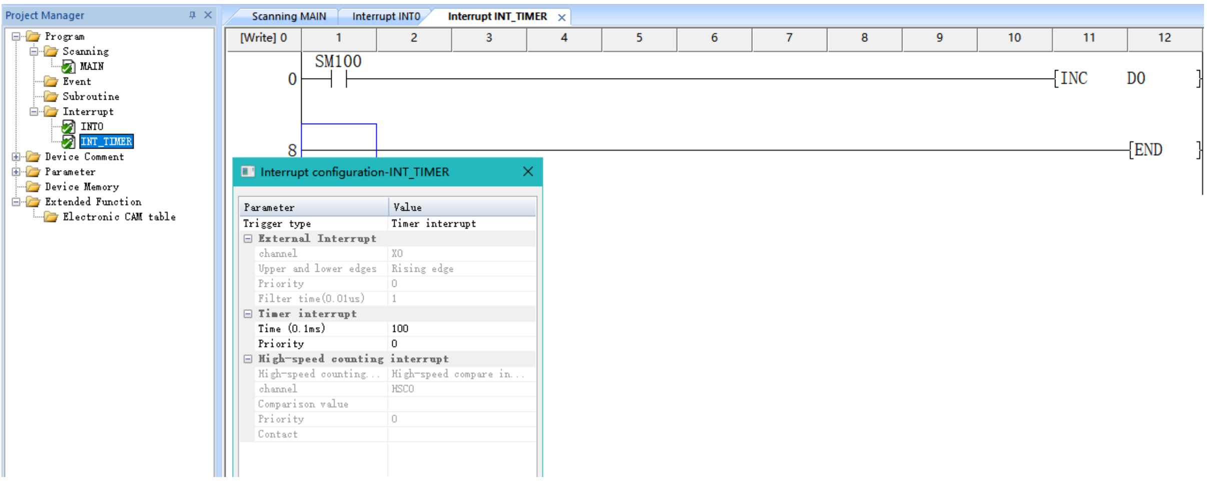


Interrupt configuration-INT0

Parameter	Value
Trigger type	Timer interrupt
⊟ External Interrupt	
channel	X0
Upper and lower edges	Rising edge
Priority	0
Filter time(0.01us)	1
⊟ Timer interrupt	
Time (0.1ms)	
Priority	0
⊟ High-speed counting interrupt	
High-speed counting...	High-speed compare in...
channel	HSC0
Comparison value	
Priority	0
Contact	

Project		Content	Setting range	Default
Execution type		Select the type of interrupt	Not set/External input interrupt/Timer event/ high-speed counter interrupt	Not set
Timer interrupt	Time	Set the interval time for interrupt triggering	1 to 2147483647 (100us unit)	
	priority	When multiple interrupts arrive at the same time, the order of priority execution, the smallest value is executed first	0 to 2	0

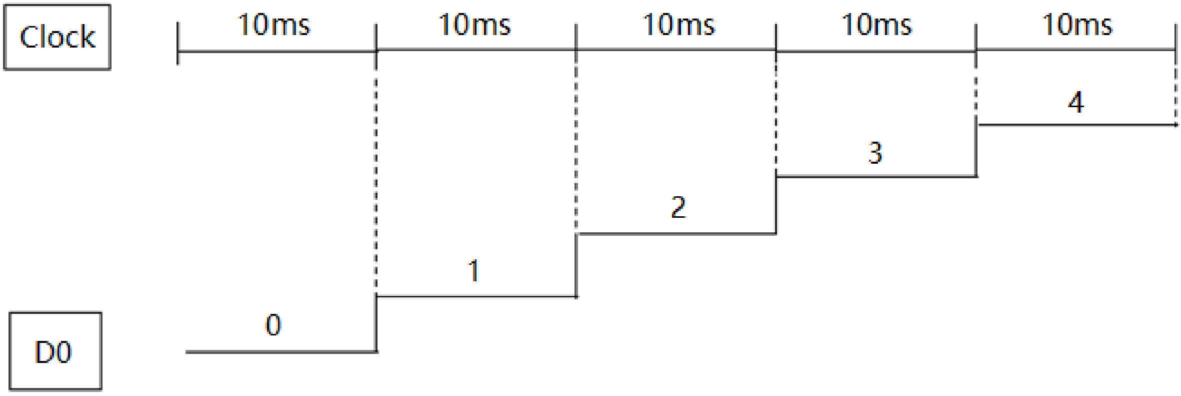
1. Write interrupt execution program



Interrupt configuration-INT_TIMER

Parameter	Value
Trigger type	Timer interrupt
⊟ External Interrupt	
channel	X0
Upper and lower edges	Rising edge
Priority	0
Filter time(0.01us)	1
⊟ Timer interrupt	
Time (0.1ms)	100
Priority	0
⊟ High-speed counting interrupt	
High-speed counting...	High-speed compare in...
channel	HSC0
Comparison value	
Priority	0
Contact	

Double-click the newly created timer interrupt program in the project management to start writing the interrupt execution program. As shown in the figure above, a newly created timer interrupt program is INT0, and the trigger condition is configured to execute the interrupt program every 10ms. If the main program uses EI to enable interrupts, all instruction programs in INT0 will be executed every 10ms, namely D0 It will add 1 to 10ms.



3. High-speed counter interrupt

1. Description of high-speed counter interrupt

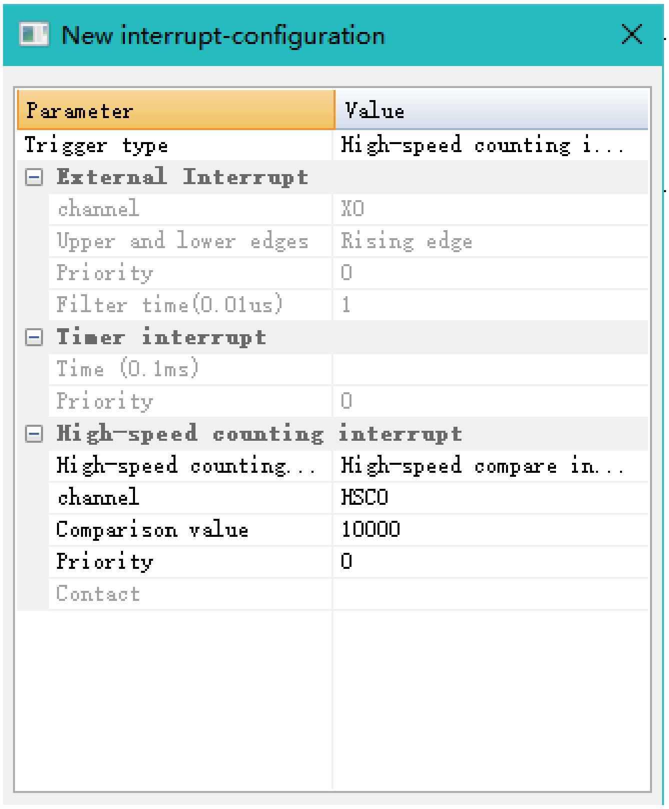
1. The high-speed counter interrupt triggers an interrupt condition after the set value of the high-speed counter HSC0 to HSC7 provided by the PLC and executes the interrupt program.
2. It can support up to 100 high-speed counter interrupt programs, but the number that can be supported by each channel does not need to be fixed.
3. When using the high-speed counter interrupt, project must configure the high-speed counter and use the OUT HSC instruction to enable the corresponding counting channel to count before it can be used (see the high-speed counter description section for the specific configuration method).
4. Each high-speed counter interrupt program should be configured with priority. When triggered at the same time, it will be executed in the order of priority. When the priority is the same, it will be executed in the order of channels HSC0-HSC7. When the channel is also the same, it is executed in order according to the creation promise.
5. Project must use EI in the scan program to allow interrupts before the interrupt execution program will be executed.

#Note: Both the HSC channel and the external input interrupt channel must use the PLC input point X. It should be noted that it

cannot be reused during configuration. For details, please refer to the configuration chapter of the high-speed counter.

1. high-speed counter interrupt step

“Project management”ð“Programð“Interrupt”ðRight click to create. Enter the program name. The program name only supports the combination of English letters, numbers, and underscores, and must start with an English letter. The default is INTx. Click Configure, select high-speed interrupt for execution type, as shown in the figure below (it can also be configured in "program parameters" in "parameters" in project management).



New interrupt-configuration

Parameter	Value
Trigger type	High-speed counting i...
⊟ **External Interrupt**	
channel	X0
Upper and lower edges	Rising edge
Priority	0
Filter time(0.01us)	1
⊟ **Timer interrupt**	
Time (0.1ms)	
Priority	0
⊟ **High-speed counting interrupt**	
High-speed counting...	High-speed compare in...
channel	HSC0
Comparison value	10000
Priority	0
Contact	

Project		**Content**	**Setting range**	**Default**
Execution type		Select the type of interrupt	Not set, External input interrupt, Timer event, high-speed counter interrupt	Not set
High count interrupt	Mode	Select the type of high-speed counter interrupt:	High-speed compare interrupt	High-speed compare interrupt
		(1) High-speed comparison interrupt: The interrupt program is executed after the trigger condition is reached.	High-speed comparison set High-speed comparison reset Not set	#
		(2) High-speed comparison setting: After reaching the trigger condition, the set contact is set.		
		(3) High-speed comparison reset: reset the set contact after reaching the trigger condition.		

Channel	Select the high-speed counter channel used	HSC0 to HSC7	HSC0
Comparison value	Set the comparison value of the high-speed counter. When the high-speed counter value of the set channel passes this value, the trigger condition is reached.	-2147483648 to 2147483647	
Priority	When multiple interrupts arrive at the same time, the order of priority execution, the smallest value is executed first	0 to 2	0
Contact	It is available when selecting high-speed comparison set and high-speed comparison reset. The contact is set or reset immediately after the trigger condition is reached.	Y/M/D.b	

3. Description of triggering rules

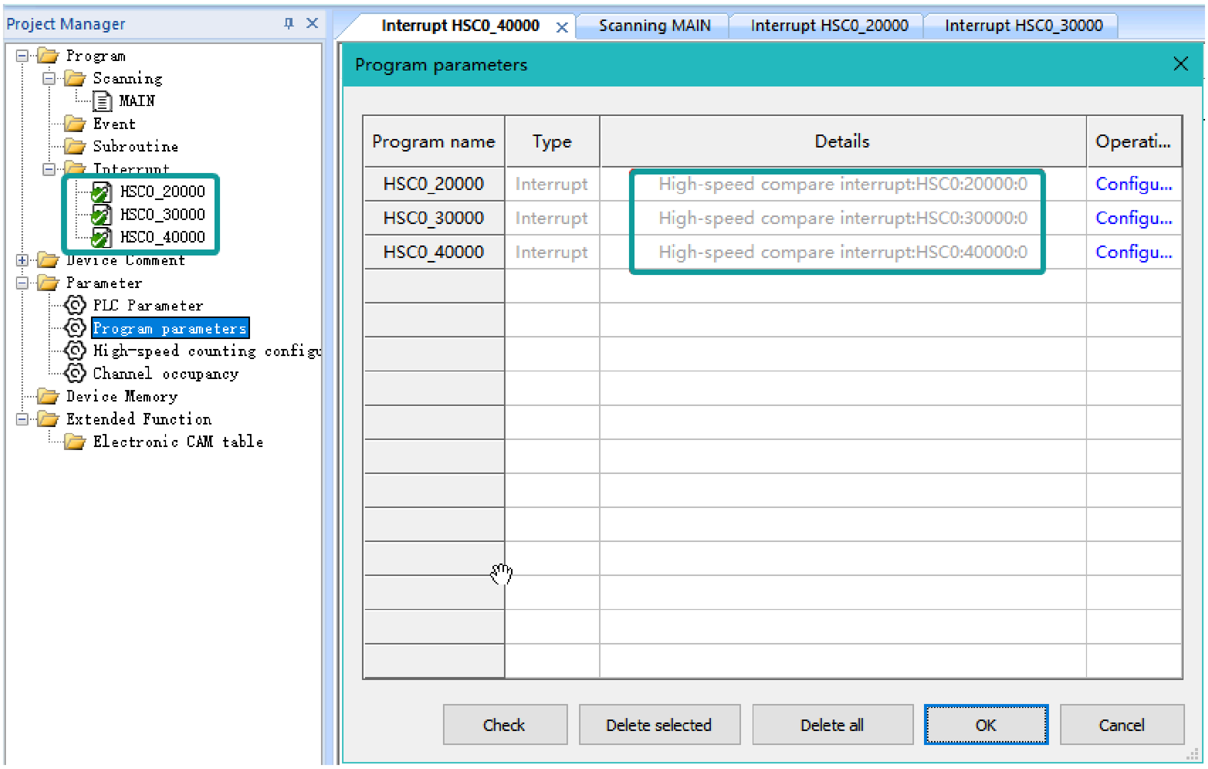
Mode	Configuration	The current value	Action
High-speed compare interrupt (INT0)	Comparison value: 10000	9999 → 10000	Execute all programs in interrupt INT0
		10001 → 10000	Execute all programs in interrupt INT0
High-speed comparison set (INT1)	Comparison value: -50,000 Contact: Y10	-50001 → -50000	Y10 is immediately set and mapped to the actual output (not affected by the scan period) The program in INT1 will not be executed
		-49999 → -50000	Y10 is immediately set and mapped to the actual output (not affected by the scan period) The program in INT1 will not be executed
High-speed comparison reset (INT2)	Comparison value: 400000 Contact: Y10	399999 → 400000	Y10 is reset immediately and mapped to the actual output (not affected by the scan period) The program in INT2 will not be executed
		400001 → 400000	Y10 is reset immediately and mapped to the actual output (not affected by the scan period) The program in INT2 will not be executed

#Note: Both HSC channel and external input interrupt channel need to use the INPUT point X, so it should be noted that it cannot be reused in configuration. For details, please refer to the configuration section of high-speed counter.

1. Write interrupt execution program

- New interrupt program

Create three new interrupt programs under the interrupt of project management, namely HSC0_20000, HSC0_30000, HSC0_40000. Configure the interrupt program in the "program parameters", as shown in the figure below.



Project Manager

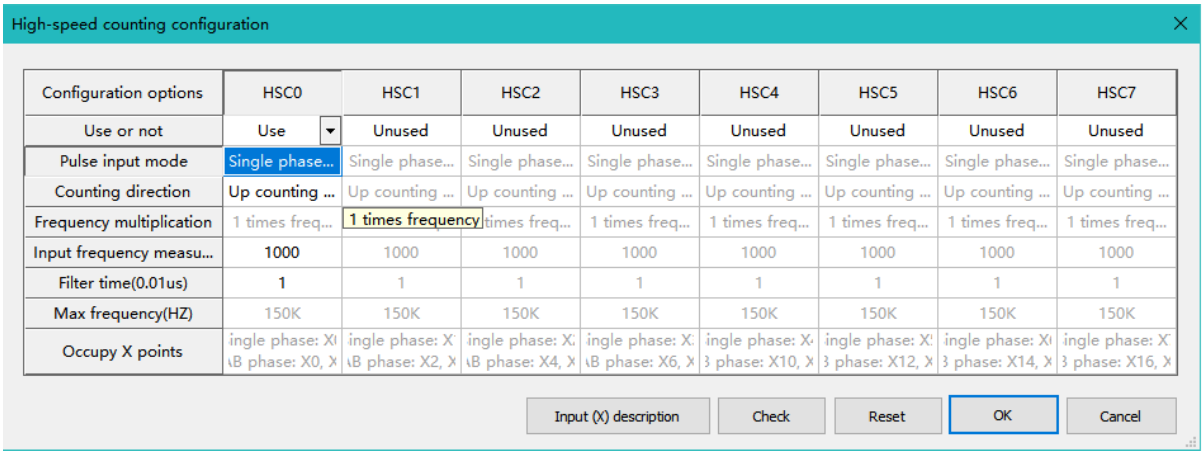
- Program
  - Scanning
    - MAIN
  - Event
  - Subroutine
  - Interrupt
    - HSC0_20000
    - HSC0_30000
    - HSC0_40000
- Device Comment
- Parameter
  - PLC Parameter
  - Program parameters
  - High-speed counting configu
  - Channel occupancy
- Device Memory
- Extended Function
  - Electronic CAM table

Program parameters

Program name	Type	Details	Operati...
HSC0_20000	Interrupt	High-speed compare interrupt:HSC0:20000:0	Configu...
HSC0_30000	Interrupt	High-speed compare interrupt:HSC0:30000:0	Configu...
HSC0_40000	Interrupt	High-speed compare interrupt:HSC0:40000:0	Configu...

- High-speed counter configuration

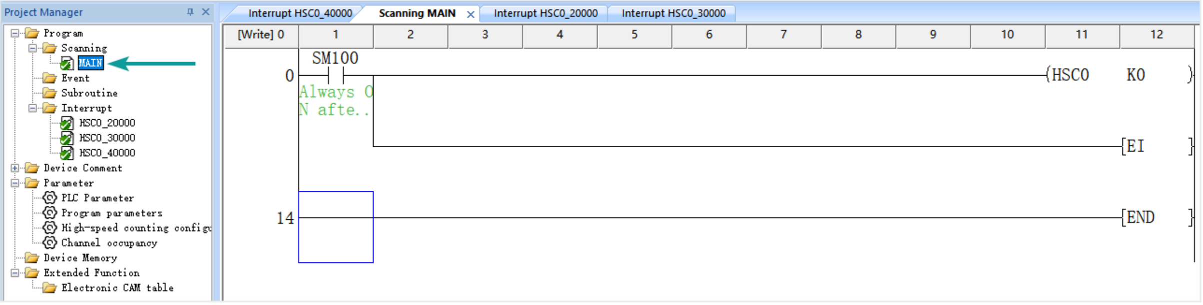
Configure HSC0 for use in the high-speed counter configuration. After selecting the working mode, click the "Check" button. After the correct configuration box pops up, click Enter.



High-speed counting configuration

Configuration options	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	HSC6	HSC7
Use or not	Use	Unused	Unused	Unused	Unused	Unused	Unused	Unused
Pulse input mode	Single phase...	Single phase...	Single phase...	Single phase...	Single phase...	Single phase...	Single phase...	Single phase...
Counting direction	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...	Up counting ...
Frequency multiplication	1 times freq...	1 times frequency	times freq...	1 times freq...	1 times freq...	1 times freq...	1 times freq...	1 times freq...
Input frequency measu...	1000	1000	1000	1000	1000	1000	1000	1000
Filter time(0.01us)	1	1	1	1	1	1	1	1
Max frequency(HZ)	150K	150K	150K	150K	150K	150K	150K	150K
Occupy X points	ingle phase: X B phase: X0, X	ingle phase: X B phase: X2, X	ingle phase: X B phase: X4, X	ingle phase: X B phase: X6, X	ingle phase: X phase: X10, X	ingle phase: X phase: X12, X	ingle phase: X phase: X14, X	ingle phase: X phase: X16, X

Call the high-speed counter in the main program and enable interrupts:



Program operation:

Assuming that the High-speed counter channel 0 has been receiving pulses:

When the count value of HSC0 accumulates from 0 to 20000, all procedures of HSC0_20000 are executed.

When the count value of HSC0 is accumulated from 20000 to 30000, all procedures of HSC0_30000 are executed.

When the count value of HSC0 is accumulated from 30000 to 40000, all procedures of HSC0_40000 are executed.

Mask interrupt

1. Mask through application instructions

The PLC interrupt is in the shielded state by default when it is powered on, and can only be used after the interrupt is allowed through the EI instruction.

The interrupt mask instruction DI masks all interrupts without parameters, and masks some priority interrupts with parameters (refer to the program flow instruction DI/EI for details).

2. Mask through special registers SM and SD

1.External input interrupt mask register

External input interrupt mask register

Special register number	Type of interrupt	Instruction	Defaults
SM352	X0 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM353	X0 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM354	X1 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM355	X1 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM356	X2 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM357	X2 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM358	X3 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM359	X3 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM360	X4 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM361	X4 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF

SM362	X5 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM363	X5 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM364	X6 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM365	X6 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM366	X7 rising edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF
SM367	X7 falling edge interrupt	ON: Mask interrupt OFF: Enable interrupt	OFF

2. Timer interrupt mask register

Timer interrupt mask register

Special register number	Type of interrupt	Instruction	Default
SD350	1st to 16th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD351	17th to 32th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD352	33th to 48th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD353	49th to 64th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD354	65th to 80th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD355	81st to 96th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD356	97th to 100th timer interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0

1. high-speed counter interrupt mask register

High-speed counter interrupt mask register

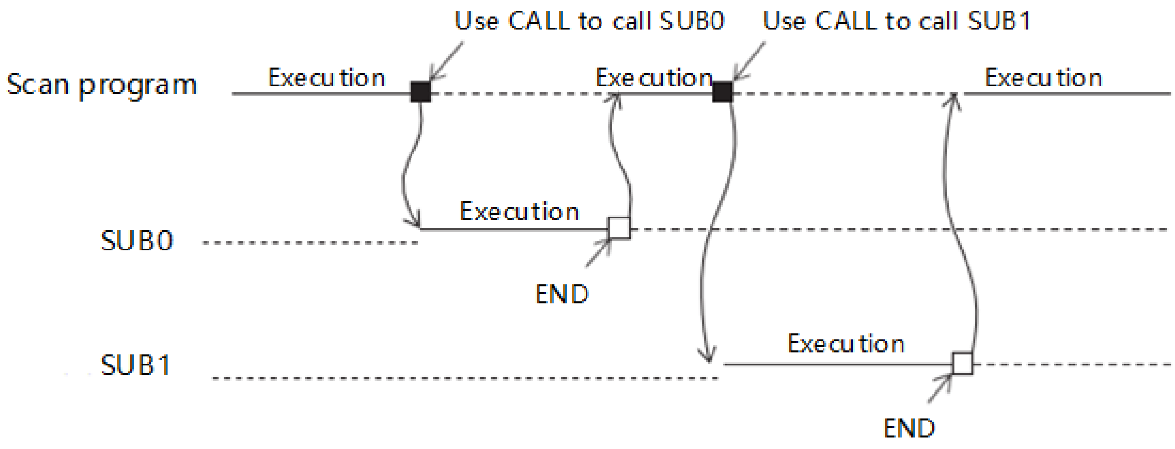
Special register number	Type of interrupt	Instruction	Default
SD382	1st to 16th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0

SD383	17th to 32nd high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD384	33th to 48th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD385	49th to 64th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD386	65th to 80th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD387	81st to 96th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0
SD388	97th to 100th high-speed counter interrupt	Each bit can control the mask of an interrupt. ON: Mask interrupt OFF: Enable interrupt	0

Subroutine

During the execution of the scan program, the executed program can be called by the CALL instruction.

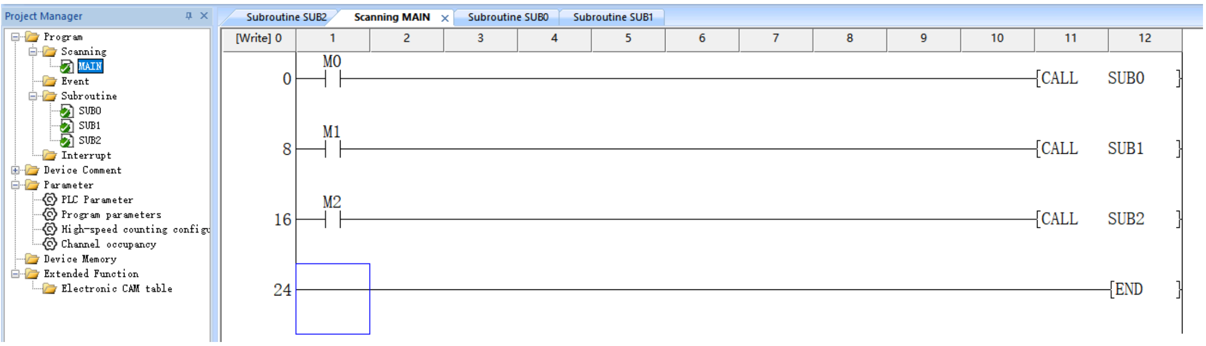
A subroutine is to split a certain module in the main program for the main program to call, which is conducive to the modularization of the program. Such as other high-level language functions, but this function has no parameters and no return value.



1. Instructions for calling subroutines

After a new subroutine is created, the content of the program is not executed. It is executed only when the CALL(P) instruction is used to call the subroutine in the scan, event, and interrupt programs, and the call is executed once. Three new subroutines SUB0, SUB1, SUB2 are created as shown in the figure below. In the main program MAIN, the subprogram can be called by using the CALL(P) subprogram program name.

Through this programming method, the use of the same logic program for different conditions can reduce the number of Circuit program steps and improve the readability of the Circuit program.



```
 0  M0  [CALL  SUB0 ]
 8  M1  [CALL  SUB1 ]
16  M2  [CALL  SUB2 ]
24      [END ]
```

1.#Note:

1. When using the timer (OUT T), note that the output will not be reset when the subroutine is not called, and a specific subroutine register must be used.
2. It is not allowed to call recursively between subprograms, that is, call SUB1 in SUB0, and then call SUB0 in SUB1. This is not allowed.
3. The subroutine can be nested up to 32 levels. If the level exceeds 32 levels, a serious error will be reported and the Circuit program operation will be forcibly stopped.
4. Unlike the LX3V series mainframe, the subroutine in the LX5V series mainframe ends with the END instruction instead of SRET.

Positioning instructions

1. Event

1.ON event

If the high-speed pulse instruction is turned on during the ON event, the high-speed pulse instruction will be sent as normal. If the ON contact of the trigger event in the scan period is turned OFF during the pulse sending, select whether to continue sending the pulse or stop the pulse according to the unscanned processing flag bit.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, continue to send pulses until it stops. At this time, it should be noted that if the trigger event OFF contact turns ON after the pulse is sent, the pulse will be sent again.

When the flag bit is 1 (stop sending pulses), if the trigger event ON contact turns OFF in a certain scan period, it will decelerate and stop.

2. TIME event

If the high-speed pulse instruction is turned on in the TIME event, the high-speed pulse instruction will be sent as normal. If the instruction is not scanned in a certain scan period during the pulse transmission, select whether to continue sending the pulse or stop the pulse according to the unscanned processing flag bit.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, the pulse will continue to be sent until it stops. In the TIME event, it is impossible to ensure that the instruction is scanned in every scan cycle, so you should avoid using high-speed pulse instructions in the TIME time, otherwise the pulse will be sent again after the pulse is sent.

When the flag bit is 1 (stop sending pulses), if the instruction is not scanned in the current scan cycle, it will decelerate and stop. In the TIME event, if the flag bit is set to 1 (stop sending pulses), there will be no pulse sending.

2. Subroutine

If the high-speed pulse instruction is turned on in the subroutine, the high-speed pulse instruction will be sent as normal. If the scanning period is closed during pulse sending, select whether to continue sending or stop the pulse according to the unscanned processing flag.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, the pulse will continue to be sent until it stops. At this time, it should be noted that if the subroutine is called again after the pulse is sent, the pulse will be sent again.

When the flag bit is 1 (stop sending pulses), if the subroutine is closed during high-speed pulse sending, the speed will decelerate and stop. If the subroutine is closed before sending the pulse, then no pulse is sent.

3. Interrupt

1. External interrupt

If the high-speed pulse instruction is enabled in the external interrupt, the high-speed pulse instruction selects whether the pulse continues to be sent or the pulse stops according to the unscanned processing flag bit.

Output shaft	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), continue to send high-speed pulses until it stops.

When the flag bit is 1 (stop sending pulse), the high-speed pulse decelerates and stops.

2. Timer interrupt

If the high-speed pulse instruction is turned on in the timer interruption, the high-speed pulse instruction is sent as normal. If the instruction is not scanned in a certain scan period in the pulse transmission, the pulse continues to be sent or the pulse stops is selected according to the unscanned processing flag.

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), if the instruction is not scanned in the current scan cycle, continue to send pulses until it stops. In the timer interrupt, it is impossible to ensure that the instruction is scanned in every scan cycle, so it is necessary to avoid using high-speed pulse instructions in the T timer interrupt. Otherwise, after the pulse transmission is completed, the pulse will be sent again.

When the flag bit is 1 (stop sending pulses), if the instruction is not scanned in the current scan cycle, it will decelerate and stop. In the TIME event, if the flag bit is set to 1 (stop sending pulses), there will be no pulse sending.

3.High-speed comparison interrupt

If the high-speed pulse instruction is enabled in the high-speed comparison interrupt, the high-speed pulse instruction selects whether the pulse continues to be sent or the pulse stops according to the unscanned processing flag.

Output axis	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Not scanned flag bit	SM899	SM959	SM1019	SM1079	SM1139	SM1199	SM1259	SM1319

When the flag bit is 0 (continue to send pulse), continue to send high-speed pulses until it stops.

When the flag bit is 1 (stop sending pulse), the high-speed pulse decelerates and stops.