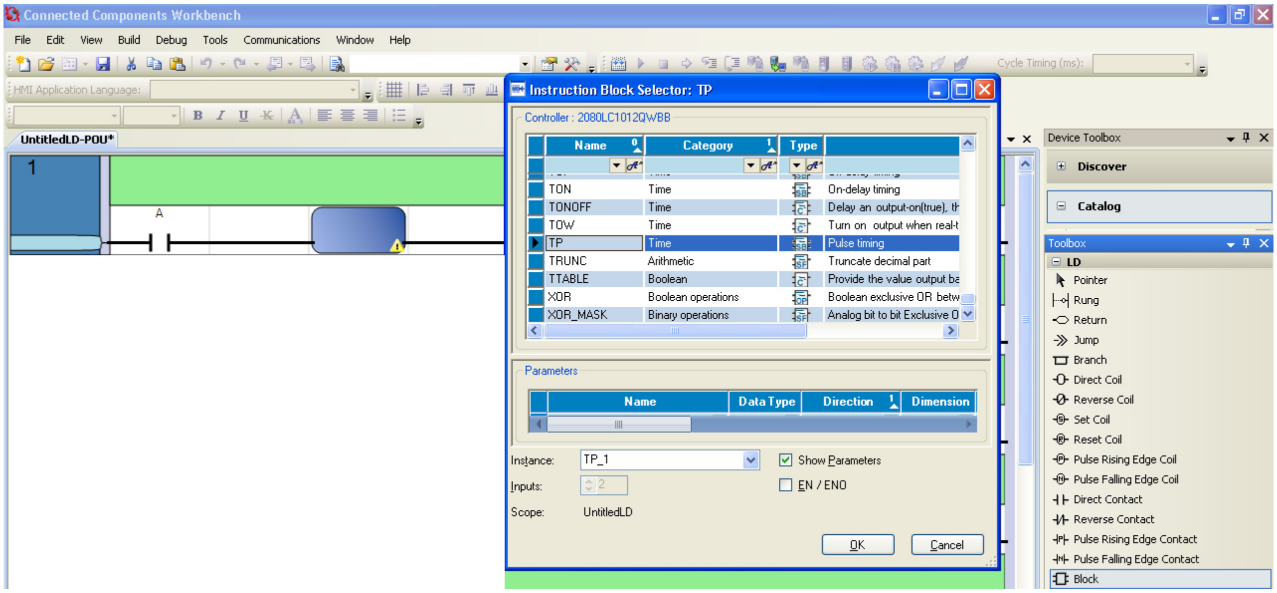
Functional Blocks - Timers

There are several ways to program using ladder logic. With ladder logic alone, you have AND, OR, and NOT. From ECE 275, that is all you need to do anything. A PLC is running on a processor, however. With a processor, it isn't that hard to build funcitons like counters, timers, and comparitors. These are the functions provided in the Functional Blocks.

To access a block, drag the block icon to the ladder diagram. It will then ask you which type of block. Select TP (timer pulse)



Instruction Block Selector: TP

Controller : 2080LC1012QWBB

Name	Category	Type	
TON	Time	SB	On-delay timing
TONOFF	Time	C	Delay an output-on(true), th
TOW	Time	C	Turn on output when real-t
TP	Time	SB	Pulse timing
TRUNC	Arithmetic	SF	Truncate decimal part
TTABLE	Boolean	C	Provide the value output ba
XOR	Boolean operations	OP	Boolean exclusive OR betw
XOR_MASK	Binary operations	SF	Analog bit to bit Exclusive O

Parameters

Name	Data Type	Direction	Dimension

Instance: TP_1

Inputs: 2

Scope: UntitledLD

☑ Show Parameters

☐ EN / ENO

In the lower left block, input the time with the format of

T#2h34m56s789ms

meaning

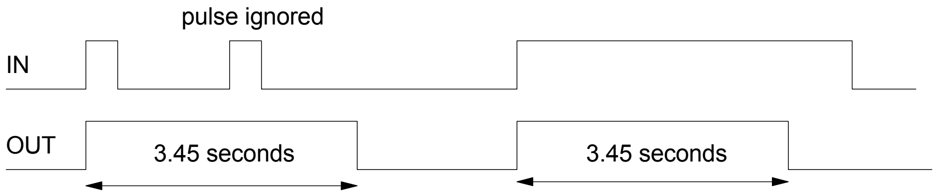
T#:	hard-code a time
h	hours
m	minutes
s	seconds
ms	milliseconds

For example,

T#2h34m56s789ms	means 2:34:56.789
T#3s45ms	means 3.45 seconds
T#4s	mens 4 seconds

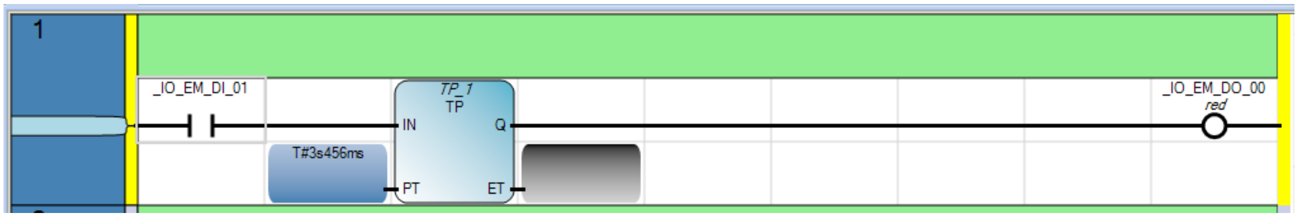
Timer Pulse (TP):

Timer Pulse turns on the output for N seconds every time it sees a rising edge on the input



Timing diagram for the Timer Pulse (TP) block

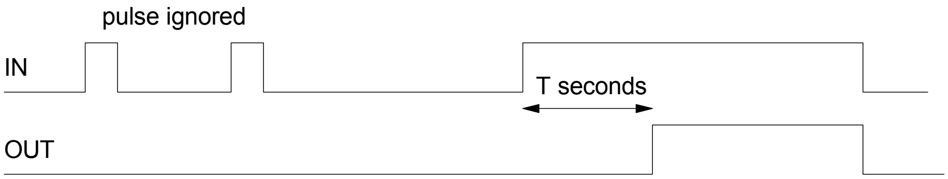
For example, the following program will turn on the red LED for 3.45 seconds whenever button 1 is pressed



Timer On: TON

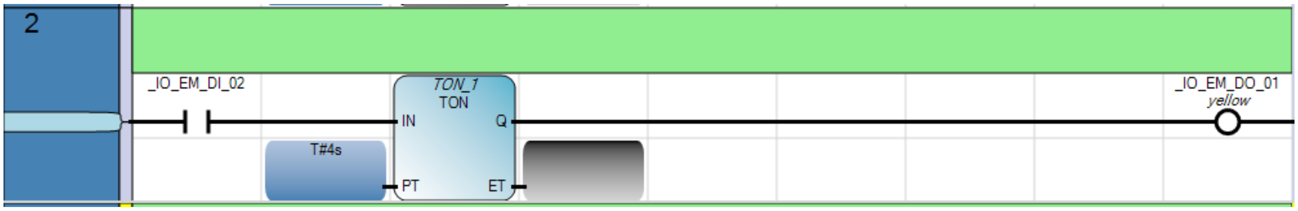
The On Timer will turn on the output once the input has been true for T seconds.

- If the input is true for less than T seconds, the output remains off.
- If the input is true for more than T seconds, the output turns on.
- Once the input goes low, the counter resets and the output turns off.



Timing diagram for the TON block

For example, the following program turns on the yellow light with button #3 is held for at least 4 seconds

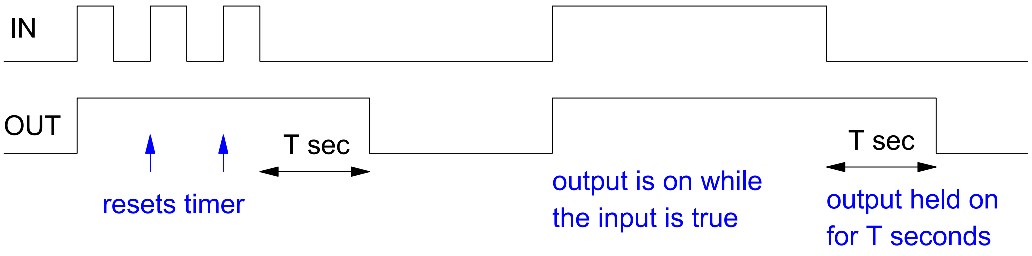


Timer Off Block: TOF

The timer off block

- Turns on the output as soon as the input goes high
- The output remains on for T seconds after the input is released

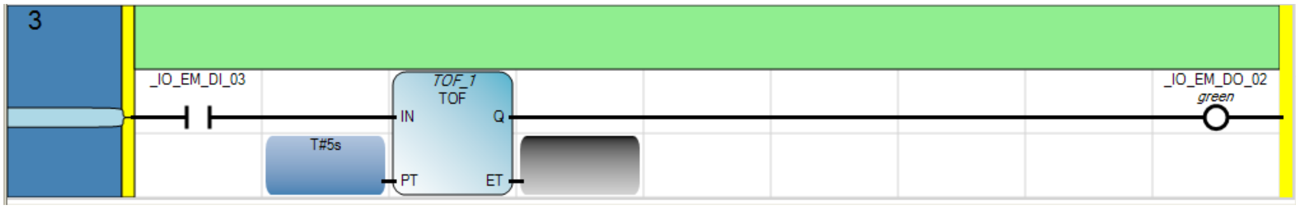
Note that the input has to remain off for T seconds for the output to clear. Each time you pulse the input, the timer resets.


IN
OUT
resets timer
T sec
output is on while the input is true
T sec
output held on for T seconds

Timing diagram for TOFF block

For example, the following program

- Turns on the green LED whenever button #3 is pressed
- The green LED remains on for 5 seconds when button #3 is released



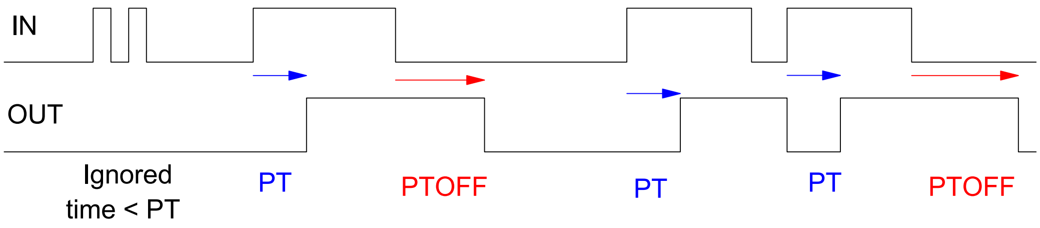
On-Off Timer: TONOFF

The On-Off Timer

- Turns on the output if the input is held on for at least PT seconds

- Keeps the output on for PTOFF seconds when the input goes low.

Note that if the input goes high again, the output clears and the counter resets.

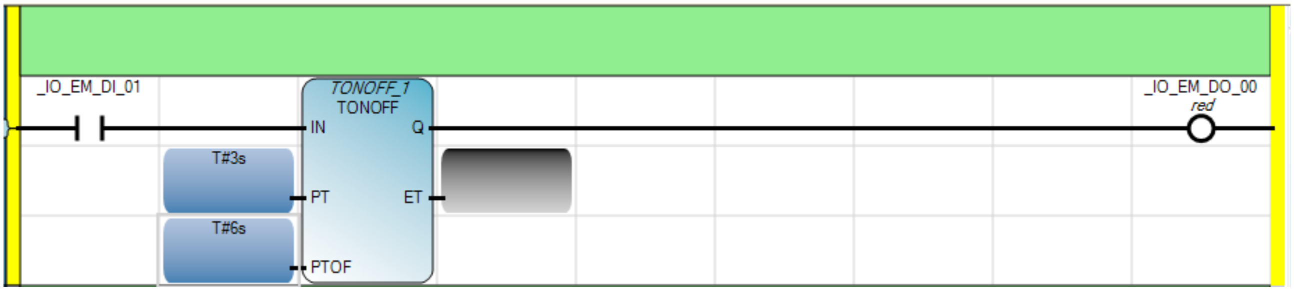


Timing Diagram for TONOFF Block

For example,

- Do nothing if button one is held for less than 3 seconds.
- Turn on the red LED when button 1 is held down for 3 seconds.
- Keep the red LED on for 6 seconds when button one is released.

A program to do this is:



Timer Example: Bathroom Light and Fan

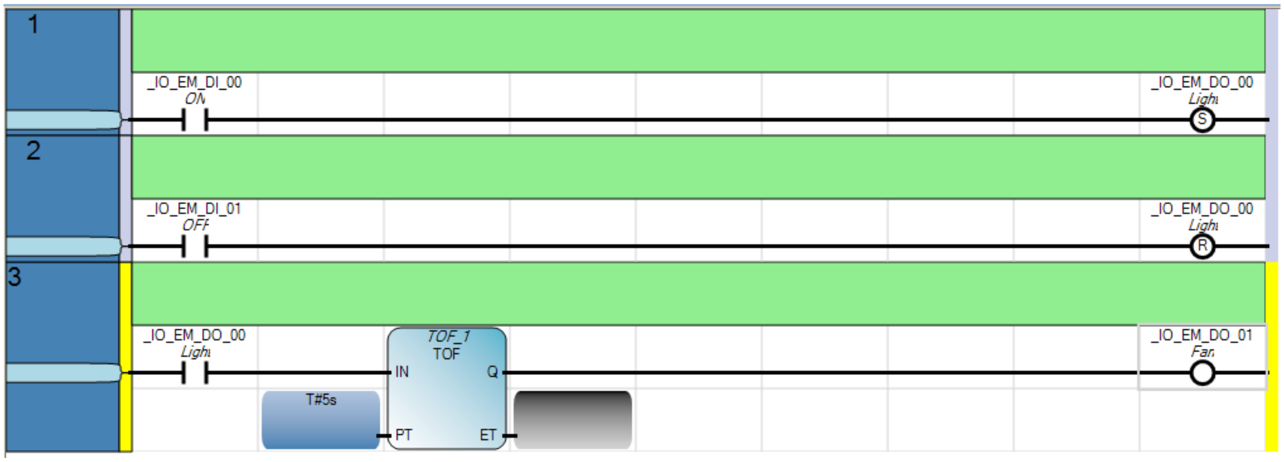
Write a program to control the lights and fan for a bathroom.

- When the light switch is turned on, the light and fan are both turned on.
- Whenthe light switch is turned off, the light turns off immediately and the fan remains on for 5 seconds.

Hardware Assignments: Let

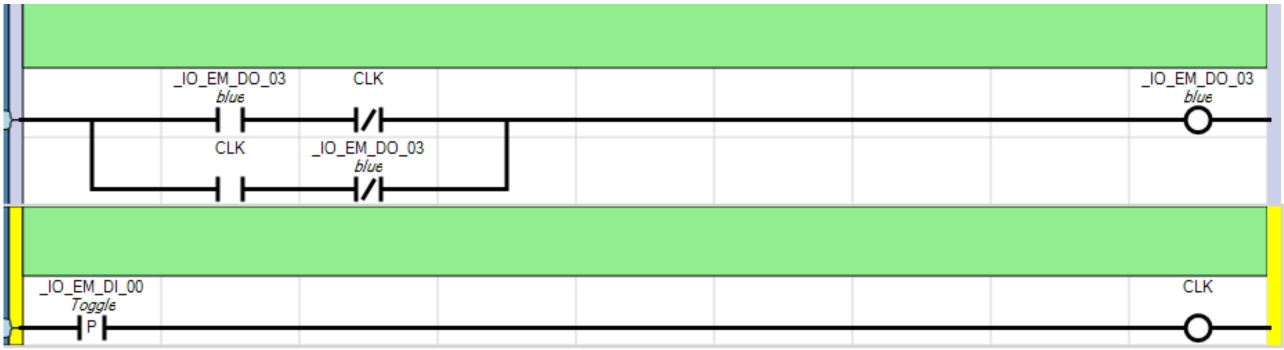
- IN0 be the on switch
- IN1 be the off switch
- OUT0 be the bathroom light
- OUT1 be the bathroom fan

Ladder Logic Program:

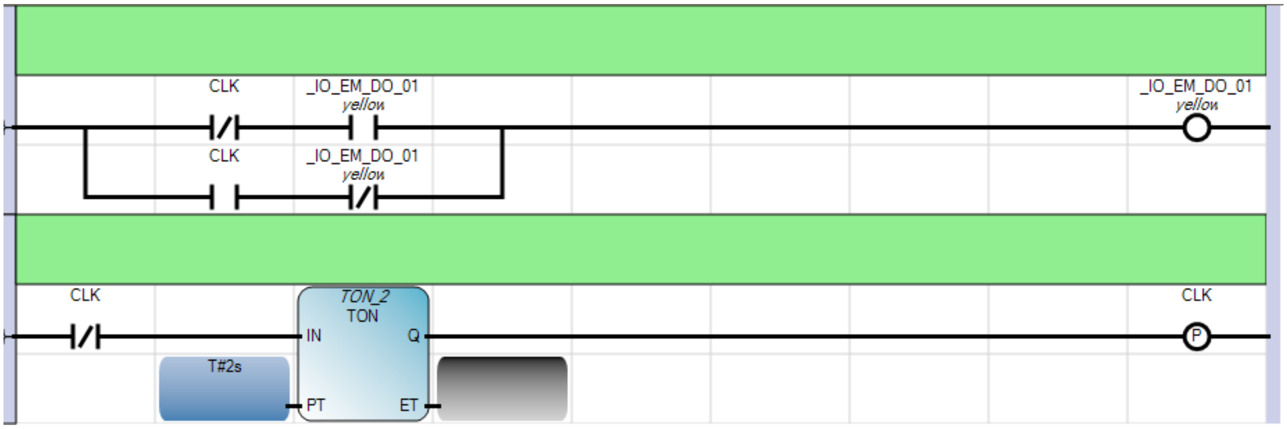


Example 2: Toggle an output every 1.00 second (flashing blue).

Step 1: Write a program to toggle an output



Step 2: Replace the push button with a timer

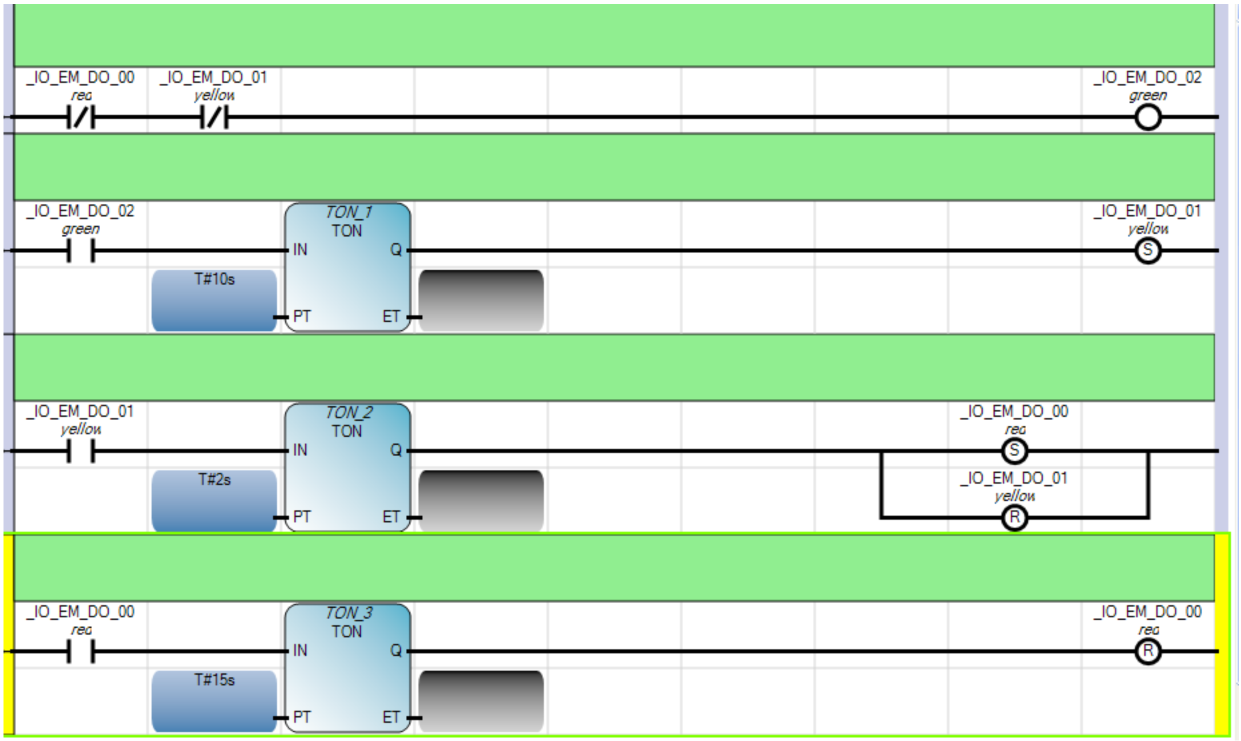


Example 3: Automated Stop-Light.

Repeat the stop-light from before using CLK to replace the push button.

Example 4: Automated Stoplight (take 2)

State-transistion logic is a bear to follow. You can also just use three timers: a separate one for green, yellow, and red:


_IO_EM_DO_00
red
_IO_EM_DO_01
yellow
_IO_EM_DO_02
green
_IO_EM_DO_02
green
TON_1
TON
IN
Q
PT
ET
T#10s
_IO_EM_DO_01
yellow
S
_IO_EM_DO_01
yellow
TON_2
TON
IN
Q
PT
ET
T#2s
_IO_EM_DO_00
red
S
_IO_EM_DO_01
yellow
R
_IO_EM_DO_00
red
TON_3
TON
IN
Q
PT
ET
T#15s
_IO_EM_DO_00
red
R