MIT/ICE/PLC-MNL/18

**Automation Lab**

**Laboratory Manual – ICE 4112**

## VI Semester B. Tech

**Name of the Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Registration Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**January – 2023**

MIT/ICE/PLC-MNL/18

CERTIFICATE

# This is to certify that the Laboratory Manual/Journal for the lab titled AUTOMATION Lab (ICE 4112) submitted by Mr./Ms.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Reg. No :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) of seventh semester, Instrumentation and Control Engineering for the academic year, as per laboratory course requirements, which has been evaluated and duly certified.

Place:

Date: Lab In-Charge

DEPARTMENT OF INSTRUMENTATION & CONTROL ENGINEERING

**January – 2023**

List of Experiments

|  |  |  |
| --- | --- | --- |
| **Expt. No.** | TITLE OF THE EXPERIMENT | **Page No.** |
| 1a | Introduction to LTPCM-DN30S1PLC and XL 5000 software | 4 |
| 1b | Implementation of combinational circuits | 16 |
| 2 | Timer functions and problems | 20 |
| 3 | Counter functions and problems | 24 |
| 4 | Compare and arithmetic instructions | 30 |
| 6 | Control of glass inspection station | 35 |
| 7 | Control of bottle filling machine | 36 |
| 8 | Control of traffic signal | 37 |
| 9 | Controlof Cleaning System | 38 |
| 10 | Tank filling device simulation | 39 |

**Evaluation Plan:**

Continuous Evaluation: 60% (Preparation, Lab performance, Journal, Assignments and

Regularity, Mini Project)

End Semester Lab Test: 40%

EXPERIMENT NO. 1a DATE:

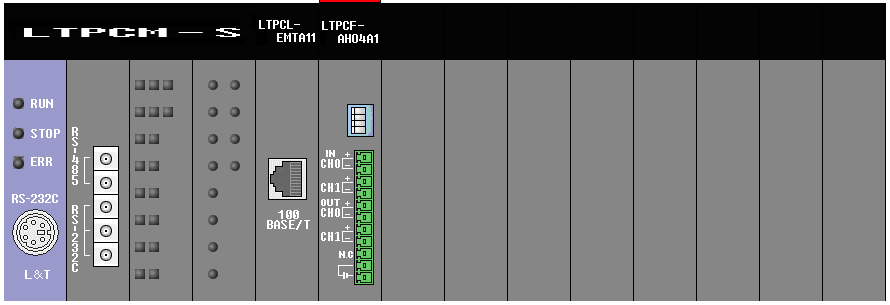
**Introduction**

1. **LTPCM-DN30S1 PLC** **:**

The **LTPCM-DN30S1** compact controller is the modular, space-saving controller for small automation systems that require either simple or advanced functionality for logic, HMI and networking. The compact design, low cost, and powerful features make the **LTPCM-DN30S1** a perfect solution for controlling small applications.

The **LTPCM-DN30S1** controller provides the flexibility and power to control a wide variety of devices in support of your automation needs. The compact design, flexible configuration, and powerful instruction set combine to make **LTPCM-DN30S1** a perfect solution for controlling a wide variety of applications.

**LTPCM-DN30S1 XLB** series PLC support 2 PTO(Pulse and Direction) of 100KZ and 8 High speed counters of same 100KZ pulse frequency it supports Linear and ring counter modes preset and comparator option. These PTO can be used to control stepper motors and servo motors.



High speed Ethernet

Analog I/O module

CPU

**Fig 1 Components LTPCM-DN30S1 XLB** series PLC

* The CPU combines a microprocessor, an integrated power supply, input and output circuits, high-speed motion control I/O, and on-board digital inputs/outputs in a compact housing to create a powerful controller.
* After you download your program, the CPU contains the logic required to monitor and control the devices in your application.
* The CPU monitors the inputs and changes the outputs according to the logic of your user program, which can include Boolean logic, counting, timing, complex math operations, and communications with other intelligent devices.

**2. Operating modes of the CPU**

The CPU has two modes of operation: STOP mode and RUN mode.

Status LEDs on the front of the CPU indicate the current mode of operation.

● In STOP mode, the CPU is not executing the program, and you can download a project.

● In RUN mode, the scan cycle is executed repeatedly. Interrupt events can occur and be

processed at any point within the program cycle phase.

Note

You cannot download a project while the CPU is in RUN mode. You can download your

project only when the CPU is in STOP mode.

**3. XL5000 software**

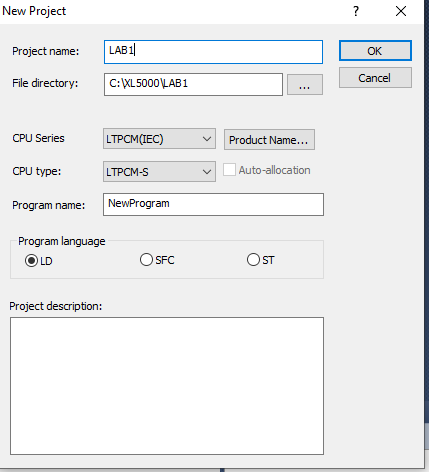
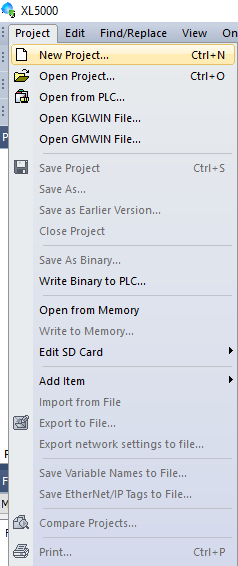
This section will help get you familiar with the XL5000 design environment. As our goal to help simplify your engineering efforts, Start the XL5000 software.

Double-click the XL5000 shortcut icon on your desktop

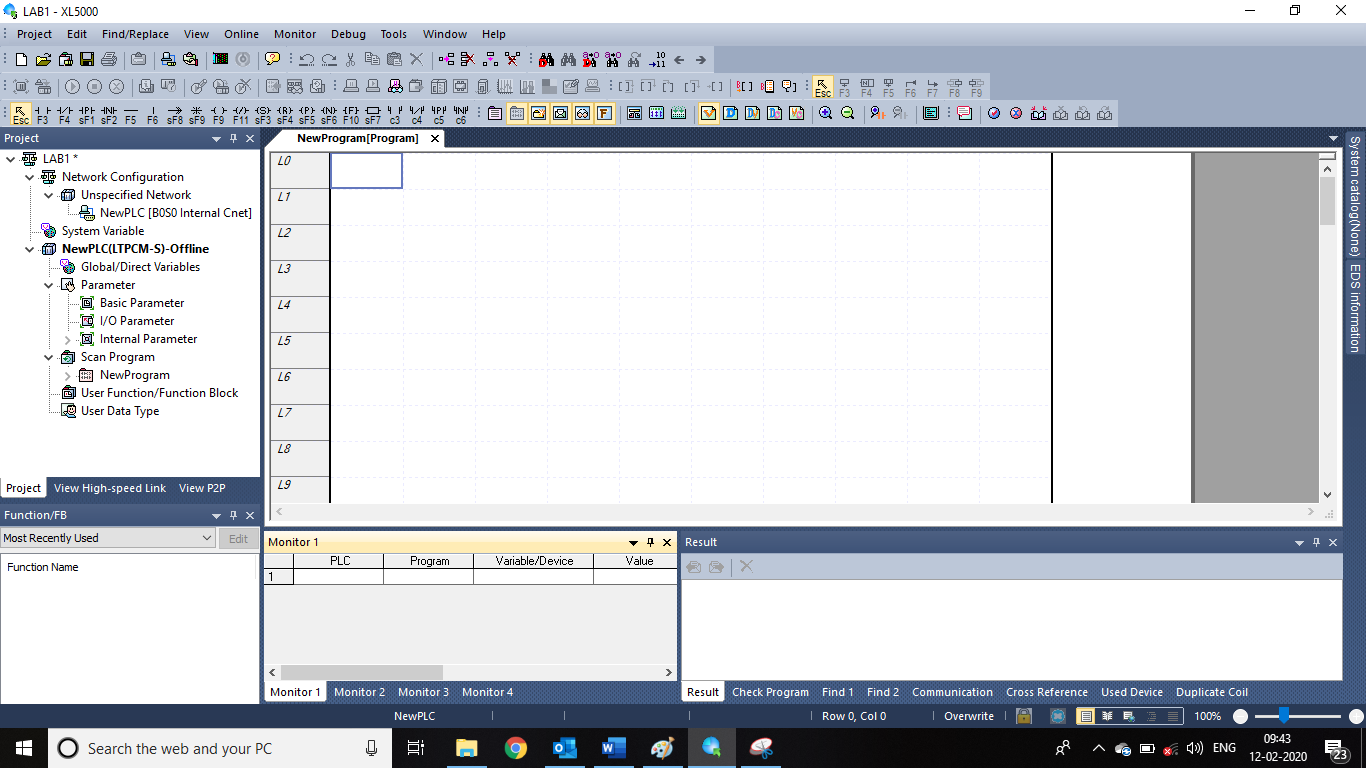


**3.1 Creating new project:**

Go to project tab select new project enter the project name and select the save directory.Here the CPU series is fixed which is LTPCM(main module),CPU type is S (standard) type,you can also give the program name also later can be renamed in properties



* Layout of XL5000 software working area.



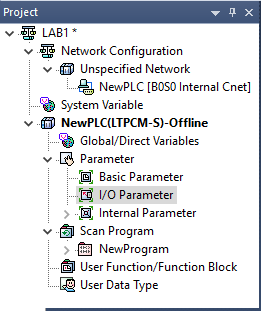
Variable monitoring and result windows

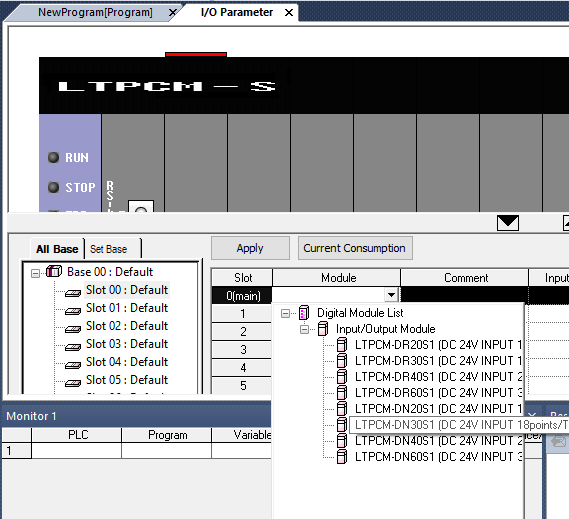
Ladder Editing area

Project Tab

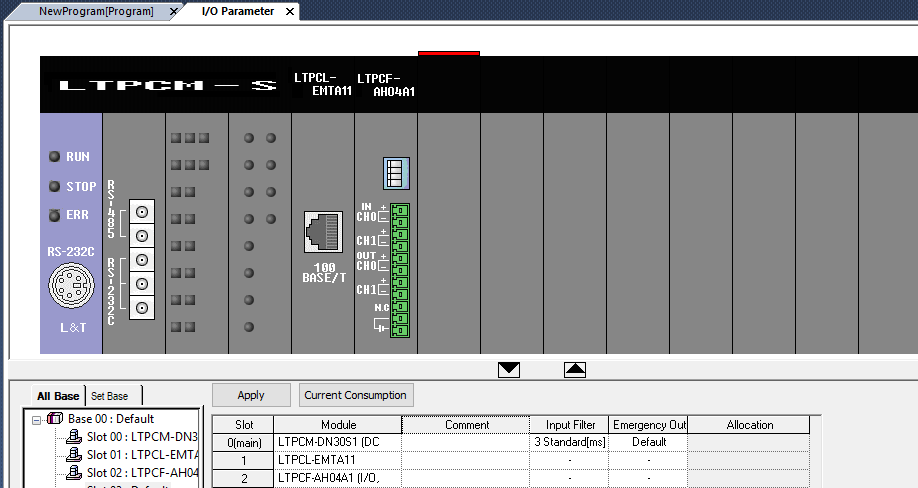
# 3.2 Hardware configuration of CPU and expansion modules

Go to IO parameter where you have to select the CPU type which is LTPCM-DN30S1 mounted in Lab kit 0(main) main slot will be fixed for CPU.





Subsequently select the expansion modules in slot 1 LTPCL-EMTA11(Available in communication module list) and slot 2 LTPCF-AH04A1 (Available in Special module analog I/O module list). Cross check in the kit modules should be in the same order as mounted in the PLC kit. Finally your configuration will look like below.

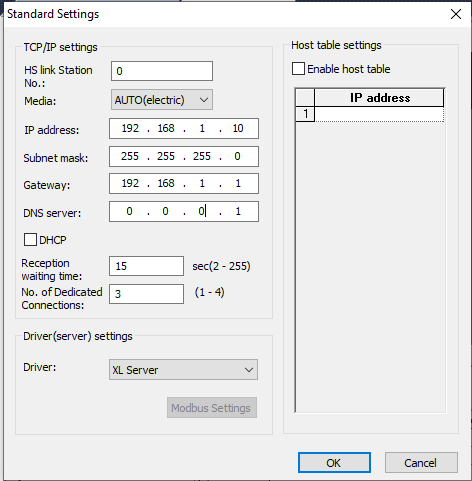
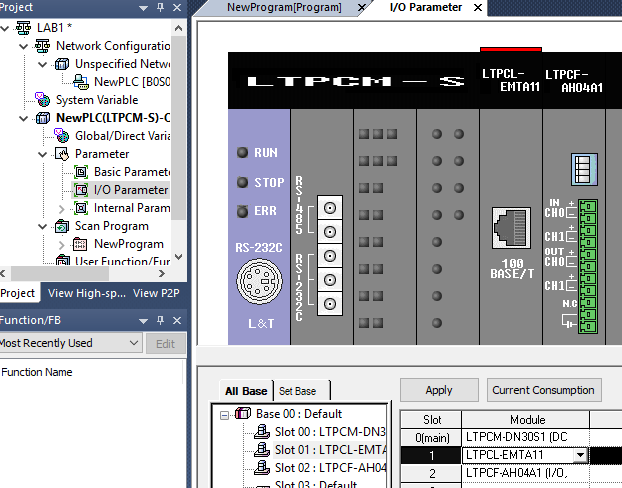


# 3.3 Brief Description and configuration of expansion modules added in the kit.

# 3.3.1 LTPCL-EMTA11 Fast Ethernet module

LTPCL-EMTA11 module provides fast Ethernet communication with L&T PLC’s. It supports Ethernet II, IEEE 802.3 standard functions, HS link function for HS data communication between L&T modules, Supports Modbus TCP/IP communication for third party devices and Downloading function from PC to PLC. To Setup Ethernet module double clicks the Ethernet module in IO parameter and enters the IP address as below.

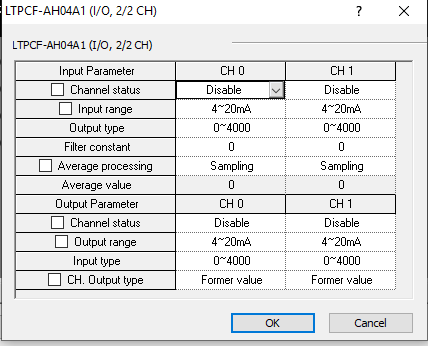
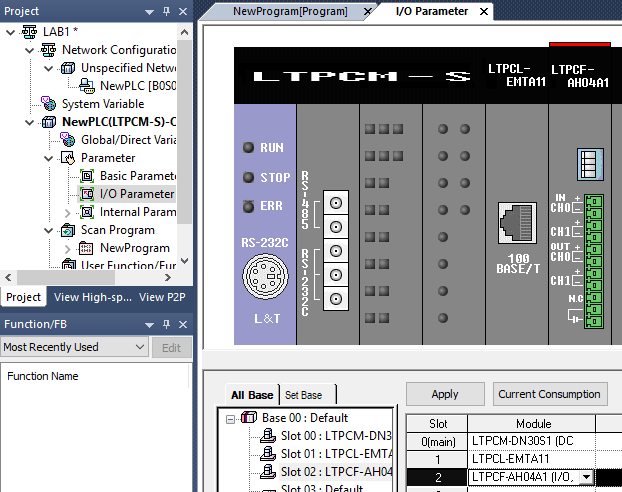
To download the program from PC to PLC through Ethernet first Ethernet module setting should be downloaded from USB or RS232 cable for the first time.



**3.3.3 LTPCF-AH04A1 Analog Input Output combination module**

LTPCF-AH04A1 module is 2 Analog inputs (Voltage/Current), 2 Analog Output (Voltage/Current) combination modules. The module has a 12 bit resolution provides different ranges of A/D conversion values for different ranges detailed setting of analog module will be explained in analog module programming lab.

To set analog module double click the analog module LTPCL-AH04A1 in IO parameter as show below.



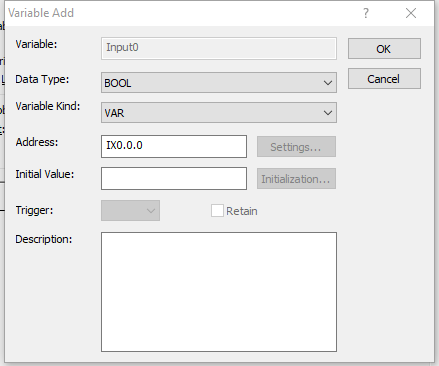
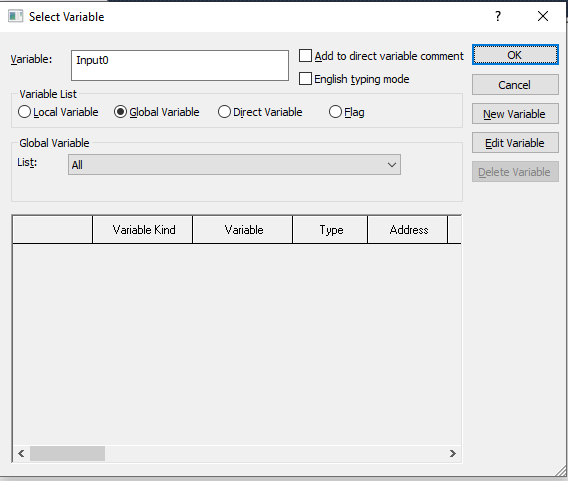
**4. Editing simple ladder program**

To edit the ladder program click on the scan program in project window and open new program.

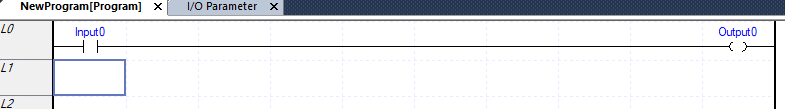
To insert NO contact either press F3 or click on NO contact symbol in contacts tool bar and click on ladder editing area to place the NO Contact.



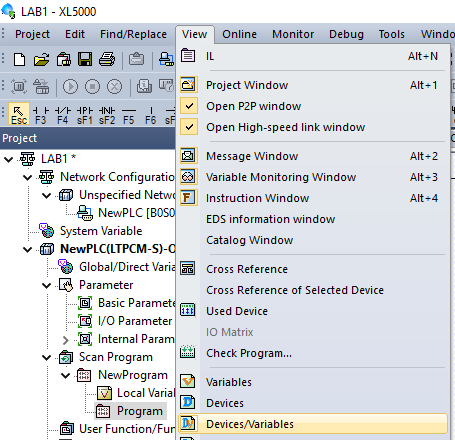
After placing the NO contact enter the variable name by selecting Global variable for Ex. Input0 click OK to give address IX0.0.0 which is the first input of test kit.

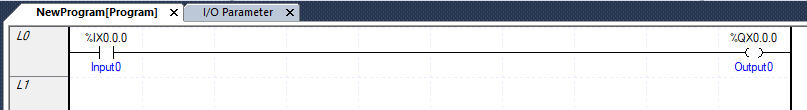


To add a coil either press F9 or click on coil symbol in contact tool bar edit variable name as Ouptut0 and give address as QX0.0.0 which is first output of test kit. After adding NO contact and Coil your ladder program looks like below and it’s ready to download



To view both address and variable name in ladder editing area go to view menu and select Device/variables. Toggle between variables, Devices and Variables to experience how the variables looks in ladder editing area.



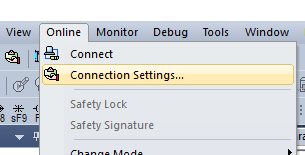
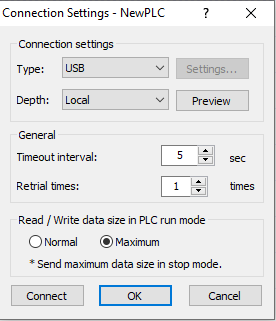


**5. Downloading the program to PLC.**

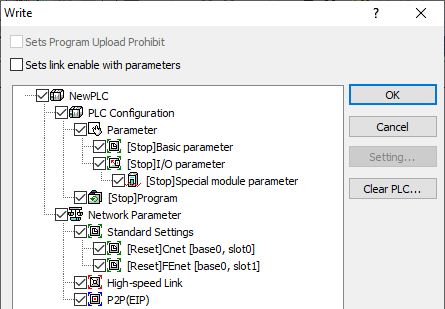
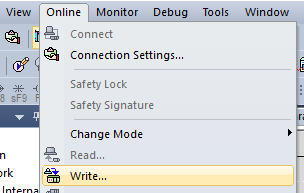
**5.1 Downloading through USB.**

To download the program to PLC through USB make sure Mini USB cable is connected to PLC USB port and any of USB port in PC.

Go to Online menu select connection settings select type as USB and depth Local and press connect it will give a popup PLC connected successful if connected.

****

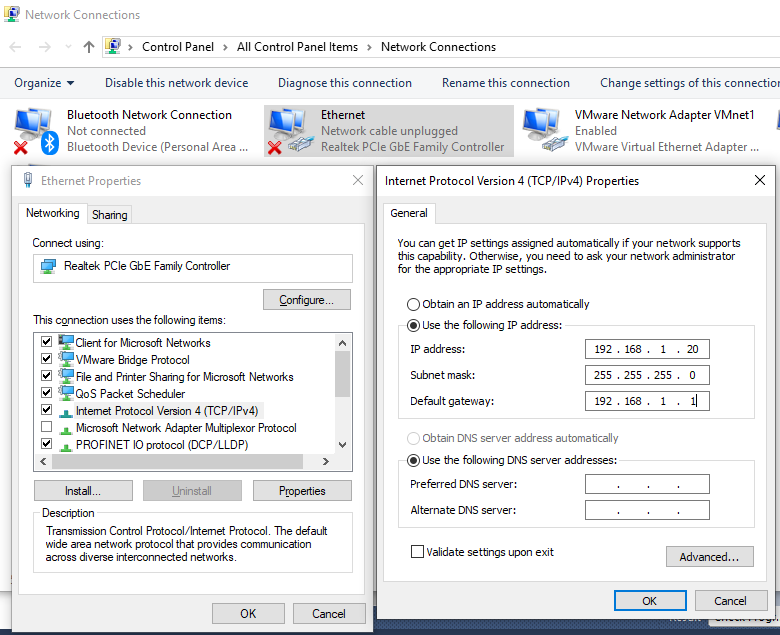
After connection is successful again go to online and click on write to start downloading the program. Select all the section for downloading first time. Make sure that PLC RUN/STOP key switch is in STOP mode to download the program after downloading PLC can be put it in RUN mode through software though key switch is in STOP mode.



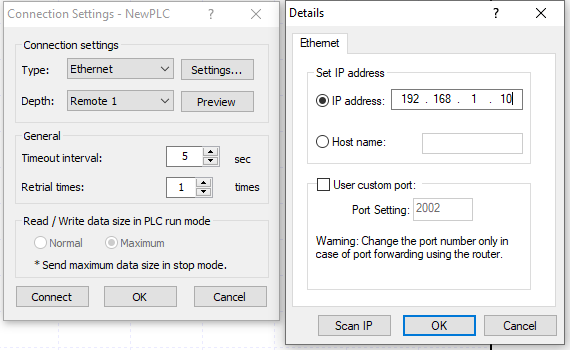
**5.2 Downloading through Ethernet**

To download the program to PLC through Ethernet, make sure Ethernet cable is connected to PLC Ethernet module and PC Ethernet port and basic Ethernet settings is done in PLC Ethernet module as explained in previous section.

For successful downloading both PLC and PC should be in same Ethernet subnet.To make the PC to same subnet go to Network settings select change adaptor option and double click on Ethernet port,Go to IPV4 and set same subnet IP address to PC. Remember both PC and PLC should be in same IP series but not same IP address.



Go to Online menu select connection settings select type as Ethernet and depth remote press settings and enter IP address of the PLC which is already edited and press connect it will give a popup PLC connected successful if connected. Make sure you have downloaded the program through USB first.

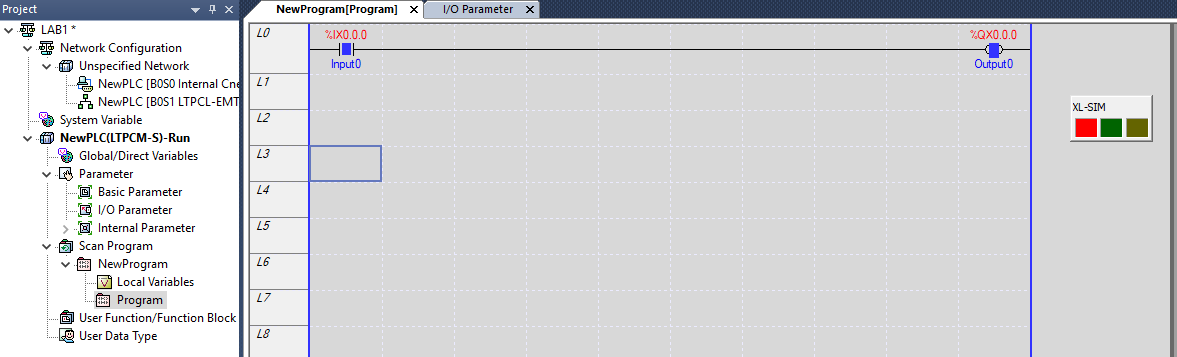


After connection is successful again go to online and click on write to start downloading the program.

**6. Monitoring the Program and variables.**

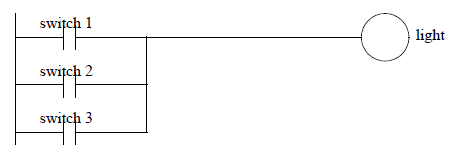
To start monitoring go to Monitor menu and press start monitoring. This will show the real time status of PLC program.

Now turn on the PLC input IX0.0.0 in Lab kit and output QX0.0.0 will turn on as per the Logic and your Ladder looks like below.



**Sample program** –

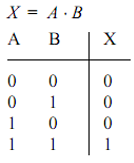
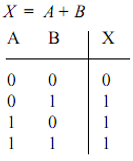
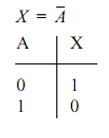
1. Turning on any one switch will activate the load



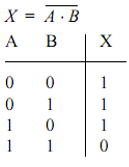
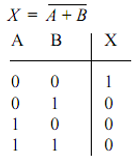
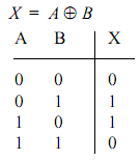
**Exercise Problems** -

1. Develop PLC program for the Logic Gates

AND OR NOT

NAND NOR EXOR

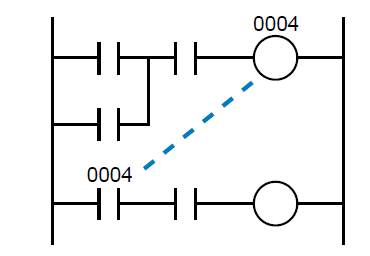
EXPERIMENT NO. 1b DATE:

**Implementation of combinational circuits**

**Aim:** To understand the working of a Programmable Logic Controller and to implement

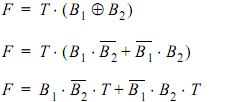
the digital circuits in PLC.

**Sample Program** –

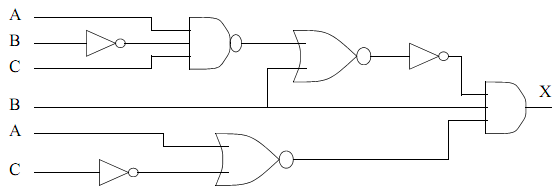


**Exercises problems:**

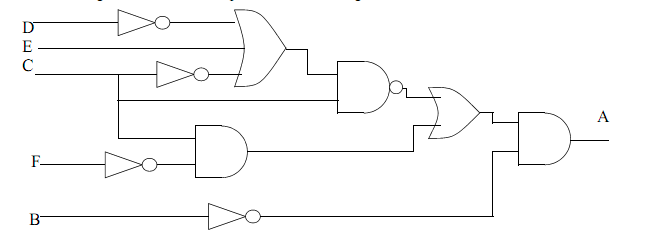
1. Develop PLC Ladder logic for Boolean equation (don’t simplify the expression ):



1. 

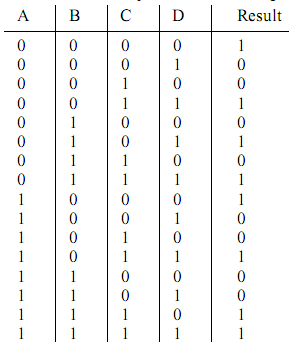


1. 

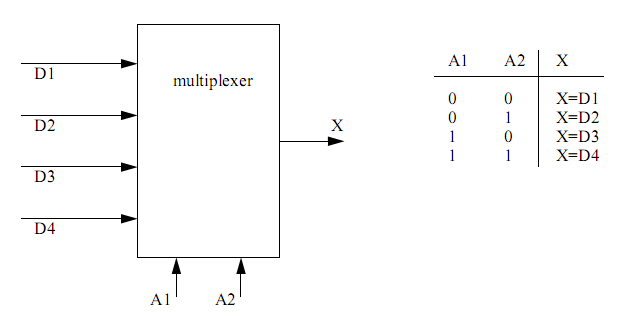


1. Develop Ladder Logic for a car door/seat belt safety system. When the car door is open, and the seatbelt is not done up, the ignition power must not be applied. If all is safe then the key will start the engine.
2. Design a motor controller that has a forward and a reverse button. The motor forward and reverse outputs will only be on when one of the buttons is pushed.

1. Write Boolean equation ladder logic for the following truth table.



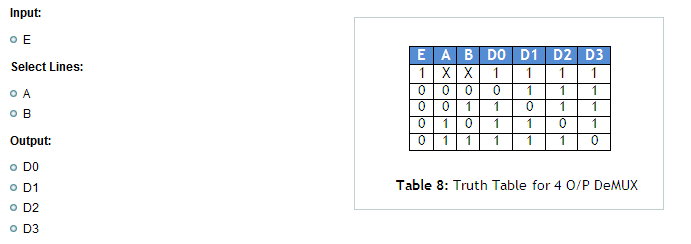
1. Develop the Ladder logic for the following truth table.



**Fig 1**

In Figure a multiplexer is shown that will take one of four inputs bits D1, D2, D3 or D4 and make it the output X, depending upon the values of the address bits, A1 and A2.

1. Develop the Ladder logic for the following truth table.



EXPERIMENT NO.2 DATE:

**Timer functions and problems**

**Aim:** To understand and implement Timer functions in a PLC.

**Working of Timers:**

An on-delay timer will wait for a set time after a line of ladder logic has been true before turning on, but it will turn off immediately. An off-delay timer will turn on immediately when a line of ladder logic is true, but it will delay before turning off. Consider the example of an old car. If you turn the key in the ignition and the car does not start immediately, that is an on-delay. If you turn the key to stop the engine but the engine doesn’t stop for a few seconds that is an off delay. An on-delay timer can be used to allow an oven to reach temperature before starting production. An off delay timer can keep cooling fans on for a set time after the oven has been turned off.

**TON: On delay :**

**Symbol**



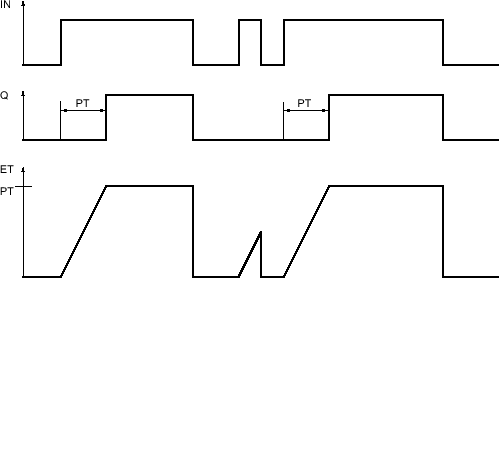
|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Data type** | **Memory area** | **Description** |
| IN | BOOL | [I, Q, M, D, L](ms-its:C:\Program%20Files\Siemens\Automation\Portal%20V10\Help\en-US\ProgKOP2MenUS.chm::/10867183243/10383119371.htm) | Start input |
| PT | TIME | I, Q, M, D, L or constant | Time by which the rising edge is delayed at the IN  input. |
| Q | BOOL | I, Q, M, D, L | Output, which is delayed by the time PT. |
| ET | TIME | I, Q, M, D, L | Elapsed time |

**Description**

You can use the "On delay" operation to delay a rising edge by the time set at PT. The "On delay" operation is executed when the result of logic operation (RLO) changes from "0" to "1" at input IN (rising edge). When the operation is started, the time set for PT starts running. When the PT time expires, output Q has signal state "1". Output Q remains set as long as the start input is still "1". If there is a signal change at the start input from "1" to "0", output Q is reset. The timer function is started again when a new positive edge is detected at the start input.

The ET output supplies the time that has elapsed since the last rising edge at the IN input. This time starts at T#0s and ends when the value set for the PT timer is reached. The elapsed time can be queried at output ET as long as input IN has signal state "1".

'When inserting the "On delay" operation, an instance data block is created in which the operation data is saved.

Timing Diagram

**TOF: Off delay :**

**Symbol**



|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Data type** | **Memory area** | **Description** |
| IN | BOOL | [I, Q, M, D, L](ms-its:C:\Program%20Files\Siemens\Automation\Portal%20V10\Help\en-US\ProgKOP2MenUS.chm::/10867183243/10382008587.htm) | Start input |
| PT | TIME | I, Q, M, D, L or constant | Time by which the falling edge is delayed at the IN input. PT must be positive. |
| Q | BOOL | I, Q, M, D, L | Output, which is delayed by the time PT. |
| ET | TIME | I, Q, M, D, L | Elapsed time |

**Description**

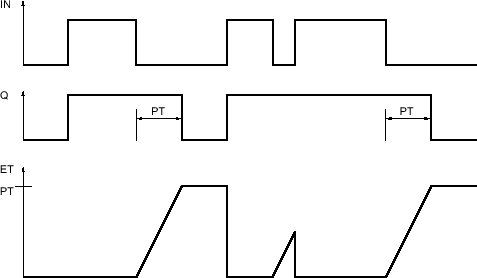
You can use the "Off delay" operation to delay a falling edge by the time set at PT. The Q output is set when the result of logic operation (RLO) at input IN changes from "0" to "1". When the signal state at the IN input switches back to "0", the time set at PT starts. Output Q remains set as long the time set at PT is running. The Q output is reset when the PT time expires. If the signal state at the IN input changes to "1" before the time set at PT time expires, the timer is reset. The signal state at the Q output will continue to be "1".

It is possible to query how long the current timer function has been running at output ET. This time starts at T#0s and ends when the value set for the PT timer is reached. When the time set at PT expires, output ET remains set to the current value until input IN changes back to "1". If the IN input switches to "1" before the PT time has expired, the ET output is reset to the value T#0.

'When inserting the "OFF delay" operation, an instance data block is created in which the operation data is saved.

The "Off delay" operation requires a preceding logic operation for the edge evaluation. It can placed within or at the end of the network.

**Timing diagram**



**Exercises**

1. Develop the ladder logic that will run a cooling fan for 15 seconds after the machine has been turned off.
2. Write a program that will turn on a flashing light for the first 15 seconds after a PLC is turned on. The light should flash for half a second on and half a second off.

1. In dangerous processes it is common to use two palm buttons that require a operator to use both hands to start a process . To develop this there are two inputs that must be turned on within 0.25 sec of each other before a machine cycle may begin.
2. We are using a pneumatic cylinder in a process. The cylinder can become stuck, and we need to detect this. Proximity sensors are added to both endpoints of the cylinder’s travel to indicate when it has reached the end of motion. If the cylinder takes more than 2 seconds to complete a motion this will indicate a problem. When this occurs the machine should be shut down and a light turned on. Develop ladder logic that will cycle the cylinder in and out repeatedly, and watch for failure.
3. Consider the control of a heating oven. The system is started with a Start button that seals in the Auto mode. This can be stopped if the Stop button is pushed. (Remember: Stop buttons are normally closed.) When the Auto goes on initially the TON timer is used to sound the horn for the first 10 seconds to warn that the oven will start, and after that the horn stops and the heating coils start. When the oven is turned off the fan continues to blow for 300s or 5 minutes after.
4. Design a ladder diagram that will cause a stepper motor so that it moves 10 steps forward, waits for 20 seconds and then cause the motor to move 10 steps in the reverse direction.[the stepper input pulse is 0.5 second on and 0.5 second off.] the 2 inputs used are – a. start and b. stop. the 2 outputs used are – a. output(motor forward/reverse) and b. Stepper pulse out.

EXPERIMENT NO. 3 DATE:

##### Counter functions and problems

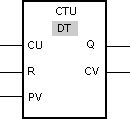
**Aim:** To understand and implement counter functions of a PLC.

**Working:**

There are three counter types: count-up, count-down and counter up - down. When the input to a count-up counter goes true the accumulator value will increase by 1 (no matter how long the input is true.) If the accumulator value reaches the preset value the counter Q bit will be set. A count-down counter will decrease the accumulator value until the preset value is reached. Two different inputs will be available in a count up – down counter, either to increase or decrease the accumulator value.

**CTU: Count up:**

Symbol



|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter**  **English** | **Data type** | **Memory area** | **Description** |
| CU | BOOL | [I, Q, M, D, L](ms-its:C:\Program%20Files\Siemens\Automation\Portal%20V10\Help\en-US\ProgKOP2MenUS.chm::/10867183243/10460030219.htm) | Count input |
| R | BOOL | I, Q, M, D, L | Reset input |
| PV | SINT, INT, DINT, USINT, UINT, UDINT | I, Q, M, D, L or constant | Preset count |
| Q | BOOL | I, Q, M, D, L | Counter status |
| CV | SINT, INT, DINT, USINT, UINT, UDINT | I, Q, M, D, L | Actual count value |

You can select the data type for the operation from the "DT" drop-down list.

**Description**

You can use the "Count up" operation to increment the value at the CV output. If the signal state at the CU input changes from "0" to "1" (positive edge), the operation is executed and the current count value at the CV output is incremented by one. The first time the operation is executed, the current count at the CV output is set to zero. The count is incremented each time a positive edge is detected, until it reaches the high limit for the specified data type at the CV output. When the high limit is reached, the signal state at the CU input no longer has an effect on the operation.

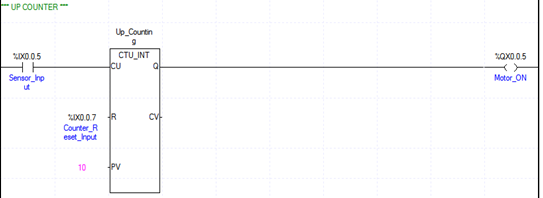
The counter status can be queried at the Q output. The signal state at output Q is determined by the PV parameter. If the current count value is greater than or equal to the value of the PV parameter, the Q output is set to signal state "1". In all other cases, the signal state at the Q output is "0".

The value at the CV output is reset to zero when the signal state at R input changes to "1". As long as there is the signal state "1" at R input, the signal state at input CU has no effect on the operation.

'When inserting the "Count up" operation, an instance data block is created in which the operation data is saved.

The "Count up" operation requires a preceding logic operation for the edge evaluation. It can placed within or at the end of the network.

**Example**

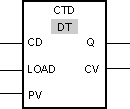


When the signal state at the input I 0.0 changes from "0" to "1", the "Count up" operation is executed and the current count at the MW30 output is increased by one. With each further positive edge, the count value is incremented until the high limit value of the specified data type (32 767) is reached.

The value at the MW20 parameter is applied as the limit for determining the output Q 4.0. Output Q 4.0 has signal state "1" as long as the current count is greater than or equal to the value at the MW20 parameter. In all other cases, output Q 4.0 has signal state "0".

**CTD: Countdown :**

Symbol



|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter**  **English** | **Data type** | **Memory area** | **Description** |
| CD | BOOL | [I, Q, M, D, L](ms-its:C:\Program%20Files\Siemens\Automation\Portal%20V10\Help\en-US\ProgKOP2MenUS.chm::/10867183243/10460032779.htm) | Count input |
| LOAD | BOOL | I, Q, M, D, L | Load input |
| PV | SINT, UINT, DINT, USINT, UINT, UDINT | I, Q, M, D, L or constant | Preset count |
| Q | BOOL | I, Q, M, D, L | Counter status |
| CV | SINT, UINT, DINT, USINT, UINT, UDINT | I, Q, M, D, L | Actual count value |

You can select the data type for the operation from the "DT" drop-down list.

Description

You can use the "Count down" operation to decrement the value at the CV output. If the signal state at the CD input changes from "0" to "1" (positive edge), the operation is executed and the current count value at the CV output is decremented by one. The first time the operation is executed, the current count at the CV output is set to zero. Each time a positive edge is detected, the count value is further decremented until it reaches the low limit value of the specified data type. When the low limit value is reached, the signal state at the CD input has no further effect on the operation.

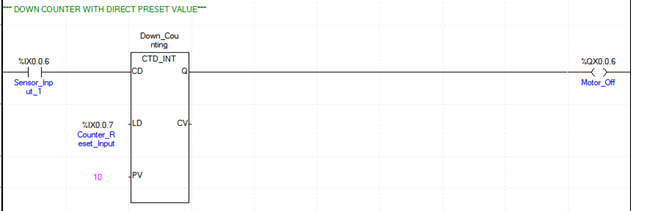
The counter status can be queried at the Q output. If the current count value is less than or equal to zero, output Q is set to signal state "1". In all other cases, the signal state at the Q output is "0".

The value at the CV output is set to the value of the PV parameter when the signal state at LOAD input changes to "1". As long as there is the signal state "1" at the LOAD input, the signal state at the CD input has no effect on the operation.

'When inserting the "Count down" operation, an instance data block is created in which the operation data is saved.

The "Count down" operation requires a preceding logic operation for the edge evaluation. It can placed within or at the end of the network.

**Example**

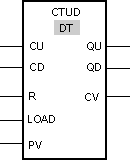


When the signal state at the input I 0.0 changes from "0" to "1", the "Count down" operation is executed and the value at the MW30 output is decreased by one. With each further positive edge, the count value is decremented until the low limit value of the specified data type (-32 768) is reached.

Output Q 4.0 has signal state "1" as long as the current count value is less than or equal to zero. In all other cases, output Q 4.0 has signal state "0".

**CTUD: Count up and down**

Symbol



|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter**  **English** | **Data type** | **Memory area** | **Description** |
| CU | BOOL | [I, Q, M, D, L](ms-its:C:\Program%20Files\Siemens\Automation\Portal%20V10\Help\en-US\ProgKOP2MenUS.chm::/10867183243/10460035339.htm) | Count up input |
| CD | BOOL | I, Q, M, D, L | Count down input |
| R | BOOL | I, Q, M, D, L | Reset input |
| LOAD | BOOL | I, Q, M, D, L | Load input |
| PV | SINT, UINT, DINT, USINT, UINT, UDINT | I, Q, M, D, L or constant | Preset count |
| QU | BOOL | I, Q, M, D, L | Status of the incremental counter |
| QD | BOOL | I, Q, M, D, L | Status of the decremental counter |
| CV | SINT, UINT, DINT, USINT, UINT, UDINT | I, Q, M, D, L | Actual count value |

You can select the data type for the operation from the "DT" drop-down list.

**Description**

You can use the "Count up and down" operation to increment and decrement the count value at the CV output. If the signal state at CU input changes from "0" to "1" (positive edge), the current count value is incremented by one and stored in the CV output. If the signal state at CD input changes from "0" to "1" (positive edge), the count value at the CV output is decremented by one. If there is a positive edge at the CU and CD inputs in one program cycle, the current count value at the CV output remains unchanged.

The count value can be incremented until it reaches the high limit value of the data type specified at the CV output. When the high limit value is reached, the count value is no longer incremented on a positive edge. When the low limit value of the specified data type is reached, the count value is not decremented any further.

When the signal state at the LOAD input changes to "1", the count value at the CV output is set to the value of the PV parameter. As long as there is the signal state "1" at the LOAD input, the signal state at the CU and CD inputs has no effect on the operation.

The count value is set to zero when the signal state at input R changes to "1". As long as the signal state at the R input is "1", changes of signal state at inputs CU and CD and LOAD have no effect on the "Count up and down" operation.

The incremental counter status can be queried at QU output. If the current count value is greater than or equal to the value of the PV parameter, QU output has signal state "1". In all other cases, the signal state at the QU output is "0".

The decremental counter status can be queried at QD output. If the current count value is less than or equal to zero, output QD has signal state "1". In all other cases, the signal state at the QD output is "0".

'When inserting the "Count up and down" operation, an instance data block is created in which the operation data is saved.

**Exercises:**

1. An indicating light is to go on when the count reaches 12. The light is then to go off when a count of 20 reached. Use reset to reset counters at any time.
2. A pneumatic cylinder is used to remove 5 out of every 10 parts from a conveyer. The cycle repeats after every 10 parts. Realize it using PLC ladder logic.
3. In a glass manufacturing plant glasses are stacked in a double shelved trolley. Initially 15 glasses are stacked in first shelf and another 15 glasses in second shelf. Turn ON the output motor of trolley to move it when it contains total 30 glasses.
4. A buffer can hold up to 10 parts. Parts enter the buffer on a conveyor controller by output conveyor. As parts arrive they trigger an input sensor enter. When a part is removed from the buffer they trigger the exit sensor. Write a program to stop the conveyor when the buffer is full, and restart it when there are fewer than 10 parts in the buffer. As normal the system should also include a start and stop button.
5. An auditorium consists of two entrances and one exit. Twenty seats are placed inside that and both the entrances can allow the first 20 entries. After 20 entries both the doors shall be closed automatically with a siren sound for 5 seconds. If there is an exit, then entrance A shall be open and B remains closed. An indicating lamp will glow to indicate the door opening. A reset switch shall reset all the system at any time.
6. A motor will be connected to a PLC and controlled by two switches. The GO switch will start the motor, and the STOP switch will stop it. If the motor is going, and the GO switch is thrown, this will also stop the motor. If the STOP switch was used to stop the motor, the GO switch must be thrown twice to start the motor. When the motor is running, a light should be turned on (a small lamp will be provided).
7. Develop a program that will latch on an output B (O/1), 20 seconds after input A (I/1) has been turned on. The timer will continue to cycle up to 20 seconds, and reset itself, until input A has been turned off. After the third time the timer has timed to 20 seconds, the output B will be unlatched.
8. A conveyor is run by switching on or off a motor. We are positioning parts on the conveyor with an optical detector. When the optical sensor goes on, we want to wait 1.5 seconds, and then stop the conveyor. After a delay of 2 seconds the conveyor will start again. We need to use a start and stop button - a light should be on when the system is active. Gages have been attached that indicate good or bad. If the part is good, it continues on. If the part is bad, we do not want to delay for 2 seconds, but instead actuate a pneumatic cylinder.
9. A handicap door opener has a button that will open two doors. When the button is pushed (momentarily) the first door will start to open immediately, the second door will start to open 2 seconds later. The first door power will stay open for a total of 10 seconds, and the second door power will stay on for 14 seconds. Use a timing diagram to design the ladder logic.
10. Design the ladder diagram for a device that will count parts as they pass by an inspection stand. The sensing device for the PLC is a switch that will close each time a part passes. This switch is connected to IN1 of the PLC. A reset switch, IN2, is also connected to the PLC to allow the operator to manually reset the counter. After 15 parts have passed the inspection stand, the PLC is to reset the counter to again begin counting parts and turn on a light which must stay on until reset by a second reset switch connected to IN3. The output from the PLC that lights the light is OUT111.

EXPERIMENT NO.4 DATE:

**Compare and arithmetic instructions**

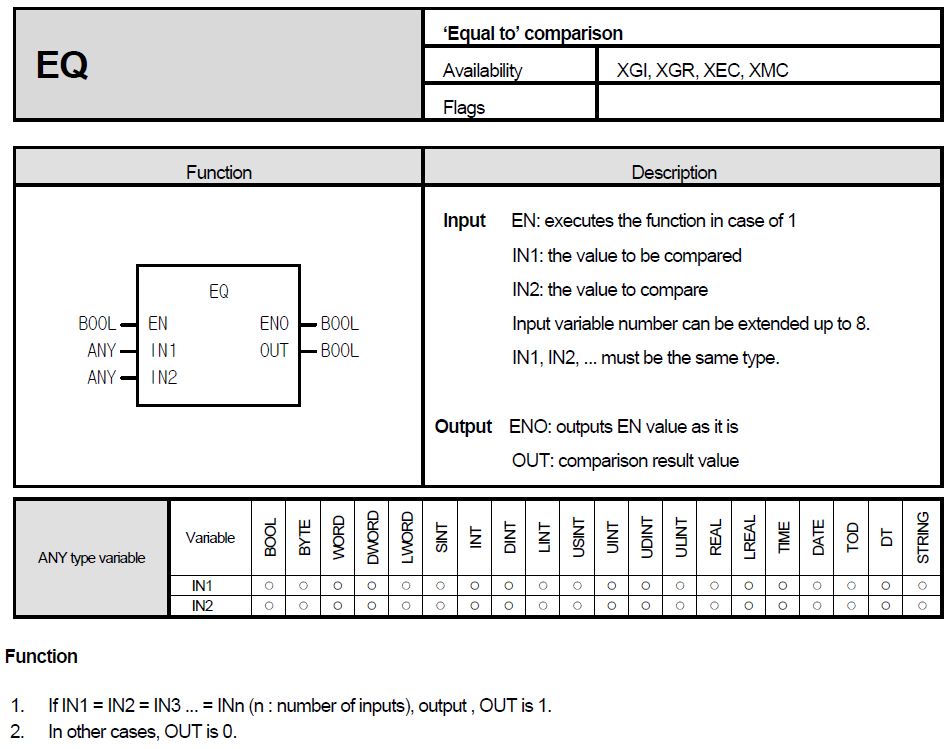
**Aim**: To understand the compare and arithmetic operations of a PLC.

1. **Compare operations:**

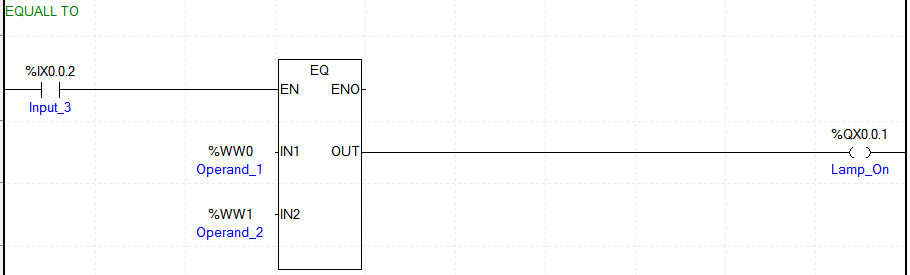
The compare Byte instruction is used to compare two operands. Comparisons include: Operand1 = Operand2, Operand1 >= Operand2, Operand1 <= Operand2, Operand1 > Operand2, Operand1 < Operand2, Operand1 <> Operand2. The output is ON when the comparison is true.

Consider a comparison where Operand1 > Operand2:

Symbol



Example

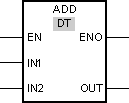


Output Q X0.0.1 is set when the following conditions are fulfilled:

* Inputs I X0.0.2 has signal state "1".
* The condition of the comparison operation is fulfilled (MW0 = MW1).

1. **Arithmetic functions :**
   1. **ADD**

Symbol

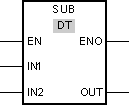


You can use the "Add" operation to add the value at the IN1 input to the value at the IN2 input and query the sum at the OUT output (OUT =IN1+IN2).

The operation is only executed if the signal state at the enable input EN is "1". If no errors occur during execution of the operation, the ENO output also has signal state "1".

* 1. **Subtract**

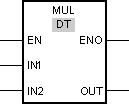
Symbol



You can use the "Subtract" operation to subtract the value at input IN2 from the value at input IN1 and query the difference at the OUT output (OUT =IN1-IN2).

* 1. **Multiply**

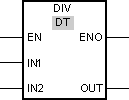
Symbol



You can use the "Multiply" operation to multiply the value at the IN1 input to the value at the IN2 input and query the product at the OUT output (OUT =IN1\*IN2).

* 1. **Division**

Symbol



You can use the "Divide" operation to divide the value at the IN1 input by the value at the IN2 input and query the quotient at the OUT output (OUT =IN1/IN2).

* 1. **Increment**

Symbol

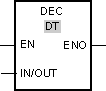


You can use the "Increment" operation to change the tag at the IN/OUT output to the next higher value and query the result. The "Increment" operation can only be started when the signal state at the EN enable input is "1". If no overflow error occurs during the execution, the ENO output also has signal state "1".

If the signal state is "0" at the enable input EN, the operation is not executed. In this case, the ENO enable output is reset.

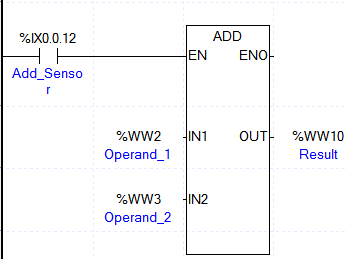
* 1. **Decrement**

Symbol



With the "Decrement" operation, you can change the value of the tag at the output IN/OUT output to the next lower value and query the result. Execution of the "Decrement" operation is started when the signal state at the EN enable input is "1". If no underflow error occurs during the execution, the ENO output also has signal state "1".

**Example of ADD instruction:**



If the input IX 0.0.12 has signal state "1", the "Add" operation is executed. The value at the WW2 input is added to the value at the WW3 input. The result of the addition is stored at the WW10 output. If no errors occur during execution of the operation, the ENO output has signal state "1".

**Exercises:**

* 1. Compare the values of two counters and enable an output if the valves are same.
  2. Increment / Decrement the valve in a memory register by enabling an input.
  3. Implement the following function



* 1. A machine ejects parts into three chutes. Three optical sensors (A, B and C) are positioned in each of the slots to count the parts. The count should start when the reset (R) button is pushed. The count will stop, and an indicator light (L) turned on when the average number of parts counted equals 10.
  2. Create a ladder logic program that will start when input *A* is turned on and calculate the series below. The value of *n* will start at 0 and with each scan of the ladder logic *n* will increase by 2 until n=20. While the sequence is being incremented, any change in *A* will be ignored.



EXPERIMENT NO. 5 DATE:

**Control of Glass Inspection Station**

**Aim:** To understand and simulate the control of glass inspection process using PLC.

**Description:**

The objective is to control the inspection station of a glass manufacturing system for automobiles. Glasses flow in regular intervals on conveyor.

The required cycle is as follows:

1. Master ON is pressed to start the operation.
2. If a glass is detected on conveyor, the conveyor will start to move.
3. If the glass reaches inspection station, then the conveyor will stop.
4. 15 seconds of time delay is given for operator to inspect. After time delay a lamp glows.
5. Now the operator needs to acknowledge if the glass is OK or NOTOK by pressing green or red push buttons respectively. Lamp goes off then.
6. If the glass is NOTOK then another 3 seconds is delayed to unload it.
7. If the glass is OK then conveyor starts again thus continuing the operation.
8. Master OFF is pressed to stop the operation.
9. An emergency stop button will reset the operation at any instant.

Lamp

OK NOTOK

ON OFF

Glass

Conveyor

Motor

Sensor

EXPERIMENT NO. 6 DATE:

**Control of bottle filling station**

**Aim:** To understand and simulate the control of bottle filling process using PLC.

**Description:**

The objective is to control a bottle filling station. Bottles move on conveyor with equal space separating each other. This system involves two operations – filling operation and sealing operation.

1. Master ON push button is pressed to turn ON the system.
2. System is put in Auto mode by pressing Auto mode push button.
3. Conveyor starts to move.
4. Conveyor stops if bottle reaches filling station.
5. Bottle stays in filling station for 5 seconds.
6. After 5 seconds conveyor starts to move.
7. Bottle reaches the next station thus turning off the conveyor.
8. Here bottle is sealed in 5 seconds.
9. After 5 seconds conveyor starts to move.
10. Operation continues. Master OFF push button is used to turn off the system.
11. An emergency push button will reset the operation at any instant.

Sealing station

Filling station

Conveyor

Sensor 2

Sensor 1

Station 2

Station 1

Motor

EXPERIMENT NO. 7 DATE:

**Control of 4 way traffic signal**

**Aim:** To understand and simulate the control of traffic signal using PLC.

**Description:**

The objective is to control the operation of a traffic signal. The working is as follows:

1. There are four roads connected as shown in fig. Each road consists of 3 signals green, yellow and red.
2. Initially green signal of road 1 is ON. After 60 seconds Green signal of road 1 goes off and Yellow signal of road 1 is ON. After 5 seconds Yellow signal of road 1 goes off and red signal is ON. During this operation of road 1 other roads show red signal.
3. After road 1 shows red signal, cycle repeats for road 2. During this operation other roads show red signal.
4. After road 2 shows red signal, cycle repeats for road 3. During this operation other roads show red signal.
5. After road 3 shows red signal, cycle repeats for road 1.

1

4

3

2

R

Y

G

R

Y

G

R Y G

R Y G

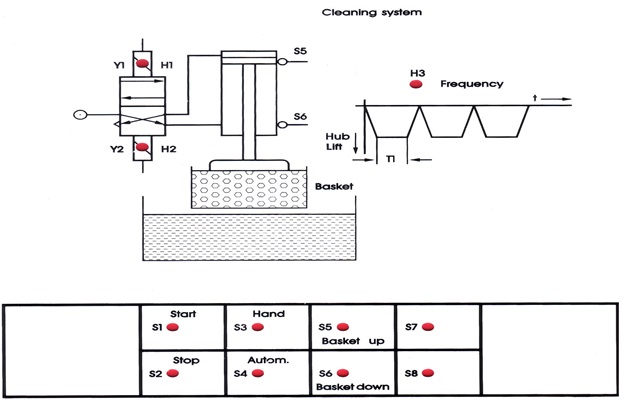
EXPERIMENT NO. 8. DATE:

**Control of cleaning system**

**AIM:** To understand and simulate the control of cleaning using PLC.

**Description -**

1. The tank of a cleaning system is to be lowered and lifted pneumatically. After the lowering- lifting cycle has been completed 3 times, the basket is to remain at the double-acting work cylinder in its retracted position.
2. During the cleaning process, the basket is to remain in the cleaning system for a period of 5 seconds. The system is switched on using the S1 pushbutton “Start” and switched off using the S2 pushbutton “Stop”.
3. Using the mode S3, manual operation is possible, using S4, automatic operation can be selected. If by accident, both buttons are activated, manual operation will take priority. During automatic operation, the process proceeds automatically via the S1 pushbutton. In manual operation, the cylinder is extended using the pushbutton S7 and retracted using S8.
4. In manual operation, it must be guaranteed that the immersion process does not take longer than 5 seconds and is automatically switched off after 3 cycles. The immersion cycle is to be displayed by means of the LED H3.
5. The automatic operation must be re-started after each sequence (3 cycles) using the pushbutton S1.



EXPERIMENT NO. 9 DATE:

**Tank filling device simulator system**

**Aim**

To understand and simulate the control of inspection process using PLC.

**Description:**

A tank filling device simulator consists of 3 tanks that are equipped with signal encoders. The tank filling device simulator system can be switched on using the S1 pushbutton called “Start”. For switching off the device simulator system, S2 pushbutton “Stop” can be used. For the "Max" notifications of tank, the signal encoders S3, S5 and S7 are used. For the "Min" notifications, signal encoders S4, S6 and S8 are used. The storage tanks can be arbitrarily filled and emptied by hand manually.

Conditions:

1. For filling the tanks 1, 2 & 3, the valves Y1, Y2 and Y3 are used. Filling of the tanks must happen in series i.e H1,H2,&H3.
2. A control is to secure that after a “empty” ( Ex: for tank 1, S3 & S4 both must be OFF ) notification occurred, only 1 tank can be filled at a time in the specified order.
3. The filling of the next tank continues until the corresponding “full” notification has occurred.

(i.e for Ex : if S3 & S4 both are ON , then only H1 should stop filling & next H2 should start for filling)

1. Once all the tank fills, its water should start to drain out one after the other from H4, H5 & H6.

Prepare the PLC program.

****

MIN

MAX

OUT H6

OUT H5

OUT H4

IN