

## PLC Exercises - Part 1

Practice your PLC Programming for Ladder Diagram, FBD and Structured Text Programming

The solutions in Ladder Diagram, FBD and ST are available in the zip file named “1-Wiki Solutions v01.06.2023” shared in the resources.

Solve the following exercises in the following programming languages:

- Ladder Diagram
- FBD (Functional Block Diagram)
- ST (Structured Text)

In the following exercises, consider declaring the variables as follows:

Variable type	Data type	Variable declaration	Remarks
Input, Output	Bool	xStart, xStop, xReset, xInput, xSensor, xInductiveSensor, xMotor, xSolenoid etc.	The first alphabet defines the datatype followed by variable name
Value	Int or Int16	iCount, iLevel, iSpeed etc.	
Value	DInt or Int32	diCount, diLevel, diOrderID etc.	
Value	Real	rSpeed, rLevel, rHeight etc.	
Name	String	sID, sRecipe, sPassword etc.	
Parameter	Time	tDelay, tTimeout etc.	

### Section 1 - Series & Parallel logics

1. Write a program when xStart is TRUE, xMotor should be TRUE, and when xStart is False, xMotor should be FALSE.
2. Write a program to fulfill the following boolean table:

xStartA :BOOL (Input)	xStartB :BOOL (Input)	xOutputX : BOOL (Output)
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

3. Write a program to fulfill the following boolean table:

<b>xStartA :BOOL (Input)</b>	<b>xStartB :BOOL (Input)</b>	<b>xOutputX : BOOL (Output)</b>
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

4. Write a program to fulfill the following boolean table:

<b>xStartA :BOOL (Input)</b>	<b>xOutputX : BOOL (Output)</b>
FALSE	TRUE
TRUE	FALSE

5. Write a program to fulfill the following boolean table:

<b>xStartA :BOOL (Input)</b>	<b>xOutputX :BOOL (OUTPUT)</b>	<b>xOutputY : BOOL (Output)</b>
FALSE	TRUE	TRUE
TRUE	FALSE	FALSE

6. Write a program to fulfill the following boolean table:

<b>xStartA :BOOL (Input)</b>	<b>xOutputX :BOOL (OUTPUT)</b>	<b>xOutputY : BOOL (Output)</b>
FALSE	TRUE	FALSE
TRUE	FALSE	TRUE

7. Write a program to fulfill the following boolean table:

<b>xStartA :BOOL (Input)</b>	<b>xStartB :BOOL (Input)</b>	<b>xOutputY : BOOL (Output)</b>
FALSE	FALSE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
TRUE	TRUE	FALSE

## Section 2 - Latching and Interlocking logics using contacts

- Write a program when xStart (Input) is pressed, xMotor (output) should be latched & when xStop is Pressed, xMotor (output) should be unlatched. In this case, consider xStart and xStop as NO Push Button.
- Write a program when xStart (Input) is pressed, xMotor (output) should be latched & when xStop is Pressed, xMotor (output) should be unlatched. In this case, consider xStart and xStop as NC Push Button.
- Write a program to interlock xSolenoidA (Output) and xSolenoidB (output). - When xStartA (Input) is pressed, xSolenoidA (output) should be latched, and xSolenoidB should be unlatched - When xStartB (Input) is pressed, xSolenoidB (output) should be latched, and xSolenoidA should be unlatched - When xStop (Input) is pressed or when xStart A and xStartB both are pressed , both solenoids should be unlatched Consider xStartA, xStartB as NO push button, and xStop as NC Push Button.

### Section 3 - RS and SR flip-flops

1. Using RS flipflop write a program when xStart (Input) is pressed, xMotor (output) should be latched & when xStop is Pressed, xMotor (output) should be unlatched. In this case, consider xStart as NO push button and xStop as NC Push Button.
2. Using SR flipflop write a program when xStart (Input) is pressed, xMotor (output) should be latched & when xStop is Pressed, xMotor (output) should be unlatched. In this case, consider xStart as NO push button and xStop as NC Push Button.
3. Realize the difference in operations when both the inputs xStart and xStop are TRUE. Which type of flipflop should be used ideally in this situation?

### Section 4- Use of Timers

**TON: On-delay timer; TOF: Off-delay timer; TONOFF: On Off delay Timer**

1. Write a program such that when xStart (Input) is TRUE, xMotor (Output) gets TRUE after a delay of 5 secs. When xStart is FALSE, xMotor should be FALSE.
2. Write a program when xStart (Input) is TRUE, xMotorA (Output) gets TRUE after 3 seconds delay, and then after a delay of 3 more seconds, xMotorB (Output) should get TRUE & after a delay of 3 more seconds, xMotorC (Output) should get TRUE.
3. Write a program if xStart (Input) is TRUE xMotor (Output) is TRUE for 10 seconds, then gets FALSE. Consider xStart as Push button NO
4. Write a program such that: - xStart (Input) latches xMotorA (Output) - After a delay of 10 seconds, xMotorB (Output) latches - xStop (Input) unlatches xMotorA - After a delay of 10 seconds, xMotorB should get unlatched. Consider xStart as Push button NO and xStop as Push button NC
5. Write a program in which xMotor (Output) latches only when xStartB is TRUE within 10 seconds just after the xStartA gets TRUE. Otherwise, if xStartB is TRUE after 10 seconds, xMotor should not be latched. xMotor should unlatch using xStop Consider xStartA as Toggle Sw and xStartB as Push button NO and xStop as Push button NC

### Section 5- Use of Timers with Comparators

**EQ: Equal**

**NE: Not equal**

**LT: Less than**

**LE: Less than or equals**

**GT: Greater than**

**GE: Greater or equals**

1. Use xStart (NO Push button) to actuate xMotor in the following sequence: xMotor is TRUE for 2 seconds, then FALSE for 2 seconds, then TRUE for 3 seconds, then FALSE. The cycle should repeat when xStop (NC Push button) is pressed.
2. Use xStartA (Input) to blink xLamp (Output) such that the On-time is 2 seconds and Off-time is 3 seconds.
3. Write a program such that: - When xStartA (Input) is TRUE and xStartB (Input) is FALSE, xLampA (Output) and xLampB should blink continuously with a delay of 1 second. - When xStartB (Input) is TRUE and xStartA (Input) is FALSE, xLampA (Output)

and xLampB should blink continuously with a delay of 2 seconds. - In other situations, the lamps should not be ON

- Write a simple traffic light program to fulfill the following conditions. The cycle should start with xStart. The Timer should reset, and the cycle should repeat when the time is above 25 seconds. The cycle should stop and all outputs should be FALSE when xStart is FALSE .

Time (in seconds)	xRed (Output)	xOrange (Output)	xGreen (Output)
0 ~ 10	TRUE	FALSE	FALSE
11 ~ 12	TRUE	TRUE	FALSE
12 ~ 23	FALSE	FALSE	TRUE
24 ~ 25	FALSE	TRUE	TRUE

## Section 6- Use of Counters with Comparators

**CTU: Up Counter**

**CTD: Down Counter**

**CTUD: Up-Down Counter**

- Write a program such that when xStart is pressed three times, xOutput should get TRUE. And when xReset is TRUE once, the xOutput should be FALSE, and the counter should RESET.
- Write a program to fulfill the following conditions: - When xStart is pressed once - xOutputA should be TRUE, xOutputB and xOutputC should be FALSE - When xStart is pressed again -xOutputB should be TRUE, xOutputA and xOutputC should be FALSE - When xStart is pressed again -xOutputC should be TRUE, xOutputA and xOutputB should be FALSE - When xStart is pressed again - all the outputs should be FALSE, and the cycle should repeat itself

## Section 7- Use of variables with Math Operators

The following exercises are only possible with **Ladder diagram or Structured Text** as some commands with FBD does not support EN/ENO which are required for some operations.

**+: Addition**

**-: Subtraction**

**\*: Multiplication**

**/: Division**

**MOV: Move**

- Write a program to fulfill the following conditions: - xStart (NO Push Button) should latch the xLamp (output) after certain delay. - xStop (NC Push Button) should unlatch the bLamp - Use xTimeA (Input) to set the delay time to 10 seconds. - Use xTimeB (Input) to set the delay time to 5 seconds.
- Write the same program as the above but instead of moving the constant time of 5 and 10 seconds. Use xInc (Input) to increment the time by 100ms and xDec (Input) to decrement the time by 100ms.
- Add a program to the solution of Exercise 7.2. Set the default preset time of Timer to be 2 seconds and limit the minimum and maximum time to 1 second and 3 seconds, respectively.

The **solutions in Ladder Diagram, FBD and ST** are available in the zip file named **“1-Wiki Solutions v01.06.2023”** shared in the resources.