

# PLC Exercises 1

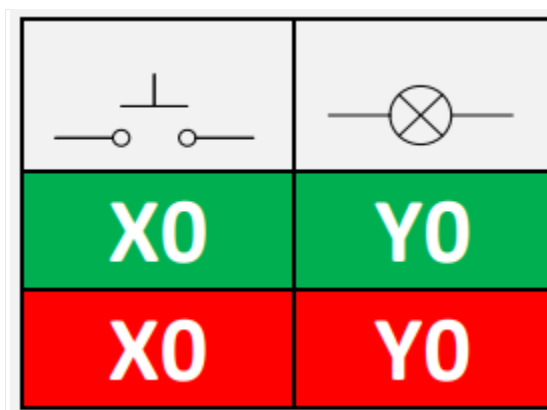
Practice your PLC Programming skills using these exercises

If you receive error while executing your logic, it could be because you cannot normally actuate inputs (X0, X1, X2 and so on..) in the simulator so in this case, kindly practice your exercise using memory bits (M0, M1, M2 and so on..) instead of inputs (X0, X1, X2 and so on..)

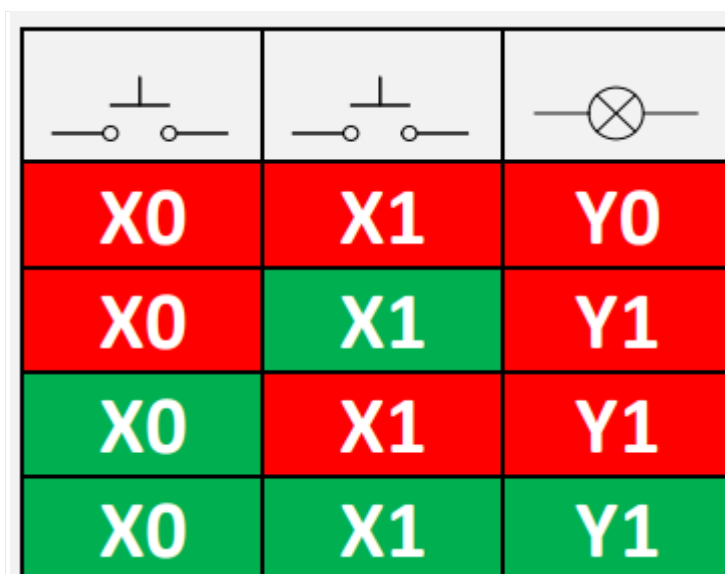
If you are using AHCPU, you should use input bits as X0.0, X0.1 instead of X0, X1 and outputs as Y0.0, Y0.1 instead of Y0, Y1 and so on.

## Section 1 - Series & Parallel Circuits (LD, LDI, OUT)

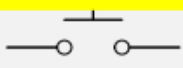

1. Code a logic when X0 is ON, Y0 should be ON and when X0 is OFF, Y0 should be OFF.



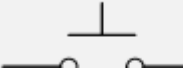


2. Code a series logic such that when X0 & X1 is ON, Y0 is ON



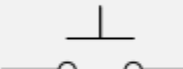


3. Code a Parallel Input logic such that when either of X0 or X1 is ON, Y0 is ON

		
<b>X0</b>	<b>X1</b>	<b>Y0</b>
<b>X0</b>	<b>X1</b>	<b>Y1</b>
<b>X0</b>	<b>X1</b>	<b>Y1</b>
<b>X0</b>	<b>X1</b>	<b>Y1</b>

4. Code a Parallel Output logic such that when X0 is ON, Y0 & Y1 are ON

		
<b>X0</b>	<b>Y0</b>	<b>Y1</b>
<b>X0</b>	<b>Y0</b>	<b>Y1</b>

5. Code a logic when X0 is ON, Y0 is ON & Y1 is OFF & when X0 is OFF, Y1 should be ON & Y0 should be OFF

		
<b>X0</b>	<b>Y0</b>	<b>Y1</b>
<b>X0</b>	<b>Y0</b>	<b>Y1</b>

## **Section 2 - Latching & Interlocking Circuits – Using OUT Commands**

1. Code a logic when X0 is pressed, Y0 should be latched & when X1 is Pressed, Y0 should be unlatched. (In this case consider X0 and X1 as NO Push Button)
2. Code a logic to interlock Y0 & Y1 using X0 & X1. Consider X0 & X1 as NO Push Buttons & use X2 as NC Push Button to reset Y0 & Y1.

## **Section 3 - Latching & Interlocking Circuits – Using SET/RESET Commands**

1. Code a logic when X0 is pressed, Y0 should be latched & when X1 is Pressed, Y0 should be unlatched. (In this case consider X0 and X1 as NO Push Button)
2. Code a logic to interlock Y0 & Y1 using X0 & X1. Consider X0 & X1 as NO Push Buttons & use X2 as NC Push Button to reset Y0 & Y1.

## **Section 4 – Use to Timers (TMR)**

1. Code a Logic such that when X0 is ON, Y0 gets ON after delay of 5 secs. When X0 is OFF, Y0 should be OFF
2. Do the similar operation, but delay should change to 0.05 seconds (or 50ms)
3. Do the similar operation, but delay should change to 0.005 seconds (or 5ms)
4. Code a Logic when X0 is ON, Y0 gets ON after 3 seconds delay & then after delay of 3 more seconds Y1 should be on & after delay of 3 seconds Y2 should be ON.
5. Code a Logic, when X0 is ON, Y0 is ON for 10 seconds then OFF.
6. Code a Logic that can be used to start a Motor 1 (Y0) using X0 (NO) and then after a delay of 10 sec start Motor 2 (Y1). When the Motor 1 is switched off using X1(NC) there should be delay of 10 sec before the Motor 2 is off.
7. Code a Logic in which Y0 is ON (latch) only when X1 (NO) is ON in 10 sec. just after start of the Switch X0 (NO). Otherwise if X1 is pressed after 10 sec. Nothing should happen. Unlatch Y0 when X0 is OFF.

## Section 5- Use of Timers (TMR) with Comparison Commands

1. Use one push button (X0) to turn ON Y0 in following sequence: Y0 on for 2 sec. then off for 2 sec. then on for 3 sec. then continuously off.
2. Use X0 to blink Y0. (On Time 0.5 second & OFF time 0.5 second).
3. Use X0 to blink Y0. (On Time 0.5 second & OFF time 1 second).
4. When you press X0, Y0 and Y1 should blink with the delay of 1 sec continuously (such that 1 sec. ON & 1 sec. OFF) and when you press X1, Y0 and Y1 should blink with the delay of 1.5 sec. (such that 1.5 sec. ON & 1.5 sec. OFF)
5. Code a basic Traffic Light Simulator such that when you press X0 following sequence should occur

Time (in secs)	Y0 (red)	Y1 (Orange)	Y2 (Green)
0 ~ 10			
10 ~ 12			
12 ~ 23			
23 ~ 25			
>=25 Timer reset and cycle repeats			

## Section 6- Use of Counters (CNT) with Comparators Commands

1. Code a logic such that When X0 is pressed 3 times, Y0 should be ON.
2. Code a logic such that When X0 is pressed 3 times, Y0 should be ON & when X1 is pressed Y0 should be OFF (Use Counter reset command)
3. Code a logic in which Sensor (X1) is being used to count the bottles on the conveyor (Y0) which is being latched by X0. Use X0 to latch the conveyor & after X1 counts 10, conveyor should be unlatched.
4. Code a logic in which
  - When X1 is pressed once– Y0 gets ON, Y1, Y2 get off,
  - When X1 is pressed twice – Y1 gets ON and Y0, Y2 get OFF
  - When X1 is pressed thrice – Y2 gets ON and Y0, Y1 get OFF
  - When X1 is pressed fourth time, all outputs should be OFF & Cycle should repeat on pressing X1 again

## Section 7- Use of Data Register (D0) & related commands MOV, RST, ADD, SUB, MUL, DIV & INC/DEC

1. Switch ON a Lamp (Y0) after 5 sec. Make a provision using “MOV” command to change the delay time to 10 second by pressing X1 & then back to 5 sec by pressing X2. You have 1 maintained button (X0) and 2 push buttons only. Maintained button to switch on/off the lamp. Other two push button for changing the time from 5 to 10 sec and viceversa
2. Code a logic to Change the time of above timer using Increment & (X3- PB) Decrement (X4- PB) commands.
3. Code a logic to blink the output Y0 using X0 with following delay time sequence using MOV command a. Default Delay Time = ON time 1 Second & OFF Time 1 second b. Case A, When X1 is pressed = ON time 2 seconds & OFF time 1 second c. Case B, When X2 is pressed = ON time 1.5 seconds & OFF Time 2 seconds
4. Code a logic as above but change the time using Increment (X1) & Decrement (X2) functions such that minimum blinking time should be 0.5 seconds (ON/OFF) & maximum should be 3 seconds (ON/OFF). The step of Increment & Decrement should be 0.1 second
5. I have a constant 10 in D0
  - Multiply it with 100
  - Add 56 to it
  - Subtract 14 from it
  - Finally store it in D10