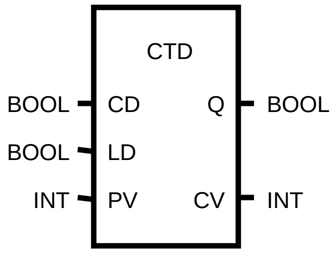
8.2 Function block libraries

1. Each function block is described.
2. Before using the function block, please review general description on function block in 3.5.2 and apply the function block libraries to the program for easy programming.

CTD

Down counter

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
 CTD BOOL CD Q BOOL BOOL LD INT PV CV INT	**Input** CD : Down_Count pulse input LD : Load preset value PV : Preset Value **Output** Q : Count_Down output CV : Current Value

■ Function

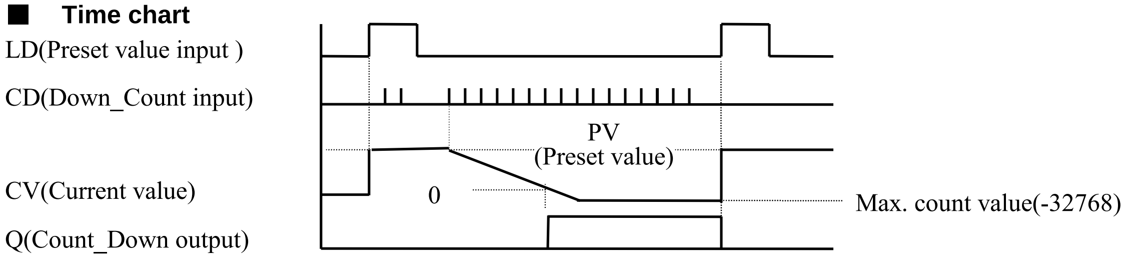
□ Down counter function block CTD decrease CV by 1 when down counter pulse input CD changes 0 to 1.

□ However, CV is decreased when it is greater than -32768 of minimum INT value and is not decreased when it becomes -32768.

□ When LD is 1, PV is load to CV.(CV=PV)

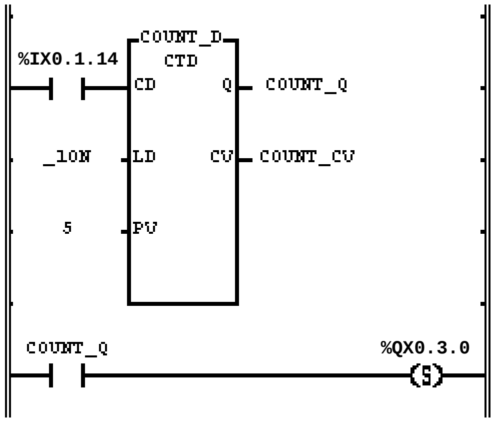
□ Output Q is 1 when CV is less than 0.

■ Time chart



■ Program example

Program that set the output contact %QX0.3.0 when 5 pulse inflows to input contact %I0.1.14.

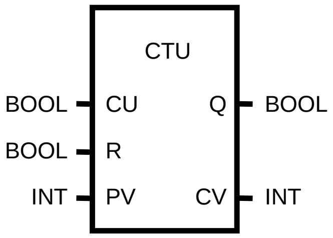
LD	IL
 COUNT_D CTD %IX0.1.14 CD Q COUNT_Q _1ON LD CV COUNT_CV 5 PV COUNT_Q %QX0.3.0 S	CAL CTD COUNT_0 CD %IX0.1.14 LD _1ON PV 5 LD COUNT_D.Q ST COUNT_Q LD COUNT_D.CV ST COUNT_CV LD COUNT_Q S %QX0.3.0

(1) Register name of CTD function block.(COUNT_D)
(2) Input the input contact %IX0.1.14, which the pulse input inflows, to CD.
(3) Input User Flag_ON(first scan on) that loads PV to CV.
(4) Input 5 to PV value within INT range(-32768~32767).
(5) Input arbitrary output variable(COUNT_CV) to CV.
(6) Input arbitrary output variable(COUNT_Q) to Q.
(7) After completing the program, execute the compile and write to PLC.
(8) Execute the mode transit(Stop → Run) after completing writing.
(9) 5 of PV value is load to CV(Count_CV) when the program runs.
(10) Reduce CV(COUNT_CV) by 1 when input pulse inflows to input contact %I.0.1.14.
(11) When No.5 pulse inflows to the input contact, CV will be 0 and Q (COUNT_Q) will be 1.
(12) When Q (COUNT_Q) is 1, output contact %Q0.3.0 is set.

CTU

Up counter(function block)

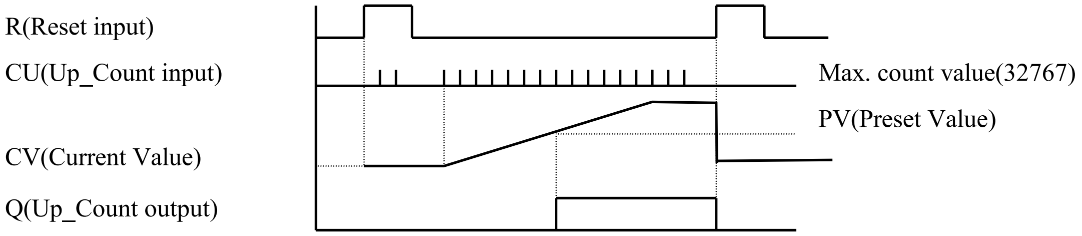
Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
 CTU BOOL CU Q BOOL BOOL R INT PV CV INT	**Input** CU : Up_Count pulse input R : Reset input PV : Preset Value **Output** Q : Up_Count output CV : Current Value

■ Function

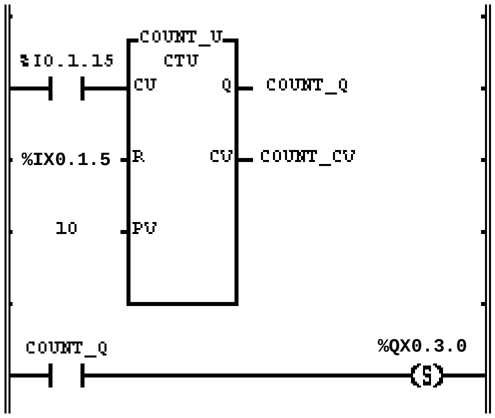
- □ Up counter function block CTU increase CV by 1 when up counter pulse input CU changes 0 to 1.
- □ However, CV is increased when it is less than 32767 of maximum INT value and is not increased when it becomes 32767.
- □ When reset input R is 1, CV is cleared(0).
- □ Output Q is 1 when CV is larger than PV.
- □ Every time when CTU executed, PV is reloaded into CTU instance memory and CTU uses this value for operation.

■ Time chart



■ Program example

Program that set the output contact %QX0.3.1 when 10 pulse inflows to input contact %IX0.1.15.

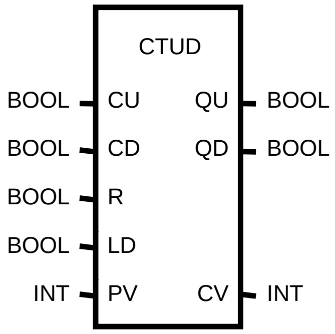
LD	IL
 COUNT_U CTU %I0.1.15 CU Q COUNT_Q %IX0.1.5 R CV COUNT_CV 10 PV COUNT_Q %QX0.3.0 S	CAL CTU COUNT_U CU %IX0.1.15 R %IX0.1.5 PV 10 LD COUNT_U.Q ST COUNT_Q LD COUNT_U.CV ST COUNT_CV LD COUNT_Q S %QX0.3.0

(1) Register name of CTU function block.(COUNT_D)
(2) Input the input contact %I0.1.15, which the pulse input inflows, to CU.
(3) Input 10 to PV.
(4) Set arbitrary input contact to R that initializes CV(%I0.1.5).
(5) Input arbitrary variable(COUNT_CV) to CV.
(6) Input arbitrary output variable(COUNT_Q) to Q.
(7) After programming, execute the compile and write to PLC.
(8) Execute the mode transit(Stop → Run) after completing writing.
(9) Current value CV(Count_CV) is increased by 1 when the input pulse inflows to input contact %I0.1.15.
(10) As current value CV is 10 and same to preset value when No.10 pulse inflows to input contact, the output Q(COUNT_Q) will be 1.
(11) When Q (COUNT_Q) is 1, output contact %Q0.3.0 is set.

CTUD

Up/Down counter(Function block)

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
 CTUD BOOL CU QU BOOL BOOL CD QD BOOL BOOL R BOOL LD INT PV CV INT	**Input** CU : Up_Count pulse input CD : Down_Count pulse input R : Reset input LD : Preset Value load PV : Preset Value **Output** QU : Count_Up output QD : Count_Down output CV : Current Value

■ Function

□ Up/Down counter function block CTUD increase CV by 1 when up counter pulse input CU changes 0 to 1 and CV is decreased by 1 when down counter pulse input CD changes 0 to 1.
CV is increased or decreased when CV is between minimum value -32768 and maximum value 32767, but CV and reaches minimum or maximum value, CV will not be increased or decreased.

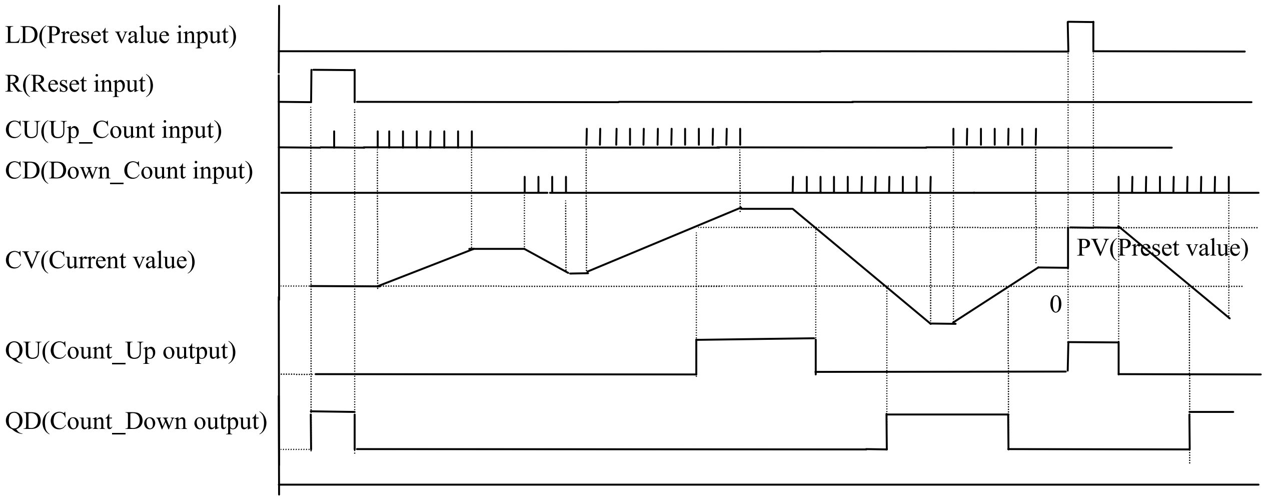
□ When LD is 1, preset value PV is loaded into current CV(CV=PV).

□ When R is 1, current value CV is cleared(0)(CV=0).

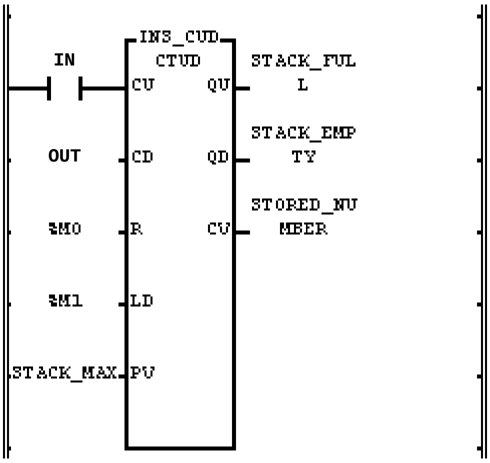
□ Output QU will be 1 when CV is greater than PV and QD will be 1 when CV is less than 0.

□ The operation for each input signal is executed in order of R > LD > CU > CD and, if the signal is duplicated, the operation of higher priority is executed.

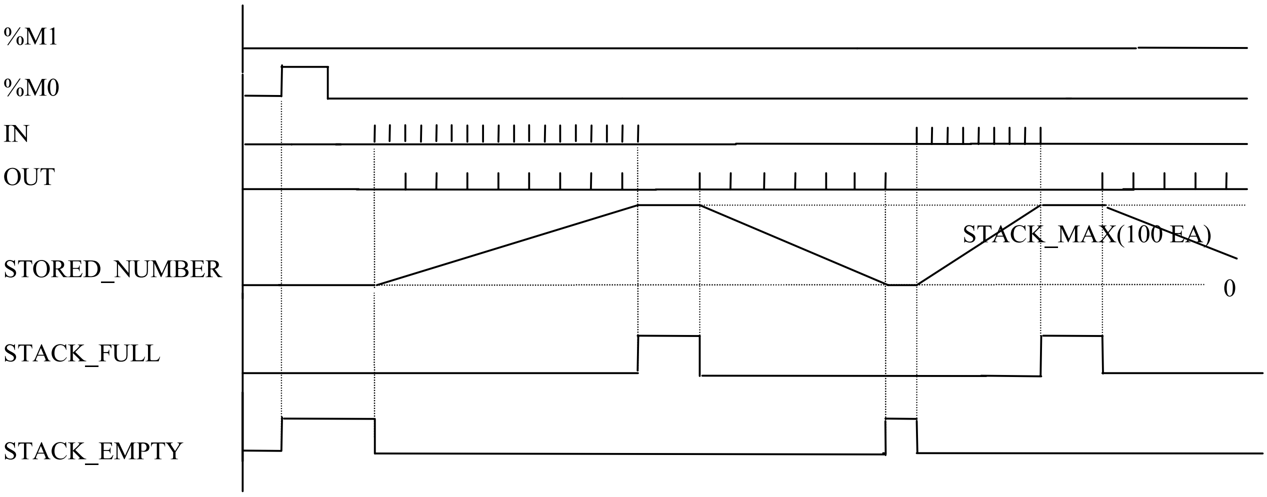
■ Time chart



■ Program example

LD	IL
 INS_CUD CTUD IN CU QU STACK_FUL L STACK_EMP TY OUT CD QD STORED_NU MBER %M0 R CV %M1 LD STACK_MAX PV	CAL CTUD INS_CUD CU:= IN CD:= OUT R := %M0 LD:= %M1 PV:= STACK_MAX LD INS_CUD.QU ST STACK_FULL LD INS_CUD.QD ST STACK_EMPTY LD INS_CU.CV ST STORED_NUMBER

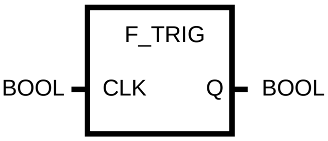
There is temporary storage stack that have storage capacity 100 (STACK_MAX) and IN signal is 1 whenever the material is loaded into stack and OUT signal is 1 whenever unloaded from stack in the production line. When material is loaded in to stack faster than unloading, STACK_FULL may 1 when STORED_NUMBER reaches 100(STACK_MAX). On the contrary, stack has no materials in it, STACK_EMPTY is 1.



F_TRIG

Drop edge detection(Function block)

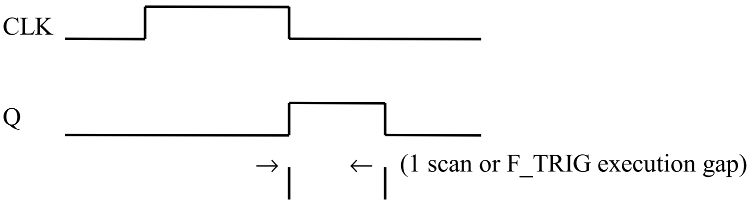
Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
	**Input** CLK : Input signal **Output** Q : Falling edge detection result

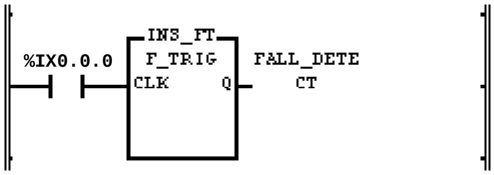
Function

F_TRIG outputs 1 to output Q when input to CLK changes from 1 to 0 and output 0 to output Q at next execution. Otherwise, output Q is always 0.

Time chart



Program example

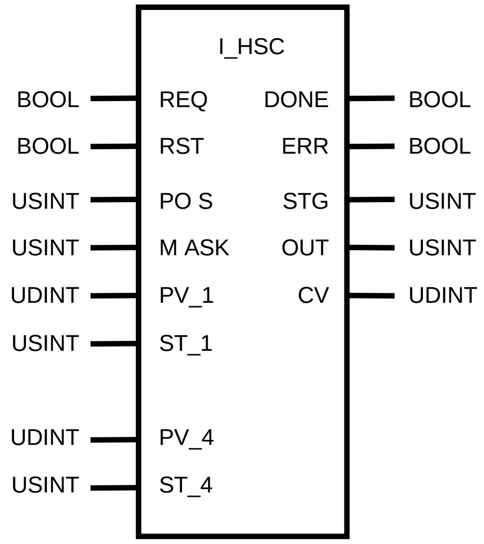
LD	IL
	CAL F_TRIG INS_FT CLK:= %IX0.0.0 LD INS_FT.Q ST FALL_DETECT

Monitor the status of input variable %I0.0.0 and output 1 to output variable FALL_DETECT when input variable %IX0.0.0 changes from 1 to 0. Output 0 to FALL_DETECT.

I_HSC

Internal high speed counter (Function Block)

Product	GM1	GM2	GM3	GM4	GM5
Applicable					●

Function	Description
 I_HSC BOOL REQ DONE BOOL BOOL RST ERR BOOL USINT PO S STG USINT USINT M ASK OUT USINT UDINT PV_1 CV UDINT USINT ST_1 UDINT PV_4 USINT ST_4	**Input** REQ : Counter operation input RST : Counter reset input POS : Direct output position selection MASK : Output contact selection of the module PV_1 : Stage 1 Preset Value ST_1 : Stage 1 output (Status) Stage range is from 1 to 4. **Output** DONE : Output '1' when final stage is completed ERR : Error display STG : Stage under count OUT : Preset value of counting stage CV : Current value of counting stage

■ Function

□ 1 phase, 1 point, 32 bit high-speed counter is installed in the main module of GM5. The contact utilizes HSC+ and HSC- of main module terminal. The specification is as below.

Voltage	: 10V~24V
Input current	: 16mA(24V), 6.7mA(10V)
Min. OFF voltage	: 2V
Count speed	: 20K pps

□ REQ : Register the count preset value of each stage to internal counter at the change of input condition from '0' to '1' and the operation continues when REQ is '0' after operating of counter and the counter will not operate again though the input condition changes from '0' to '1'. For the restart, execute RESET and satisfy the REQ condition.

□ RST : Remove the output in case of '1' and register again the counter value. If RST = '0' and REQ = '0' → '1', register the preset value and start the count again.

□ POS : Select the module position to output stage output value directly. In case of %QB0.1.0, digit '1' will be the position number of module. If the contact output section is not at the selected module, ERR will be '1' and FB will not operate. Lower 8 point of assigned module will be used.

□ MASK : Select actual output contact among 8 selected contact by POS. Though 8 output contacts are set by POS, the contact, which MASK selects '1', is used for actual output contact. For example, if POS=0 and MASK=16#0F, actual output contact will be lower 4 point of No. 0 module.

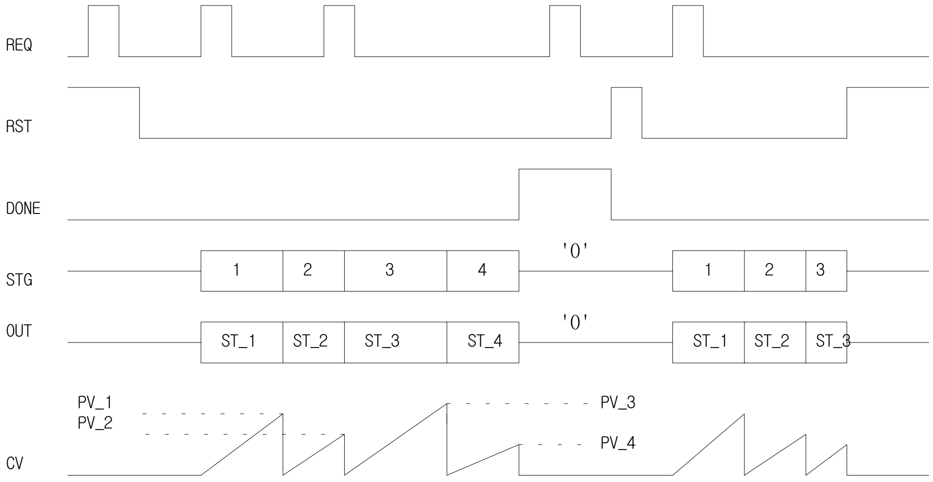
□ ERR : Display error status of FB.

□ STG : Display the current counting stage.

□ OUT : Output the selected output value(ST_x) of current counting stage.

If REQ condition is satisfied, start the count from 0 and output ST_1 value to OUT till meet the preset value of PV_1. When the count becomes the preset value PV_1, initialize the counter to next preset value(PV_2) and output ST_2 value. After executing last step that PV_x is set over 0, output '1' to DONE and '0' to OUT and wait for the restart condition.

■ Time chart

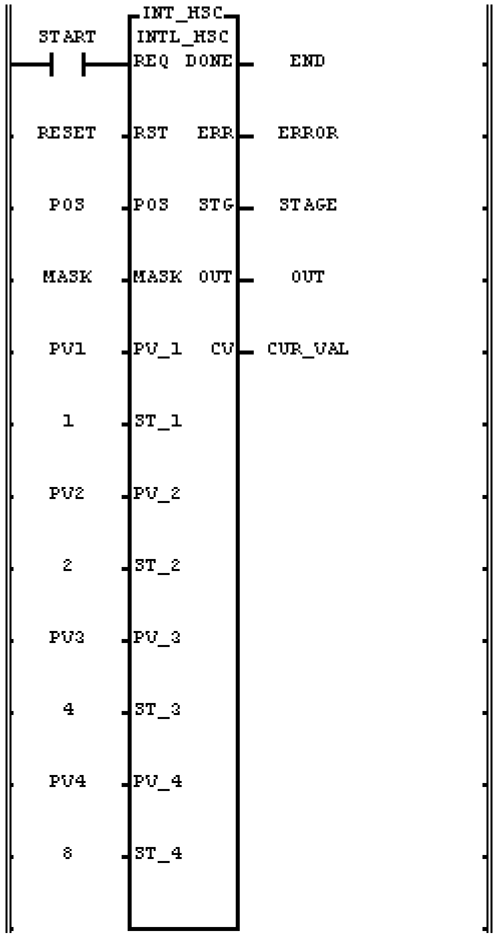


■ Program example

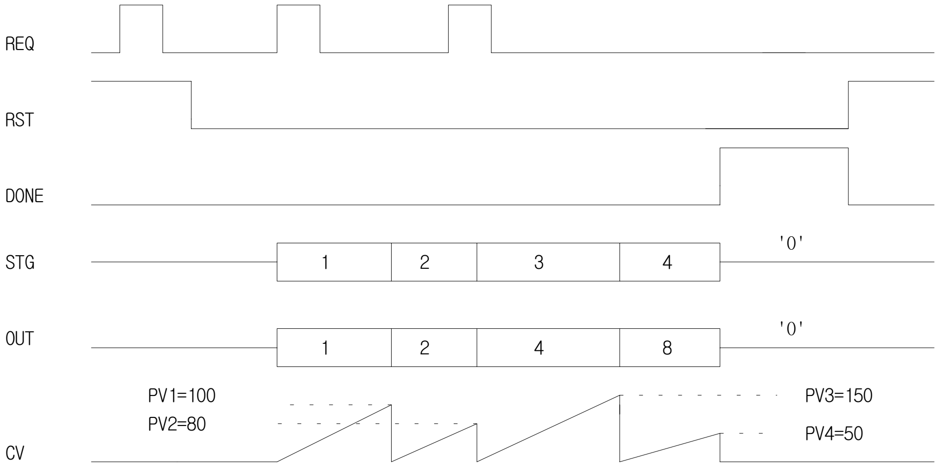
When PV1=100, PV2=80, PV3=150 and PV4=50, '0' valve is on in the district from 0 to PV1 of high-speed counter, '2' valve is on in the district from PV2 to PV3 of high-speed counter and '3' valve is on in the district from PV3 to PV4 of high-speed counter. The output preset value can be used for the input of other function or function block in other program. The program example is the case that the output contact and value is set as below.

VALVE : %QB0.0.0	=>	POS = 0
VALVE[0] : %QX0.0.0		
VALVE[1] : %QX0.0.1		
VALVE[2] : %QX0.0.2	=>	MASK = 16#0F
VALVE[3] : %QX0.0.3		

LD	IL

	CAL I_HSC INT_HSC
	REQ := START
	RST := RESET
	POS := POS
	MASK:= MASK
	PV_1:= PV1
	ST_1:= 1
	PV_2:= PV2
	ST_2:= 2
	PV_3:= PV3
	ST_3:= 4
	PV_4:= PV4
	ST_4:= 8
	LD INT_HSC.DONE
	ST END
	LD INT_HSC.ERR
	ST ERROR
	LD INT_HSC.STG
	ST STAGE
	LD INT_HSC.OUT
	ST OUT
	LD INT_HSC.CV
	ST CUR_VAL

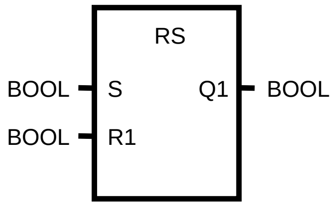
The time chart is as below.



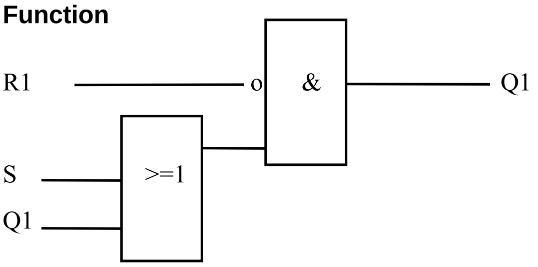
RS

Reset Priority Bistable(Function block)

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

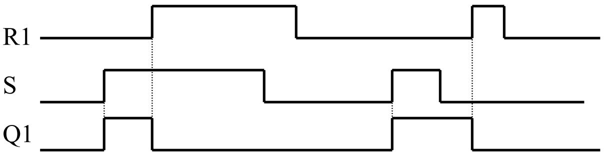
Function	Description
	**Input** R1 : Reset condition S : Set condition **Output** Q1 : Operation result

■ Function

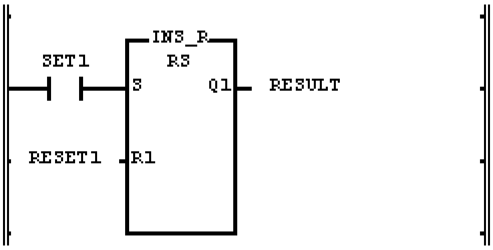


If R1 is 1, the output Q1 shall be always 0 regardless of S.
The output Q1 holds previous status and becomes 1 when R1 is 0 and S is 1.
The initial status of Q1 is 0.

■ Time chart



■ Program example

LD	IL
	CAL RS INS_R S: = SET1 R1: = RESET1 LD INS_R.Q ST RESULT

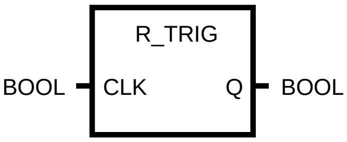
Take RESET1 as Reset condition and SET1 as Set condition and output the result to RESULT.
For the operation result, replace R1 by RESET1 and S by SET1 and Q1 by RESULT in the time chart.

(1) When the input variable SET1 and RESET1 is on simultaneously, the output variable RESULT will be 1.
(2) When the input variable RESET1 is on, the output variable RESULT will be 0.
(3) When the input variable SET1 and RESET1 is on simultaneously, the output variable RESULT will be 0.

R_TRIG

Rising edge detection (Function Block)

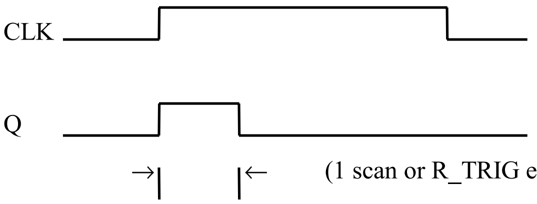
Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
	**Input** CLK : Input clock **Output** Q : Rising edge detection result

Function

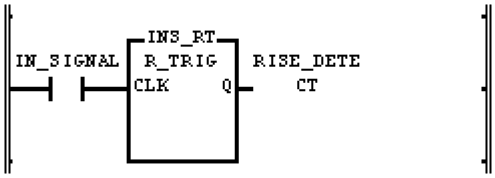
R_TRIG changes output Q to 1 when the input to CLK is changing from 0 to 1 and changes it to 0 at next execution.

Time chart



(1 scan or R_TRIG execution gap)

Program example

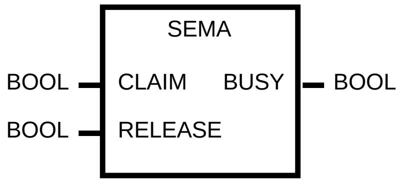
LD	IL
 INS_RT IN_SIGNAL R_TRIG CLK Q RISE_DETE CT	CAL R_TRIG INS_RT CLK: = IN_SIGNAL LD INS_RT.Q ST RISE_DETECT

(1) Output 1 to RISE_DETECT when IN_SIGNAL is changing from 0 to 1 and 0 to RISE_DETECT during INS_RT execution.

SEMA

Semaphore for system resource control(Function block)

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
	**Input** CLAIM : Resource monopoly request signal RELEASE : Release signal **Output** BUSY : Acquisition disable signal of requested resource(Wait)

Function

This function block is used to acquire the exclusive control right of the system resource.

During executing SEMA function(CLAIM = 1 or 0, RELEASE = 0), BUSY will be 1 if other program is using the resource. To acquire the control right of resource, operate SEMA with CLAIM = 1 and RELEASE = 0 and wait till BUSY becomes 0. When BUSY becomes 0, control relating resource and execute SEMA once with CLAIM = 0 and RELEASE = 1 after control operation to transfer the control right.

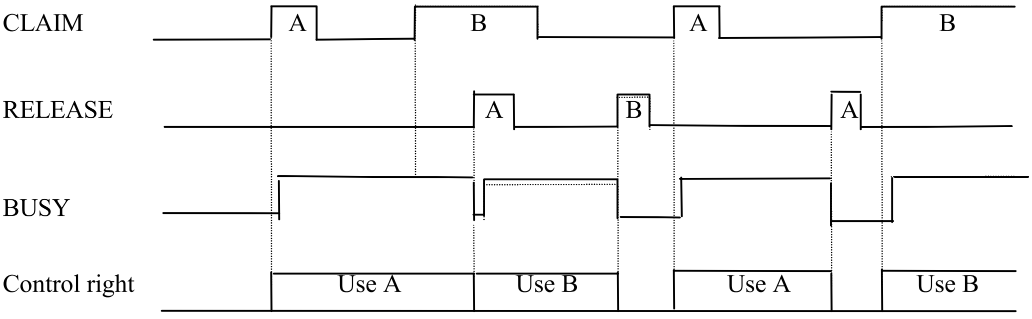
(The transfer of control right by SEMA with CLAIM = 0 and RELEASE = 1 shall be executed in the program that has the control right currently.)

- SEMA's instance shall be set in the global area so that access commonly in the program requiring the resource.
- Each program requiring same resource shall be set by same priority.
- Shall not be used between GM1 multi CPU modules.
- Internal execution structure of SEMA function block

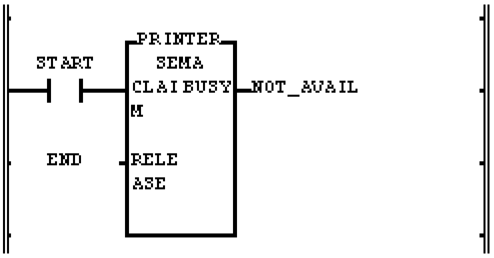
```
VAR   X : BOOL : = 0 ;   END_VAR
   BUSY : = X ;
   IF   CLAIM   THEN   X : = 1 ;
   ELSIF   RELEASE   THEN   BUSY : = 0; X : = 0 ;
   END_IF
```

Time chart

In case of exchanging access right for same resource in program block A and program block B.



■ Program example

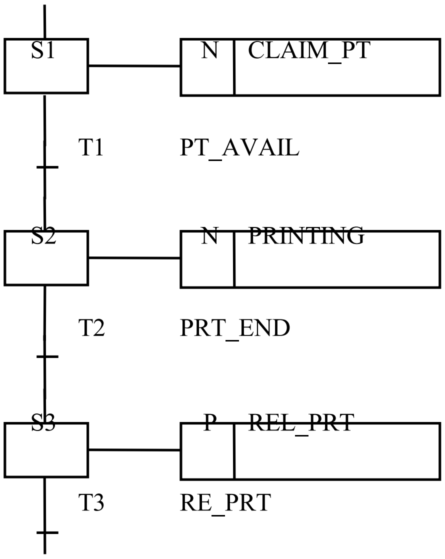
<table>
<tr><th>LD</th><th>IL</th></tr>
<tr><td>

PRINTER
START
SEMA
CLAIM
BUSY
NOT_AVAIL
END
RELEASE

</td><td>CAL SEMA PRINTER
CLAIM:= START
RELEASE:= END
LD PRINTER.BUSY
ST NOT_AVAIL</td></tr>
</table>

In order to output in different program block by the printer in PLC system, declare the instance 'PRINTER' as global and control the control right of printer by SEMA function block named as 'PRINTER' easily.
BUSY signal will be 1 and output 1 to NOT_AVAIL if other program block uses the printer at request of control right by 'PRINTER' SEMA when START is 1 and END is 0 at the time of printer output.
If other block does not use the print, BUSY becomes 0 and the program for output the signal to printer shall be operated.
After completing the print operation, execute 'PRINTER' SEMA with START = 0 and END = 1 so that other block can take the control right of printer.

```
VAR_EXTERNAL
PRINTER : SEMA
END_VAR
```



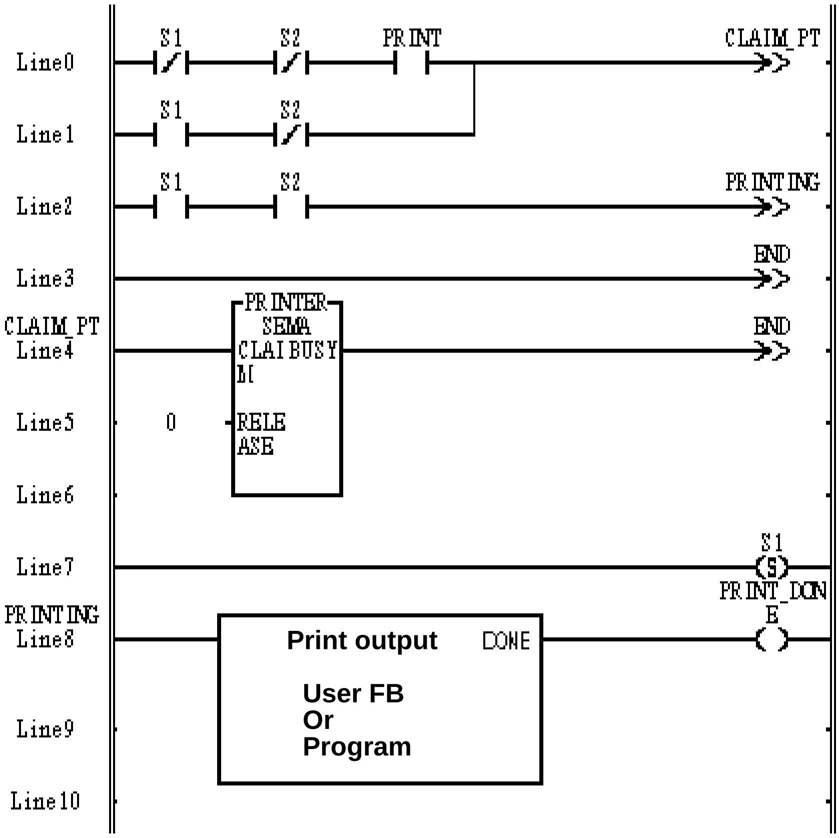
S1	CLAIM_PT; Printer control right request

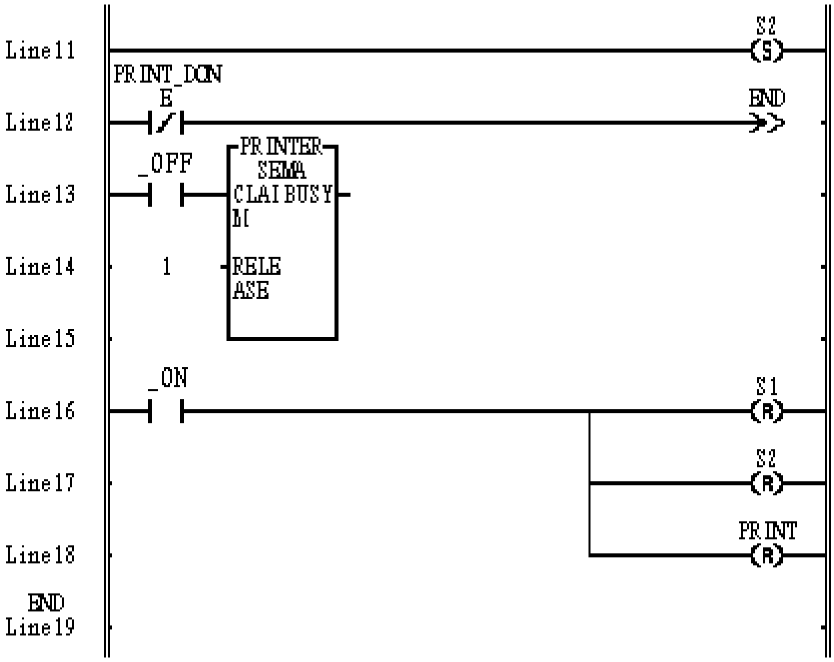
	CAL SEMA PRINTER
	CLAIM:= 1
	RELEASE:= 0

T1	PT_AVAIL; Printer control right acquisition check
	LDN PRINTER.BUSY ST TRANS
S2	PRINTING ; Print output
	Printer control program When completing the print, PRINT_DONE:= 1
T2	PRT_END ; Print completion check
	LD PRINT_DONE ST TRANS
S3	REL_PRT ; Printer control right transfer

	CAL SEMA PRINTER
	CLAIM:= 0
	RELEASE:= 1

T3	RE_PRT ; Printer request again
	LD PRT_REQ ST TRANS

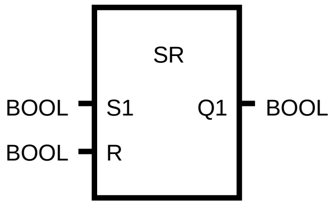

Line0
S1
S2
PRINT
CLAIM_PT
Line1
S1
S2
Line2
S1
S2
PRINTING
Line3
END
CLAIM_PT
Line4
PRINTER-SEMA
CLAIM
BUSY
END
Line5
0
RELEASE
Line6
Line7
S1
S
PRINTING
Line8
Print output
DONE
PRINT_DONE
User FB
Or
Program
Line9
Line10


Line11
S2
S
PRINT_DONE
Line12
END
_OFF
Line13
PRINTER-SEMA
CLAIM
BUSY
Line14
1
RELEASE
Line15
_ON
Line16
S1
R
Line17
S2
R
Line18
PRINT
R
END
Line19

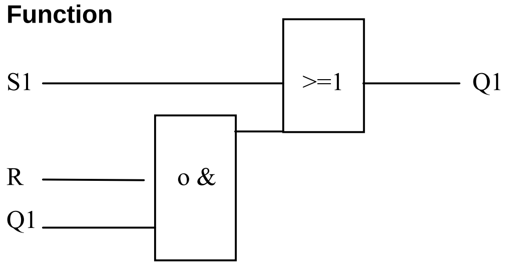
SR

Set Priority Bistable

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

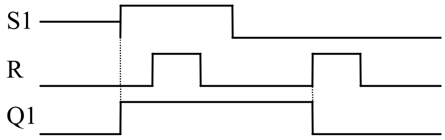
Function	Description
	**Input** S1 : Set condition R : Reset condition **Output** Q1 : Operation result

■ Function

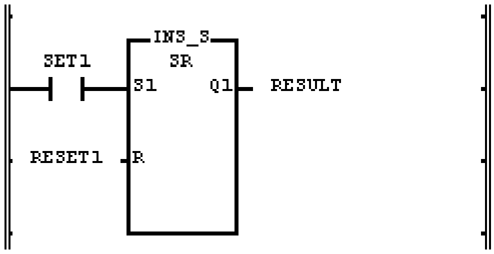


When S1 is 1, output Q1 becomes 1 always regardless of R.
Output Q1 holds previous status and becomes 0 when S1 is 0 and R is 1.
Initial status of Q1 is 0.

■ Time chart



■ Program example

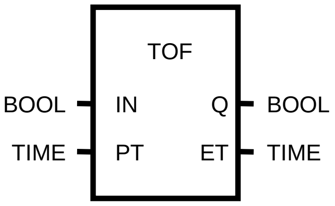
LD	IL
	CAL SR INS_S S1: = SET1 R: = RESET1 LD INS_S.Q1 ST RESULT

(1) When the input variable SET1 On, the output variable RESULT will be 1.
(2) If input variable SET1 is OFF and RESULT1 is On, output variable RESULT will be 0.

TOF

OFF delay timer(Function block)

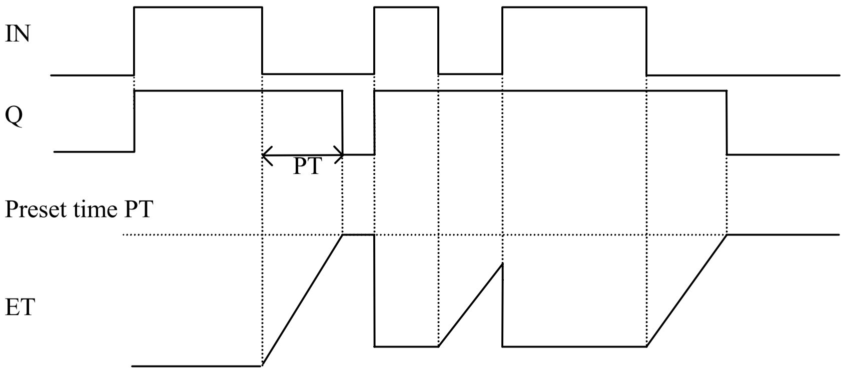
Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
	**Input** IN : Timer operation condition PT : Preset Time **Output** Q : Timer output ET : Elapsed Time

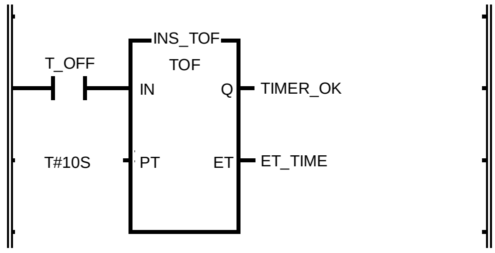
Function

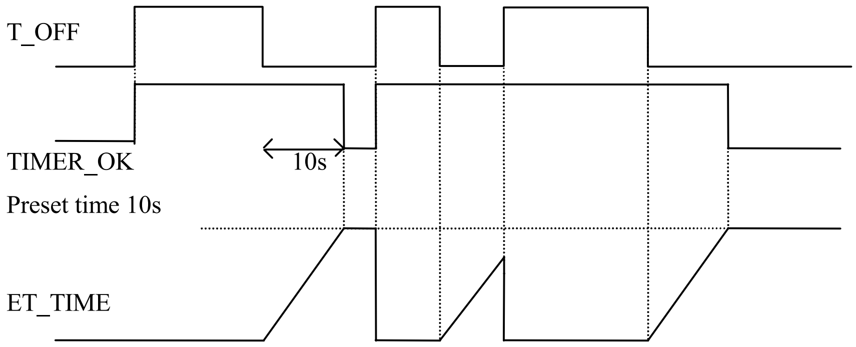
If IN is 1, Q becomes 1 and Q becomes 0 after elapsed time from IN=0 to preset time by PT.
Output the elapsed time to ET after IN becomes 0.
If IN becomes 1 before ET reaches the preset time, the elapsed time return 0.

Time chart



Program example

LD	IL
 INS_TOF TOF T_OFF IN Q TIMER_OK T#10S PT ET ET_TIME	CAL TOF INS_TOF IN:= T_OFF PT:= T#10S LD INS_TOF.Q ST TIMER_OK LD INS_TOF.ET ST ET_TIME

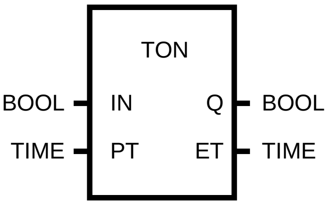


If input variable T_OFF is 1, output variable TIMER_OK outputs 1 and TIMER_OK becomes 0 in ten seconds after T_OFF. If T_OFF becomes 1 in 10 seconds after it is 0, the timer is initialized. The measuring time of timer is output to ET_TIME.

TON

ON delay timer(Function block)

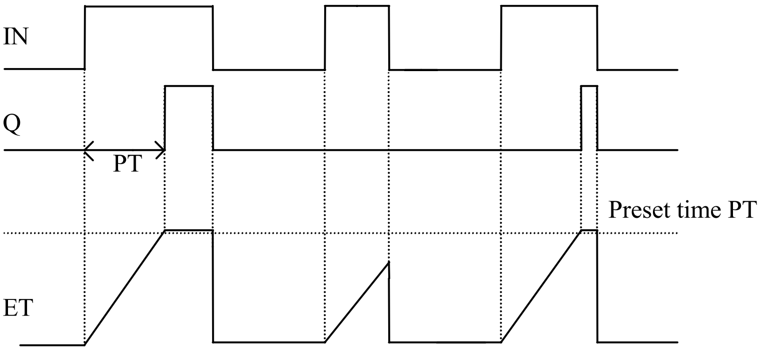
Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

Function	Description
	**Input** IN : Timer operation condition PT : Preset Time **Output** Q : Timer output ET : Elapsed Time

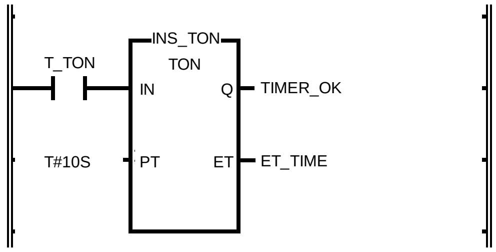
Function

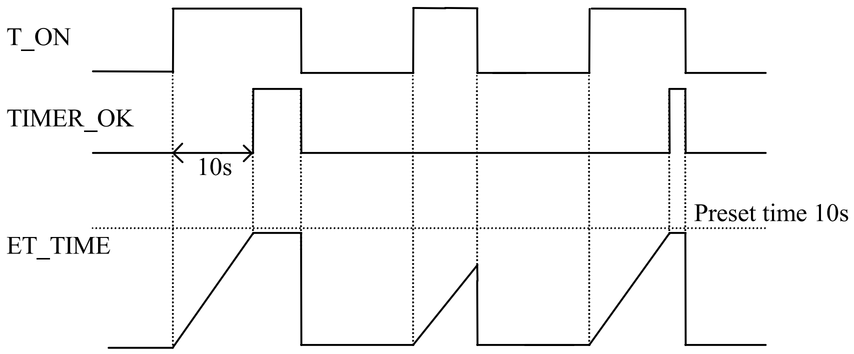
Output the elapsed time to ET after IN is 1.
If IN is 0 before ET reaches the preset time, the elapsed time becomes 0.
If IN is 0 after Q is 1, Q will be 0.

Time chart



Program example

LD	IL
 INS_TON TON T_TON IN Q TIMER_OK T#10S PT ET ET_TIME	CAL TON INS_TON IN: = T_ON PT: = T#10S LD INS_TON.Q ST TIMER_OK LD INS_TON.ET ST ET_TIME



(1) In 10 seconds after input variable T_ON is 1,　output variable TIMER_OK is 1.

(2) After input variable T_ON is 1, the elapsed time is output to output variable ET-TIME.

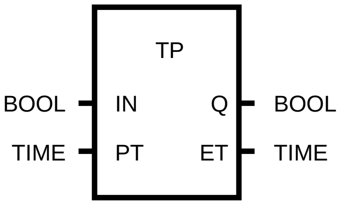
(3) If T_ON becomes 0 before ET_TIME reaches the preset time 10 seconds, the elapsed time ET_TIME will be 0.

(4) If T_ON becomes 0 after TIMER_OK is 1, TIMER_OK becomes 0 and the elapsed time ET_TIME becomes 0.

TP

Pulse timer(Function block)

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●	●	●

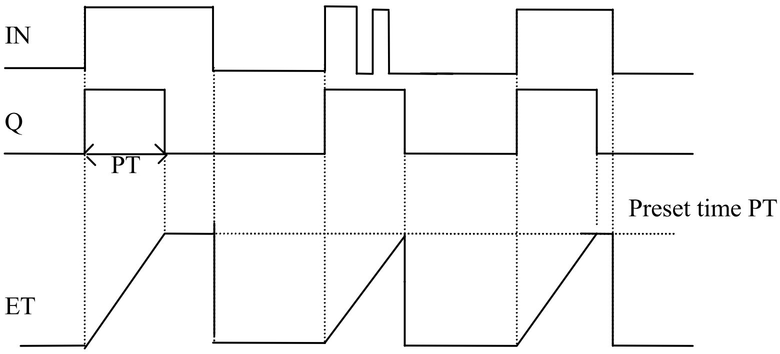
Function	Description
	**Input** IN : Timer operation condition PT : Preset Time **Output** Q : Timer output ET : Elapsed Time

■ Function

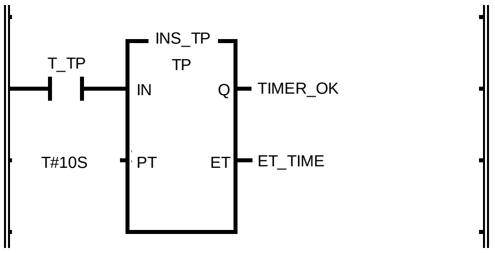
If IN is 1, Q becomes 1 during preset time and if ET reaches PT, Q becomes 0 automatically.

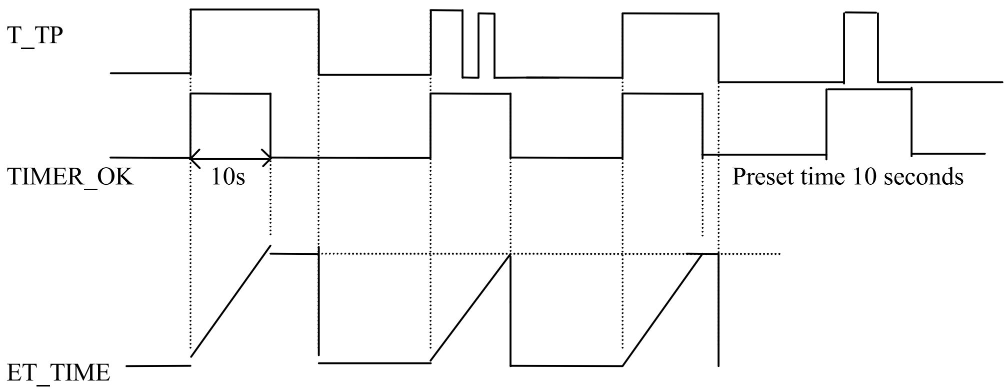
Elapsed time ET is increased when IN is 1 and holds the value when it reaches PT and becomes 0 when IN is 0. It does not matter whether IN is 0 or 1 during increasing ET.

■ Time chart



■ Program example

LD	IL
 INS_TP TP T_TP IN Q TIMER_OK T#10S PT ET ET_TIME	CAL TP INS_TP IN: = T_TP PT: = T#10S LD INS_TP.Q ST TIMER_OK LD INS_TP.ET ST ET_TIME



(1) After input variable T_TP changes from 0 to 1, TIMER_OK will be 1 during 10 seconds. After driving the timer, the change of T_TP signal is ignored for 10 seconds.

(2) ET_TIME value stops at T#10S after increasing. When T_TP is 0, ET_TIME becomes