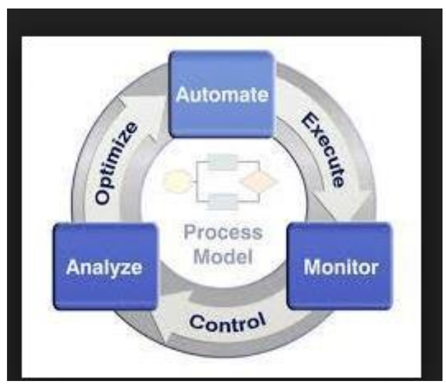
University of Jordan

Mechatronics Engineering Department

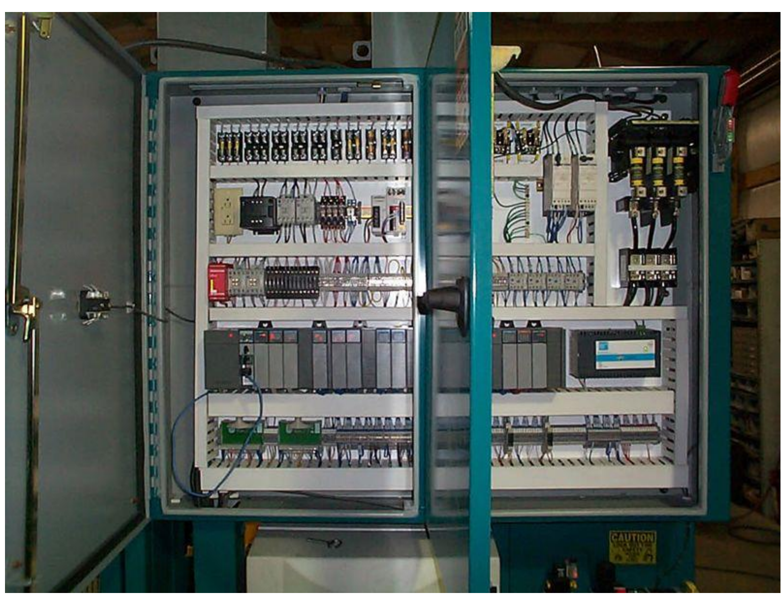
Automation & Process control Laboratory Manual

MX – 0908462



University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

Hard-Wiring and Timer Basics



<u>Administrative Policy of the Laboratory</u>

1) You are not allowed to smoke, eat or drink in the Laboratory. You are expected to conduct yourself professionally, and to keep your bench area *clean* and *neat*.

2) You are allowed to discuss the experiment and the results with your colleagues, but cheating and copying is strictly prohibited and will be taken very seriously. The student will earn a **ZERO** in the lab when caught.

3) All questions should be solved in order. Moreover, each student is expected to fully and clearly demonstrate her/his solution whenever required.

4) No one is allowed to leave the lab until she/he has cleaned and arranged her/his bench and turned off the PC she/he used.

5) Always ask your instructor to check your setup before turning the power on.

6) The above mentioned polices should be strictly followed. Note that disregarding any of the rules above will seriously affect your grade!

7) Makeup Midterm: There will be no make-up for the midterm. In case of medical/ or other disabling emergencies, the instructor should be notified before the midterm and his approval for missing the midterm should be obtained before the midterm. If for any reason the instructor could not be reached, the department secretary should be notified before the midterm. The phone number is 535-5000 Extension 23025

8) Class Attendance: Class attendance will be taken. University regulations regarding attendance will be strictly enforced.

9) **<u>You cannot change your lab time, if done zero grade will be considered as that lab mark.</u>**

You can obtain the experiment sheets from the following Internet web site:
https://elearning.ju.edu.jo/

Objective

To be familiar with Electrical Hard-Wired components and Electrical Timer basics.
After you Identifying Hard component & its symbols you can :

- Write any hard wired diagram for any logical expression or system .
- Read & wire any hard wired diagram .

Procedure:

This lab experiment is composed of two parts.

Part 1

use the control Panel Trainer which is exist in the lab to wire the following :

Exercise 1:

A certain lamp switched ON when Pressing on a Pushbutton & switched OFF when released the same Pushbutton .

Draw the circuit before wiring :

Exercise 2:

Use two pushbuttons and one lamp to make **AND gate**

Draw the circuit before wiring :

Exercise 3

Use two pushbuttons and one lamp to Make **OR gate**

Draw the circuit before wiring :

Exercise 4

A certain lamp is switch ON when pressing in a pushbutton & the lamp still on after releasing the Pushbutton .

Draw the circuit before wiring :

Exercise 5

A certain lamp is switch ON when pressing in a pushbutton & the Lamp still on after releasing the pushbutton , then pressing on another Pushbutton to switch off the Lamp

Draw the circuit before wiring :

Exercise 6

Use two pushbuttons and one lamp to make **XOR gate**

Draw the circuit before wiring :

Exercise 7

Use the component you need to satisfy the following Expression :

Draw the circuit before wiring :

Part 2 : Timers

Objective : To be familiar with different type of electrical Timer

Exercise 1

Read the data sheet for ETR Timer which is found at the back of the sheet to understand the different type of timers and its timing diagram .

Exercise 2 :

Make calibration for ETR4-69-A timer which is found in the control Panel Trainer To derive the relationship between Time & Range.

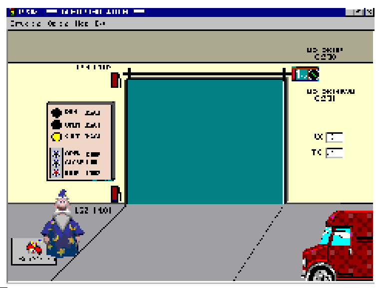
Exercise 3 :

Design the lighting system which is found in residential building stairs .

Discussion and Follow-up

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

Door Simulator



Objective

To be familiar with basic relay logic.

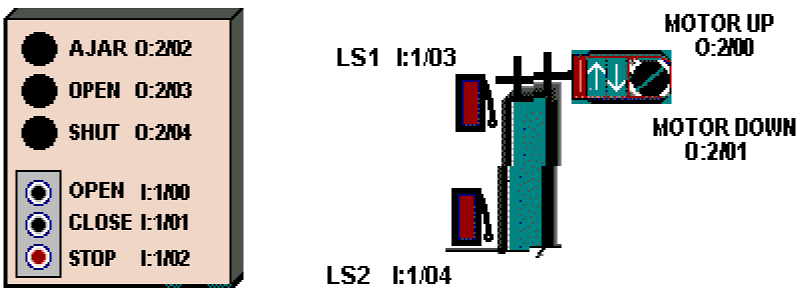
Pre-lab Preparation:

1- Read Chapter 3

Procedure:

From the Simulations Menu at the top of the screen, Select the Door Simulation.

Take the time to familiarize yourself with the components used in the Door system, and take particular note of the current state of the limit switches. When the door is in the closed position, both limit switches are in their activated state (Not Normal). Run your mouse over each switch, and you should see a tool-tip text box appear, which denotes that the selected switch is wired using a set of Normally Open contacts. With the door fully closed, what signal level would you expect to see at the limit switch inputs I:1/03 and I:1/04?



Student Programming Exercise #1:

In this exercise we want you to apply your knowledge of Relay Logic Instructions to design a program which will control the Door. The Door System includes a Reversible Motor, a pair of Limit Switches and a Control Panel, all connected to your PLC. The program you create will monitor and control this equipment while adhering to the following criteria:

- In this exercise the Open and Close pushbuttons will be used to control the movement of the door. Movement will not be maintained when either switch is released, and therefore the Stop switch is neither required nor used in this exercise. However, all other available Inputs and Outputs are employed in this exercise.

- Pressing the Open Switch will cause the door to move upwards (open) if not already fully open. The opening operation will continue as long as the switch is held down. If the switch is released, or if limit switch LS1 opens, the door movement will halt immediately.

- Pressing the Close Switch will cause the door to move down (close) if not already fully closed. The closing operation will continue as long as the switch is held down. If the switch is released, or if limit switch LS2 closes, the door movement will halt immediately.

- If the Door is already fully opened, Pressing the Open Switch will Not energize the motor.

- If the Door is already fully closed, Pressing the Close Switch will Not energize the motor.

- Under no circumstance will both motor windings be energized at the same time.

- The Open Lamp will be illuminated if the door is in the Fully Open position.

- The Shut Lamp will be illuminated if the door is in the Fully Closed position.

It is your responsibility to fully design, document, debug, and test your Program. Avoid the use of OTL or OTU latching instructions, and make a concerted effort to minimize the number of rungs employed.

Ensure that you have made effective use of both instruction and rung comments to clearly document your program. All I/O components referenced within your program should be clearly labeled, and rung comments should be employed to add additional clarity as required.

Student Programming Exercise #2:

In this exercise we want you to apply your knowledge of Relay Logic Instructions to design a program which will maintain the appropriate door movement once initiated by the operator. The Opening or Closing operation of the door will continue to completion even if the operator releases the pushbutton which initiated the movement. The program will adhere to the following criteria:

- Door movement will halt immediately when the Stop Switch is initially pressed, and will remain halted if the switch is released.

- Pressing the Open Switch will cause the door to Open if not already fully open. The opening operation will continue to completion even if the switch is released.

- Pressing the Close Switch will cause the door to Close if not already fully shut. The closing operation will continue to completion even if the Switch is released.

- If the Door is already fully opened, Pressing the Open Switch will Not energize the motor.

- If the Door is already fully closed, Pressing the Close Switch will Not energize the motor.

- Under no circumstance will both motor windings be energized at the same time.

- The Ajar Lamp will be illuminated if the door is NOT in either the fully closed or fully opened position.

- The Open Lamp will be illuminated if the door is in the Fully Open position.

- The Shut Lamp will be illuminated if the door is in the Fully Closed position.

It is your responsibility to fully design, document, debug, and test your Program. Avoid the use of OTL or OTU latching instructions, and make a concerted effort to minimize the number of rungs employed.

As before, ensure that you have made effective use of both instruction and rung comments to clearly document your program.

Student Programming Exercise #3:

In this exercise we want to introduce you to a simple programming technique for adding a bit of "Flash" to your program. We want you to make use of the PLC's Free Running Timer which can be viewed in the Data Table Display at location S2:4. This integer word contains a count which is incremented continuously by the PLC when it is in the Run mode, and it can come in quite handy at times for variety of purposes. In this exercise we want you to utilize this word as follows:

With the PLC in the Run mode, Display word S2:4 utilizing the Data Table display. Ensure that the Radix is set to Binary so that you can view the individual bits within the word. You should see a binary count in progress where the rate of change of each bit is directly related to it's position within the word. Bit 0 will have the highest rate, while Bit 1 will be 1/2 as fast as Bit 0, and Bit 2 half as fast as 1 etc. etc.

We want you to add a Lamp Flasher to your program by monitoring the state of one of these bits with an XIC instruction. I'm going to suggest using Bit 4 for this purpose, but depending upon the speed of your computer you may elect to substitute another Bit. With an actual AB PLC, the rate is consistent, but with LogixPro it varies from computer to computer.

Place an XIC instruction addressed to S:4/4 on the rung which controls either the Open or Shut Lamp in your previous program. Now download and Run this modified program to see the flashing effect achieved. The Lamp should flash at a reasonable rate whenever your program energizes the selected Lamp.

Now modify your program so that the following criteria is met:

- If the Door is fully open, the Open lamp will be energized but not flashing as was the case before.
- If the Door is opening, the Open lamp will flash while the door is in motion.
- If the Door is fully closed, the Shut lamp will be energized but not flashing as was the case before.
- If the Door is closing, the Shut lamp will flash while the door is in motion.
- The Ajar Lamp will flash if the door is stationary, and is not in the fully open or fully closed position. The Ajar Lamp will flash at a slower rate (1/4) then the other lamps.
- The Ajar Lamp will be illuminated in a steady state if the door is in motion.

As before, ensure that you have made effective use of both instruction and rung comments to clearly document your program.

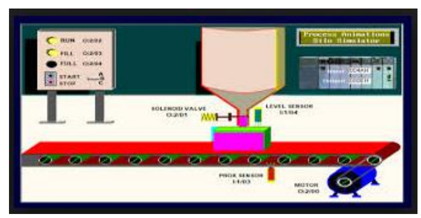
Student Programming Exercise #4:

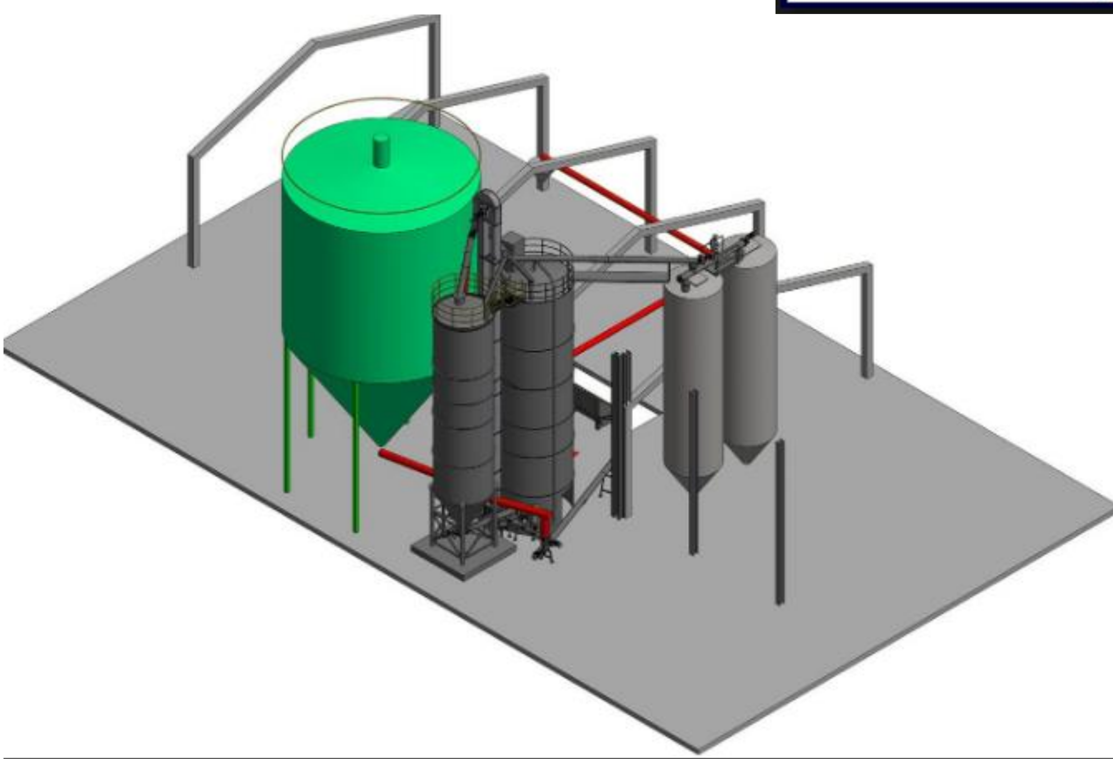
In this exercise we want you to modify your program so that it adheres to this additional criteria:

- If the door is currently opening, pressing the Close Switch will immediately halt movement. Door movement will remain halted when the switch is released.
- If the door is currently closing, pressing the Open Switch will immediately halt movement. Door movement will remain halted when the switch is released.
- Once movement is halted by the either of the foregoing actions, the operating criteria associated with the previous exercise will again take effect.
- The utilization of Binary or Integer Table bits to Flag specific conditions within your program would be appropriate. Also, the retentive OTL and OTU instructions may be utilized freely at your discretion

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control   Lab (0908462)

Silo System Simulator





Objective

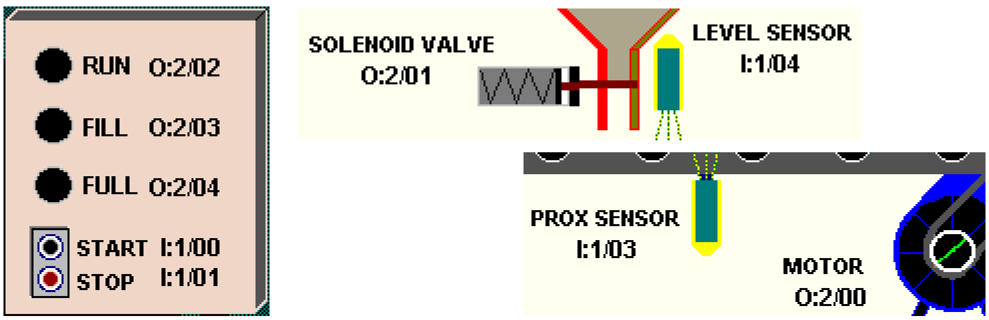
To be familiar with basic relay logic.

Pre-lab Preparation:

1- Read Chapter 3

Procedure:

From the Simulations Menu at the top of the screen, Select the Silo Simulation



Exercise #1 -- Continuous Operation

Completely design and de-bug a ladder control circuit which will automatically position and fill the boxes which are continuously sequenced along the conveyor. Ensure that the following details are also met:

- The sequence can be stopped and re-started at any time using the panel mounted Stop and Start switches.
- The RUN light will remain energized as long as the system is operating automatically.
- The RUN light, Conveyor Motor and Solenoid will de-energize whenever the system is halted via the STOP switch.
- The FILL light will be energized while the box is filling.
- The FULL light will energize when the box is full and will remain that way until the box has moved clear of the prox-sensor.

Exercise #2 -- Container Filling with Manual Restart

Alter or re-write your program so that it incorporates the following changes:

- Stop the conveyor when the right edge of the box is first sensed by the prox-sensor.

- With the box in position and the conveyor stopped, open the solenoid valve and allow the box to fill. Filling should stop when the Level sensor goes true.

- The FILL light will be energized while the box is filling.

- The FULL light will energize when the box is full and will remain that way until the box has moved clear of the prox-sensor.

- Once the box is full, momentarily pressing the Start Switch will move the box off the conveyor and bring a new box into position. Forcing the operator to hold the Start button down until the box clears the prox-sensor is not acceptable.

Exercise #3 -- Selectable Mode of Operation

Alter or re-write your program so that the panel mounted Selector switch can be utilized to select one of 3 different modes of operation. The 3 modes shall operate as follows:

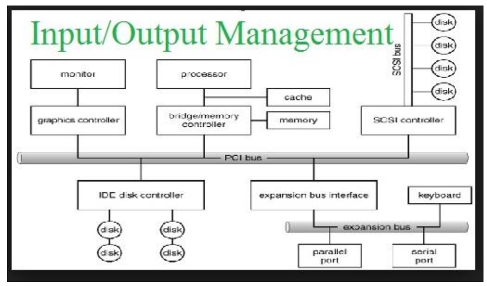
- When the selector switch is in position "A", the system shall operate in the "Continuous" mode of operation. This is the mode of operation which was used in Exercise #1.

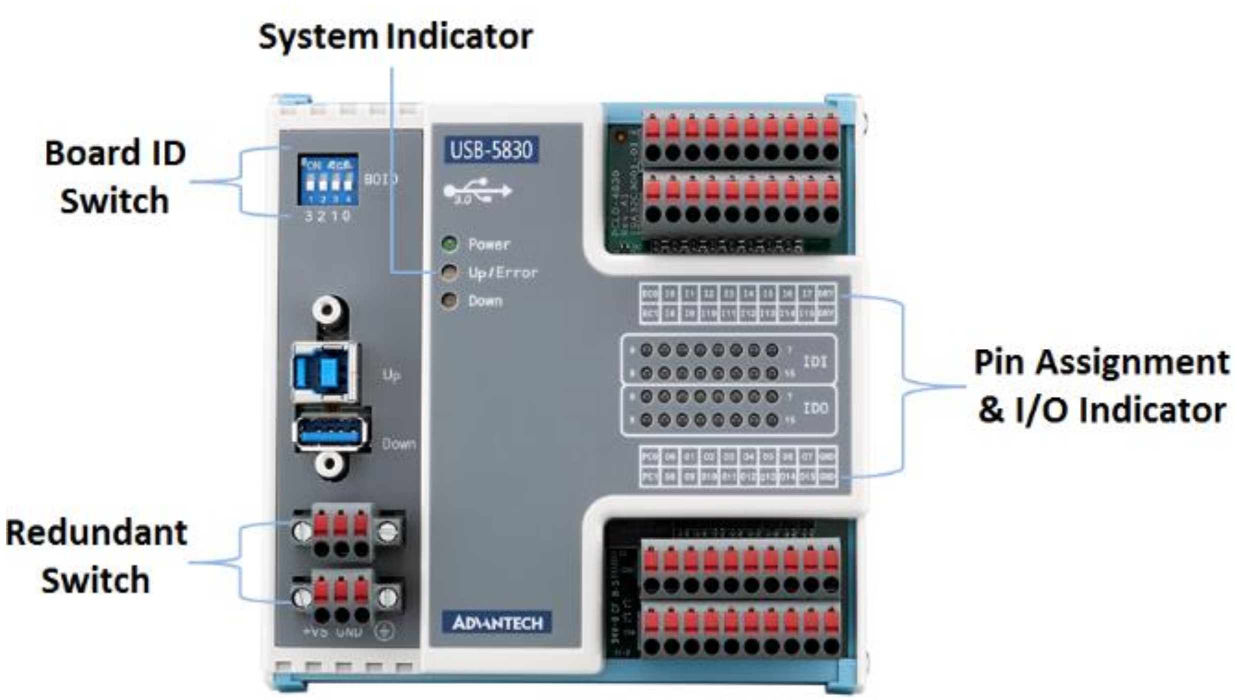
- When the selector switch is in position "B", the system shall operate in the "Manual Restart" mode of operation. This is the mode of operation which was used in Exercise #2.

- When the selector switch is in position "C", the system shall operate in the "Fill Bypass" mode of operation. In this mode, the boxes will simply move down the conveyor continuously and bypass the fill operation. As in the other modes, the Start and Stop pushbuttons will control the conveyor motion and the Run Lamp will operate as expected.

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

I/O Assignment





A set of guidelines should be followed during PLC program organization and implementation in order to develop an organized system. Approach guidelines apply to two major types of projects:

- New applications M
- Modernizations of existing equipment.

Flow charting can be used to plan a program after a written description has been developed. A flowchart is a pictorial representation of the process that records, analyzes, and communicates information, as well as defines the sequence of the process.

Two important documents that provide information about the arrangement of the PLC system are the I/O assignment table and the internal address assignment table.

1. The I/O assignment table documents the names, locations, and descriptions of the real inputs and outputs.
2. The internal address assignment table records the locations and descriptions of internal outputs, registers, timers, counters, and MCRs.

Figure 1 shows the proper wiring with addresses in PLC.
The I/O assignment table documents the names, locations, and descriptions of the real inputs and outputs.

An example of IO Address Assignment Table:

Input Listing	Address	Output Listing	Address
Inductive Sensor	X0	Pilot Light	Y0
Reed Sensor	X1	Small DC Motor	Y1
Capacitive Sensor	X2	Solenoid Valve	Y3
Pushbutton	X3		

Objective

To be familiar with the I/O assignment table preparation basics.

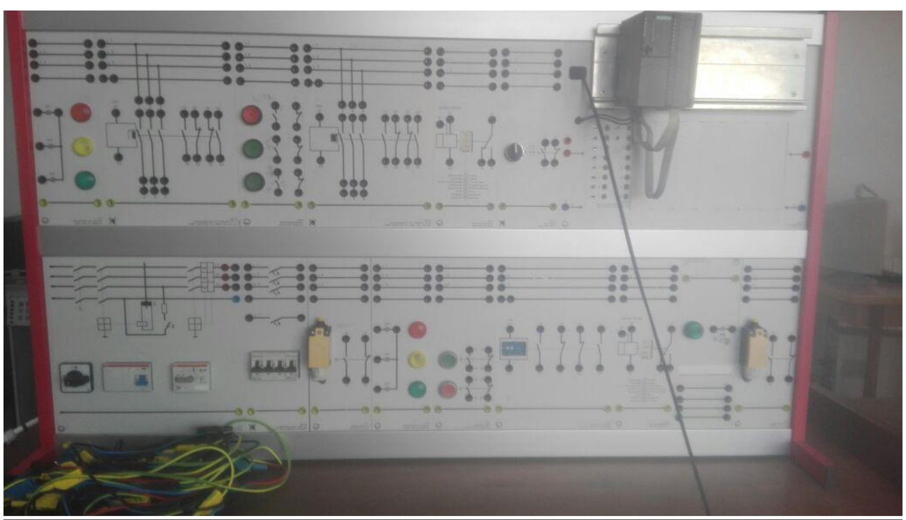
Pre-lab Preparation:

Read Chapter 2+3+4

Procedure:

This lab experiment is composed of four parts.

Part 1(Motor Control Establishing a new system)

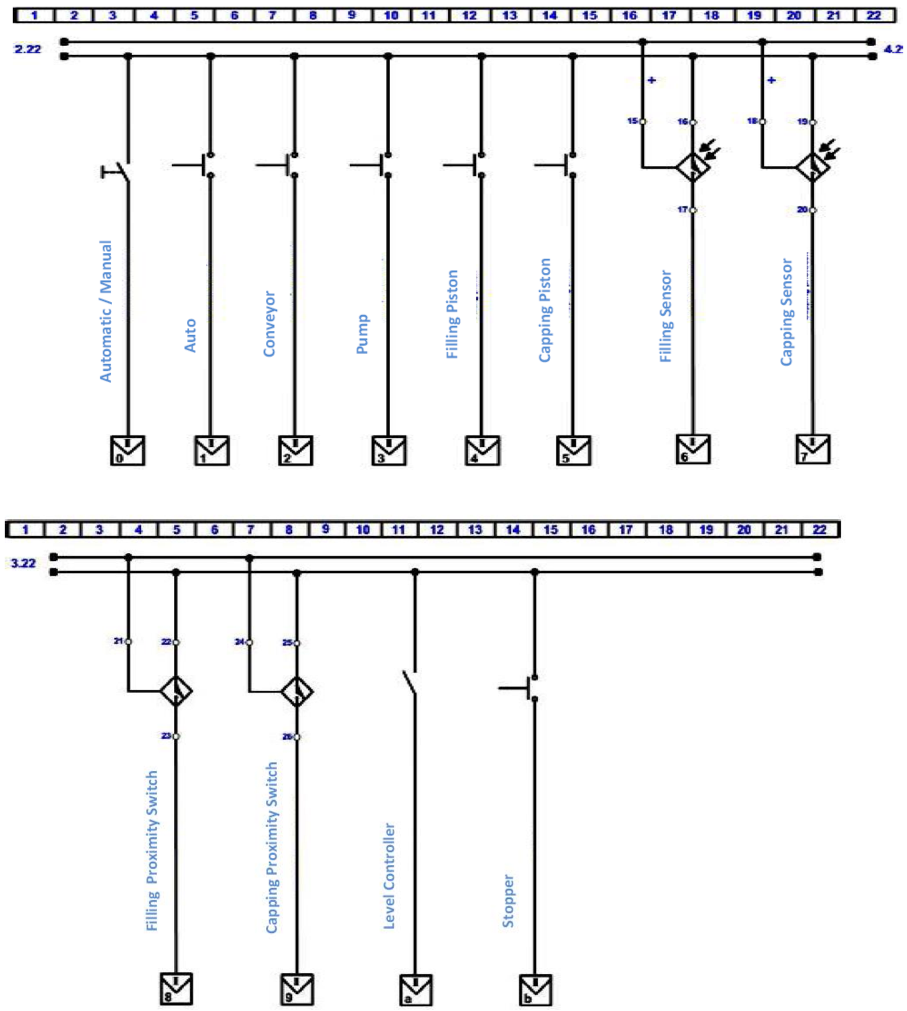


Exercise 1

Fill the following I/O assignment table:

Part 2 (Filling Machine Reading Wiring Diagram)

PLC Inputs Wiring Diagram



Exercise 2:

Fill the following I/O assignment table:

Part 3(Elevator Software Assistance)

Exercise 3:

Fill the following I/O assignment table:

Part 4 (Traffic Light Multimeter)

Exercise 4

Fill the following I/O assignment table:

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control   Lab (0908462)

Traffic Light control simulator

Objective

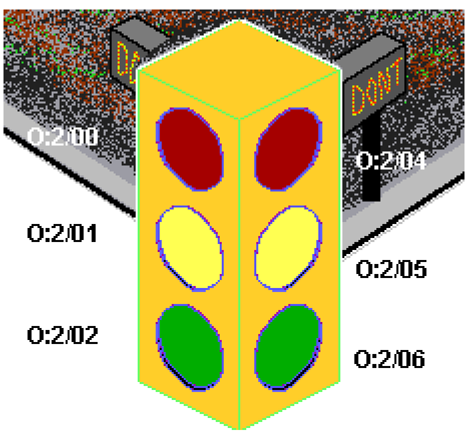
To be familiar with basic timer logic.

Pre-lab Preparation:

Read Chapter 4

Procedure:

From the Simulations Menu at the top of the screen, select the Traffic Light Simulation



Exercise #1 -- Traffic Control using 3 Lights

Using your knowledge of cascading timers, develop a ladder logic program which will sequence a set of green, amber and red lights in the following manner:

Sequence of Operation:

1. Light O:2/00 (Red) = 12 seconds ON
2. Light O:2/02 (Green) = 8 seconds ON

3. Light O:2/01 (Amber) = 4 seconds ON
4. The sequence now repeats with Red = ON.

RED	GREEN	AMBER
12 Sec.	8 Sec.	4 Sec.

<------------------------------- Time in Seconds ----------------------------->

Exercise #2 -- Traffic Control using 6 Lights

Modify your program so that the 3 lights which represent the other traffic direction are also controlled. It is tempting to use six timers for this task, but the job can be done with just four, and you'll end up with a much cleaner program as a bonus.

Red = O:2/00		Green = O:2/02	Amber = O:2/01
Green = O:2/06	Amber = O:2/05	Red = O:2/04	
8 Sec.	4 Sec.	8 Sec.	4 Sec.

<------------------------------- Time in Seconds ----------------------------->

Still getting the odd Crash? Well it's pretty obvious that these drivers aren't paying much attention to Amber Lights! No need for any more wiring however. You can solve this problem, but it's going take a little more programming.

Exercise #3 -- Traffic Light With Delayed Green

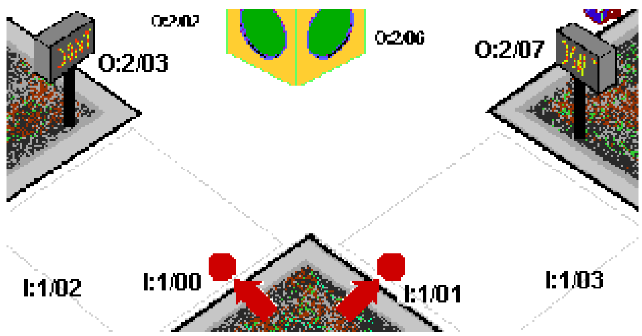
Modify your program so that there is a 1 second period when both directions will have their RED lights illuminated. Note that there are actually 2 such overlaps that need to be accounted for. The timing diagram below shows the six discrete timing intervals required to accomplish the desired sequence of operation, and with proper cascading you should be able to come up with an easy to follow solution using just 6 timers.

Red = O:2/00			Green = O:2/02	Amber = O:2/01	R
Green = O:2/06	Amber = O:2/05	Red = O:2/04			
8 sec.	4 sec.	1s	8 sec.	4 sec.	1s

<-------------------------------- Time in Seconds ------------------------------>

Exercise #4 -- Dealing with Pedestrians

Modify your program so that the crosswalks are also controlled. This is not necessarily a word comparison exercise, but it's a task that needs doing, and you should now have sufficient expertise to accomplish it. It might not be all that easy however!



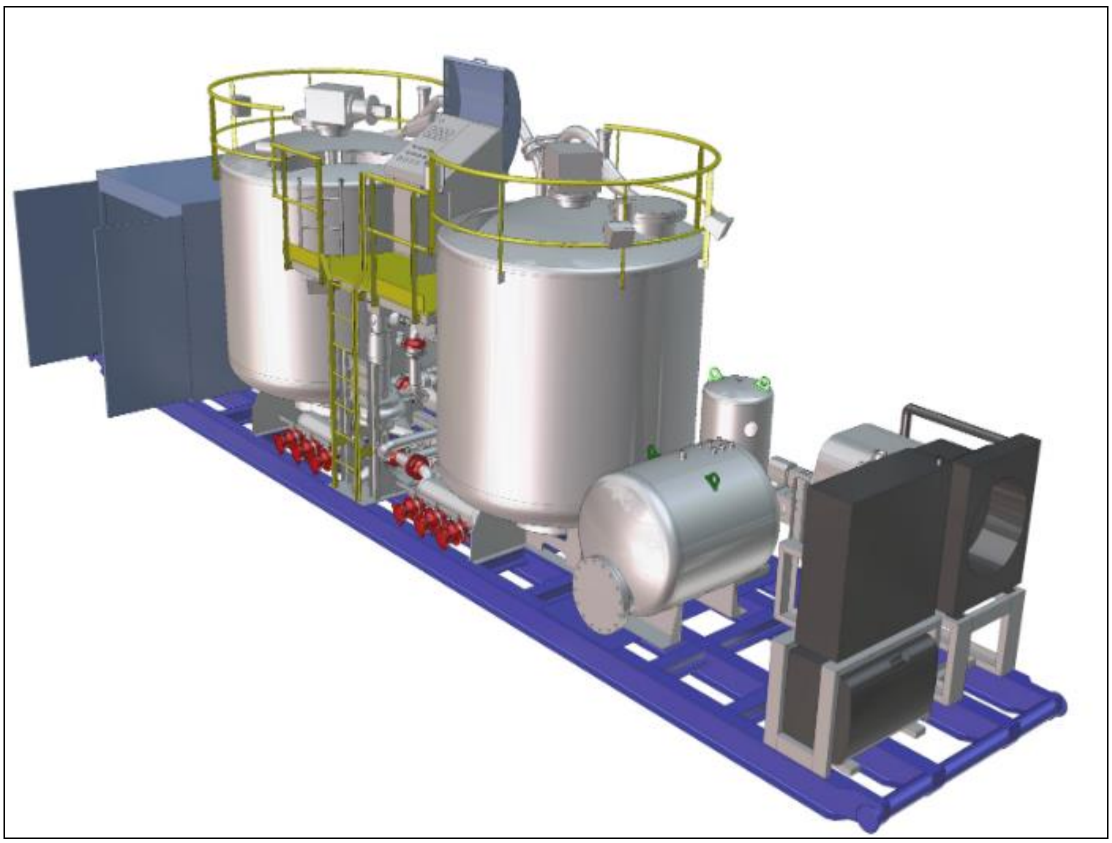
Your program should operate as follows:

- When depressed, the crosswalk pushbutton will cause the appropriate Walk Sign to be illuminated at the next occurrence of a Red to Green transition for the appropriate direction.
- If the Green light is already illuminated when the button is pressed then the Walk signal sequence will be delayed until the next Red to Green transition occurs.
- Once the Walk Sign is illuminated, it will remain so for the duration of the Green signal.
- When the Amber light appears, the Walk Sign will commence to flash On and Off and continue to do so until the Red signal appears.

You might consider using a bit from the free-running timer located in S2:4 to create the cautionary flashing effect.

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

Batch Mixing Simulator



Objective

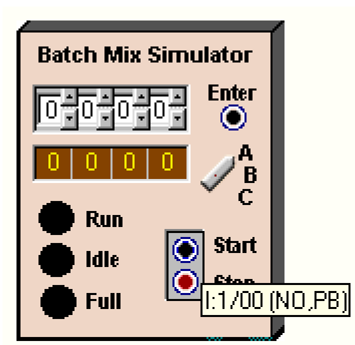
To be familiar with basic counter logic.

Pre-lab Preparation:

Read Chapter 4

Procedure:

From the Simulations Menu at the top of the screen, Select the Batch Mixing Simulation.



Exercise #1 -- Filling the Batch Mixing Tank

Using your knowledge of PLC counters, design a program to meet the following requirements:

- When the Start switch (I:1/0) is pressed, pump P1 will be energized and the tank will start to fill. The pulses generated by Flowmeter 1 should be used to increment a counter.

- When the count reaches a value where the tank is approximately 90% full, the pump is to be shut-off and and the control panels FULL light is to be energized.

- The filling operation is to halt immediately if the stop switch is pressed.

- While testing, utilize the "Reset Simulation" and the "Reset Timers and Counters" entries in the Simulations menu to re-start your program.

Exercise #2 -- Emptying the Batch Mix Tank

Modify your program so that it meets the following additional requirements:

- The mixer will run for 8 seconds once the tank is full.

- When the mixing is complete, drain pump P3 is to be started and the tank is to be drained. Flowmeter 3 will be employed to decrement the existing counter, and draining will be allowed to continue till the counters accumulator reaches zero.

- Once the tank is empty again, pressing the Start switch will cause the sequence to repeat.

Exercise #3 -- Continuous Operation

Modify your program so that the filling and emptying sequence will repeat continuously once it has been started by the initial pressing of the Start switch.

- Ensure that the RUN light is energized when the mixer or either pump is running.
- The STANDBY light should light and the process should halt when the Stop button is pressed.
- The process should restart where it left off if the the Start button is pressed following a Stop.

Exercise #4 -- Single Batch Mode of Operation

Using your knowledge of PLCs, design a program to meet the following criteria:

- **When the 3 position Selector Switch is in position "A".** The batch mixing process will run in a single batch mode. The operator may start the batch mix sequence by momentarily pressing the Start Switch.
- Once a batch sequence has begun, the sequence may be stopped and resumed at any time using the Stop and Start switches.

- The tank is to be filled with a mixture obtained from the separate fill lines utilizing fill pumps P1 and P2. A counter will track the quantity of product obtained from Line 1 (P1) while the remainder will come from Line 2 (P2). The mixture ratio of the product will be controllable by adjusting the counter's preset. The tank is to be filled to the point where the Hi-Level sensor goes true.
- When filling is complete, the Full light will turn On. Heater O:2/04 and Mixer O:2/00 will be started allowing the mixture to begin heating. Thermostat I:1/02 will be employed to control the temperature.
- The mixer will continue to run for 4 seconds after the mixture reaches the desired temperature. When the mixer stops, pump P3 will be used to drain the product from the tank. The tank will be drained to the point where the Lo-Level sensor trips.
- Once emptied, a new single batch sequence may again be started by pressing the Start Switch while the Selector switch is in position "A".

Exercise #5 -- Multiple Batch Mode of Operation

Enhance your program to include the following features:

- **When the 3 position Selector Switch is in position "B".** The process will produce multiple batches, the number of batches produced will be operator selectable, and the current batch count will be displayed on the Control panel's LED display.
- The operator will be able to enter the desired batch count using the control panel thumbwheel switches.
- The operator will be able to set the product ratio by entering the desired product one percentage via the control panel thumbwheel switches.

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

Practical lab 2&3

Objective

To be familiar with programming different PLC types.

Procedure:

This lab experiment is composed of four parts, each group need to finish two parts in the first week, then the rest two in the next week according to the schedule submitted during the previous lab.

Part 1

Use the following training panel, which is exist in the lab, to complete the Automatic Garage Door Experiment:

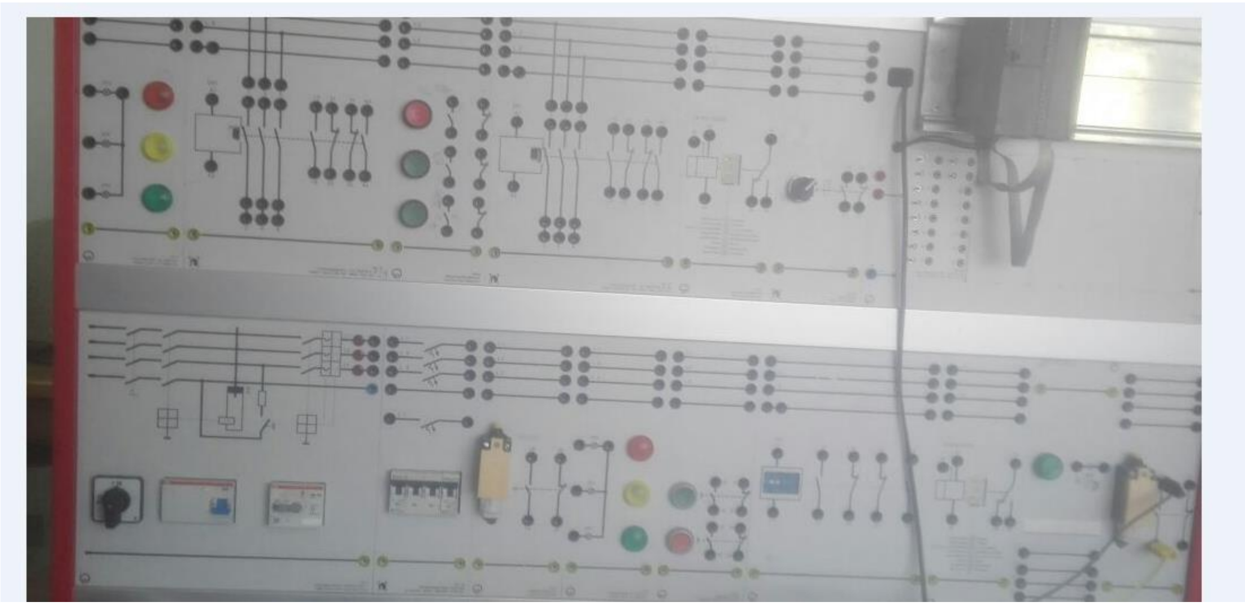


Figure 1 Training Board

Pre-lab Preparation:

- Review basic PLC instructions (Chapter 3, 4).
- Read Siemens datasheets

I/O Allocation list:

Input	Address	Output	Address
Start PB	I124.0	Motor CW	Q124.0
Stop PB	I124.1	Motor CCW	Q124.1
Sensor to indicate the car arrival (implemented by pb1).	I124.2	Stop Lamp	Q124.2
Sensor to indicate the car	I124.3	Lamp for motor CW	Q124.3

passed (implemented by pb2).			
Limit Switch1	I124.4	Lamp for motor CCW	Q124.4
Limit Switch2	I124.5		

Exercise 1:

Wire the input & output, which include the single phase motor representing the Garage Door, use S7-300 Siemens PLC as shown below.

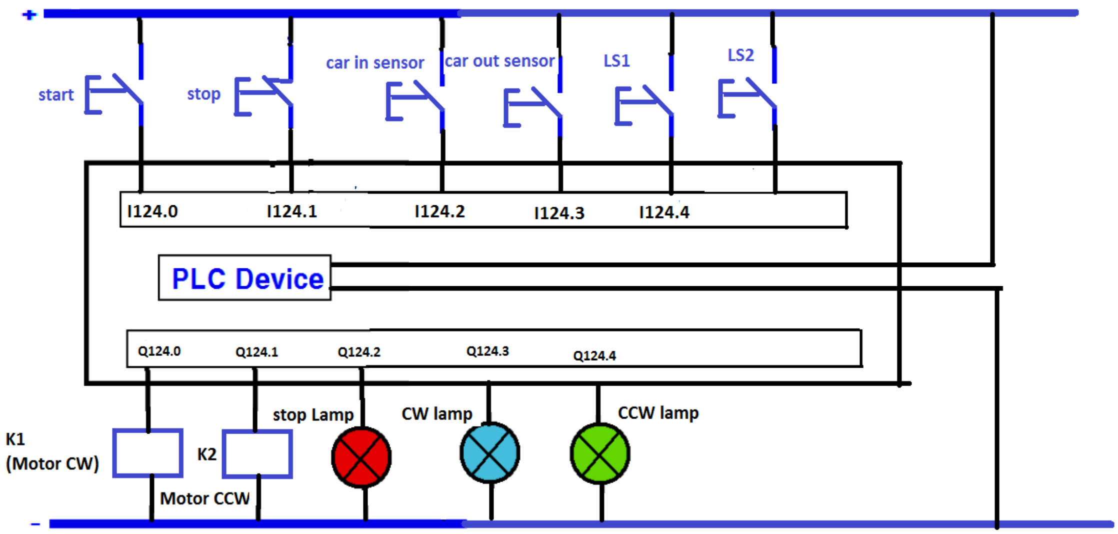


Figure 2 PLC Wiring Diagram

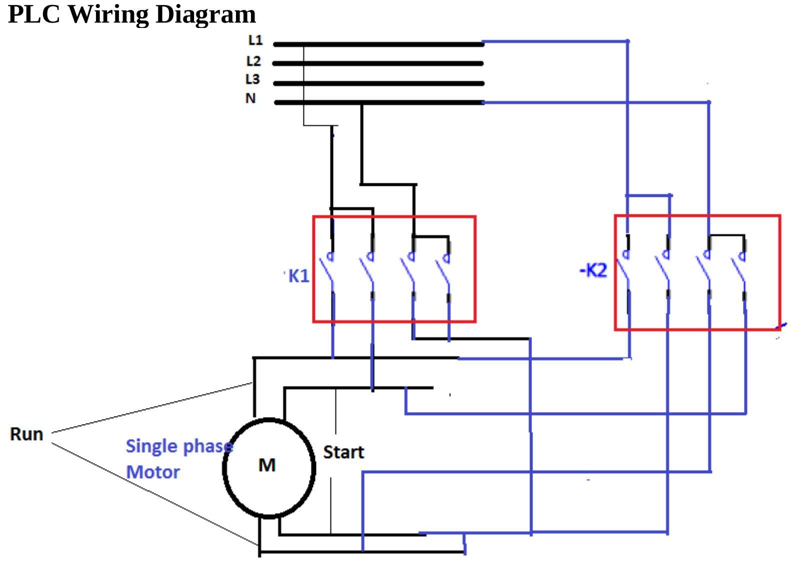


Figure 3 Power Wiring Diagram

Exercise 2:

Write a ladder logic program to satisfy the following:

1- When the sensor detect a car (implemented by reach PB on control panel), a single phase motor rotates in a clock wise direction and stop when press on a limit switch1 to indicate the door is fully opened.

2- When pass pushbutton is pressed which is indicate the car passed, the single phase motor rotate in the opposite direction a stop when press on limit switch 2 to indicate the door is fully closed.
3- Use indication lamps for the movement of the motor one for CW and other for CCW.
4- There is an emergency stop PB to stop the system any time & turned on an emergency lamp.

Part 2

Use the following Elevator Model, which is exist in the lab, to complete the following experiment:

Figure 4 Elevator Model

Pre-lab Preparation:

- Review basic PLC instructions (Chapter 3, 4).
- Read Zelio datasheets

I/O Allocation list:

Input	Address	Output	Address
External PB. For Gf	I1	Lamp for GF	Q1
External PB. For 1st floor	I2	Lamp for 1st Floor	Q2
External PB. For 2nd floor	I3	Lamp for 2nd Floor	Q3
External PB. For 3rd floor	I4	Lamp for 3rd Floor	Q4
Limit switch for Gf	I5	Motor Up	Q5
Limit switch. For 1st floor	I6	Motor Down	Q6
Limit switch. For 2nd floor	I7		
Limit switch. For 3rd floor	I8		
Internal PB. For Gf	I9		
Internal PB. For 1st floor	IA		
Internal PB. For 2nd floor	IB		
Internal PB. For 3rd floor	IC		

Exercise 1:

Write a ladder logic program to request any floor from Internal & external Pushbuttons, each lamp for each floor turned on when the elevator reach the corresponding floor.

Exercise 2:

Update your program to add priority for the nearest floor even if it was requested later.

Part 3

Use the following Traffic Light Model, which is exist in the lab, to complete the following experiment:

Figure 5 Traffic Light Model

Pre-lab Preparation:

- Review basic PLC instructions (Chapter 3, 4).
- Read Mitsubishi datasheets

I/O Allocation list:

Input	Address	Output	Address
Push button for pedestrian	X0	Red (Traffic)	Y0
		Yellow (Traffic)	Y1
		Green (Traffic)	Y2
		Red (pedestrian)	Y3
		Green (pedestrian)	Y4

Exercise 1:

Write a ladder logic diagram to run the Traffic in the following sequence:
Red: 15 sec.
Green: 10 sec.
Yellow: 5sec.
The pedestrian traffic always Red except the pushbutton of pedestrian pressed the pedestrian traffic became Green for 40 sec on condition the Car traffic is Red.

Exercise 2:

Update your Program to meet the following:
If there is 5 Pedestrian & the traffic light for cars green or yellow, the pedestrian traffic became green & the car traffic became red for 40 sec. to allow the pedestrian to cross the road, then return to its sequence

Part 4

Use the following Water Filling Machine, which is exist in the lab, to complete the following experiment:

Figure 6 Water Filling Machine

Pre-lab Preparation:

- Review basic PLC instructions (Chapter 3, 4).
- Read Omron datasheets
-

I/O Allocation list:

Input	Address	Output	Address
Automatic/Manual	I0	Conveyor motor	Q0
Auto	I1	Pump	Q1
Conveyor	I2	Filling piston solenoid valve	Q2
Pump	I3	Capping piston solenoid valve	Q3
Filling Piston Sensor	I4	Stopper piston solenoid valve	Q4
Capping piston Sensor	I5	Automatic Start Lamp	Q5
Filling sensor	I6		
Capping sensor	I7		
Filling proximity sensor	I8		
Capping proximity sensor	I9		
Level controller	IA		
Stopper	IB		

Exercise 1:

Write a Program to control the machine manually using push buttons, as the following:

- Press on Conveyor PB to run the Conveyor motor.
- Press on Stopper PB to extend the Stopper piston solenoid valve.
- Press on Filling PB to extend the Filling piston solenoid valve.
- Press on Capping PB to extend the Capping piston solenoid valve.
- Press on Pump PB to run the Pump.

  When release your hand from any PB. The corresponding output turn off.

Exercise 2:

Now update your program to control the machine in automatic mode.

When the selector switch in in Auto position:

- Press on Auto PB to start the Conveyor movement.
- Stop the Conveyor when the Filling proximity sensor detects a bottle.
- Extend the Stopper pistons.
- Extend the Filling piston.
- Turn on the Pump for 20 sec.
- Retract the Filling piston.
- Retract the Stopper pistons.
- Start the Conveyor
- Stop the Conveyor when the Capping proximity sensor detects a bottle
- Extend the Stopper pistons.
- Extend the Capping piston for 10 sec.
- Retract the Capping piston.
- Retract the Stopper pistons.
- Then repeat...

Stopper

Water Pump

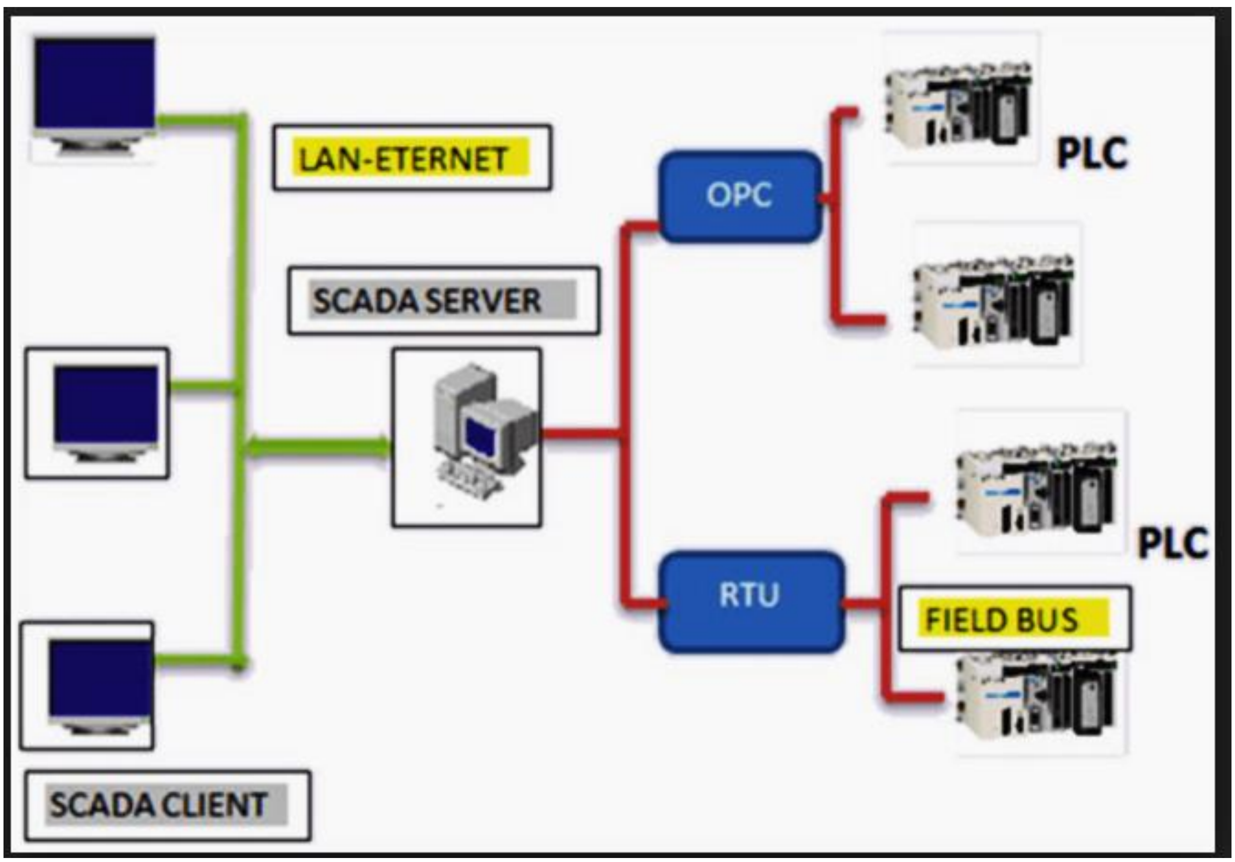
Water Filling Unit

Cap Pressing Unit

Figure 7 Different Filling Machine Parts

University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

Supervisory Control And Data Acquisition System



Objective

In this lab, you are going to build a SCADA system using LabView and other OPC server Tools. You are expected to work in a group

Pre-lab Preparation:

Read Chapter 7

Procedure:

Exercise #1 -- Creating the OPC Server

You are going to build the OPC Server that is used to connect the FEC640 Festo mini-controller that is found in the Sorting Station. This OPC server will create a communication path between the mini-controller and the OPC Client application "LabView Program" you are going to build later on:

1- Open the OPC Editor found in the Start Menu.
2- Start creating the OPC server as follows:
3- Click **Next**
4- Write SCADA_GROUP as a name for your OPC Server, and click **Next**
5- Click "Save Project", and then click **Next**.
6- Click "Register Project", and then click **Next**
7- Select "Open project in editor", and then click **Finish**
8- Right click the **Namespace**, and select **Add Resource**
9- Put the Name as: **SORTINGSTATION**, the IP Address as: 192.168.1.22 and click OK.
10- Write click the resource **SORTINGSTATION** and select **Add resource**
11- Give it **INPUTS** as a name, and choose the **Input** button.
12- Repeat steps 10 and 11 to add a resource named **OUTPUTS** of type **output**
13- Right click **INPUTS** to add an input "**WP_AVAILABLE**".
14- Repeat steps 13 to add an output "**MOTOR**" in **OUTPUTS**

Exercise #2 -- Creating the OPC Client

Now, you are going to build the OPC Client that is used to connect the FEC640 Festo minicontroller that is found in the Sorting Station, through the OPC server you have just created, and that to test the work before proceeding further The mixer will run for 8 seconds once the tank is full.

1- Open the OPC Client found in the Start Menu.
2- Connect to the OPC server you have created.

3- Select SCADA_GROUP OPC Server to connect to.
4- Double click on WP_VAILABLE, and MOTOR tags to connect to them
5- The Quality Bad is shown besides each tag indicating that a communication error occurred, and since the communication path is via Ethernet card using the TCP/IP protocol you have to adjust the setting of the computer Broadband connection as follows.
6- Open the Control Panel > Network and Internet connections > Network connections, and then right click the Broadband connection to select properties
7- Select the TCP/IP protocol and choose properties.
8- Do the following changes: Static IP : 192.168.0.Your_PC_Number
9- Remove and add the tags once again and see the difference
10- Test the system and check the status of the tags

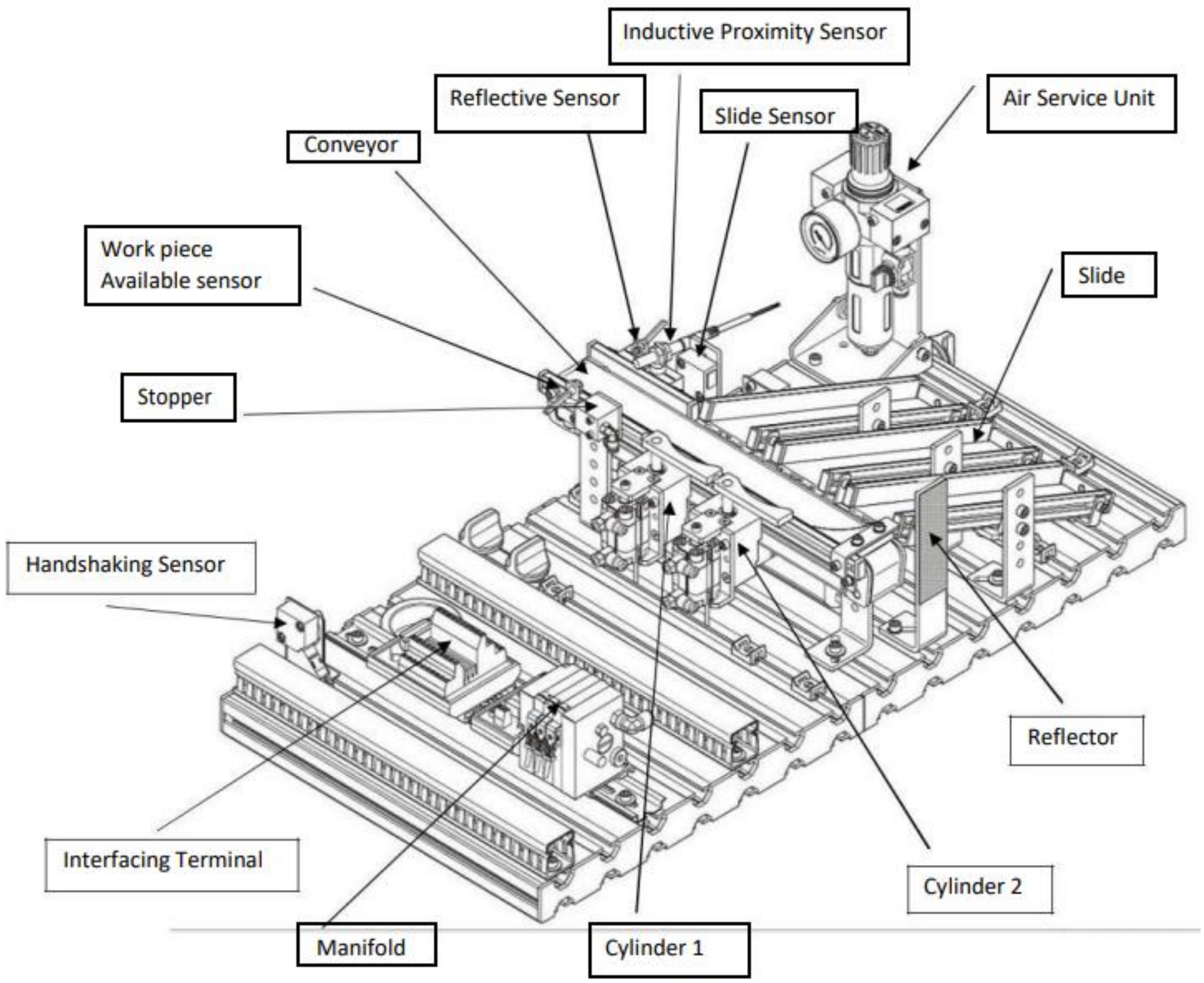
Exercise #3 -- build the OPC Client based on LabView (Explained by Engineer Only)

Now, you are going to build the OPC Client based on LabView that is used to connect the FEC640 Festo mini-controller that is found in the Sorting Station, through the OPC server you have created, and that to complete our SCADA system. See Appendix A for the allocation list of the PLC I/O.

1- Open the Sorting station VI found on the Desktop of the computer you are using, and right click the light indicator found on the conveyor picture.
2- Connect the **Conveyor** indicator with the corresponding tag previously created in the OPC server as follows:
3- Do the same for the **WP_Available** indicator and connect it to its corresponding tag.
4- Run the VI and test the system, and comment your results

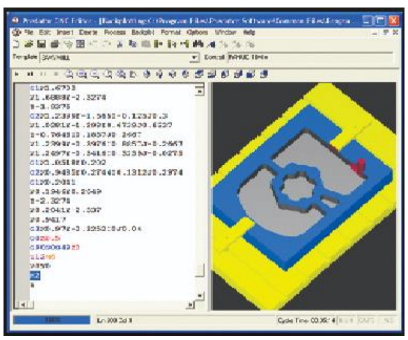
Appendix A PLC Allocation List:

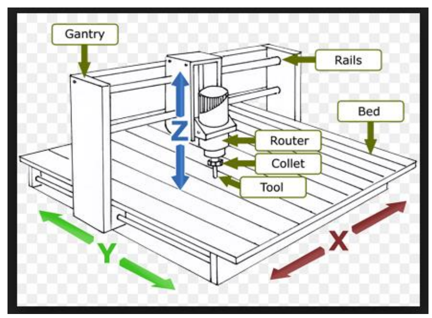
Input	***Description***	I0.5	Cylinder 1 Extended
I0.0	Work piece available	I0.6	Cylinder 2 Extended
I0.1	Metal	***Output***	***Description***
I0.2	Not black	O0.0	Conveyor belt
I0.3	Slide Sensor	O0.1	Cylinder 1
I1.0	Start Button	O0.2	Cylinder 2
I1.1	Stop Button	O0.3	Stopper
I1.5	Emergency stop	O0.6	Buzzer



University of Jordan
School of Engineering
Department of Mechatronics Engineering
Automation and Process Control Lab (0908462)

Computer Numeric Control Simulation (CNC)





Objective

In this lab, you are going to simulate a CNC system for milling machine. You are expected to work individually.

Pre-lab Preparation:

Read Chapter 8

Procedure:

Exercise #1 – Reading CNC Codes (Absolute)

Draw the shape which is generate after execute the following code:

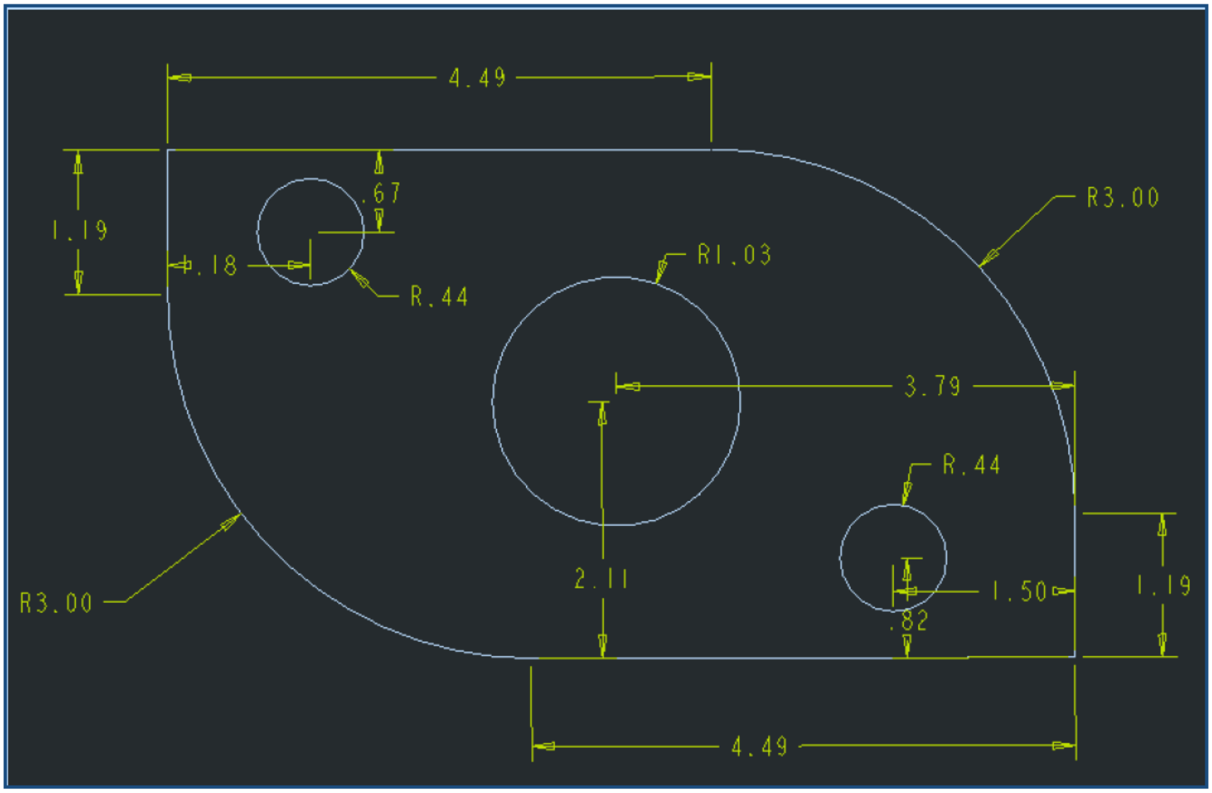
```
N1    g90  t1
N2    g00  z10
N3    g00  x60y10
N4    g00  z-30
N5    g01  x10y10
N6    g01  y130
N7    g01  x60
N8    g01  y120
N9    g01  x100
N10   g01  y110
N11   g01  x130
N12   g01  y95
N13   g01  x180
N14   g01  y45
N15   g01  x130
N16   g01  y30
N17   g01  x100
N18   g01  y20
N19   g01  x60
N20   g01  y10
N21   g01  z20 t14
N22   g00  x35y60
N23   g01  z-20
N24   g02  y80r10
N25   g02  y60r10
N26   g00  z10
N27   g01  x155y60
N28   g01  z-20
N29   g02  y80r10
N30   g02  y60r10
```

Exercise #2 -- Reading CNC Codes (Incremental)

Now, you are going to rewrite the code in **Exercise #1** using the CNC simulator. You must use Incremental addressing mode instead of absolute addressing mode.

Exercise #3 -- Writing CNC Codes (Incremental)

Write a code that will generate the following shape using Incremental addressing mode.



Exercise #4 -- Writing CNC Codes (Absolute)

Write a G-code to draw your name in Arabic, using the absolute addressing mode.

Appendix A CNC Word address commands:

The standard interpreter understands the following G- codes.

G0 or G00	Rapid movement
G1 or G01 or L	Linear interpolation
G2 or G02 or DR-	Clockwise interpolation

G3 or G03 or DR+	Anti-clockwise interpolation
G40	Turns off cutter compensation
G41	Cutter compensation to the left
G42	Cutter compensation to the right
G92	Zero point displacement
G87	Rectangular pocket milling
G81, G82, G83, G84	Drill cycle
G90	Absolute programming
G91	Incremental programming
G94	Feed rate in mm/min
G95	Feed rate in mm/revolution
G54	Zero point 1
G55	Zero point 2
G56	Zero point 3
G57	Zero point 4
G58	Zero point 5
G59	Zero point 6
G53	No zero point displacement

The standard interpreter understands the following M- codes.

M0 or M00	Program stop
M3 or M03	Spindle start clockwise
M4 or M04	Spindle start anti-clockwise
M5 or M05	Spindle stop
M8 or M08	Coolant on
M9 or M09	Coolant off
M17	Return from subprogram
M30 or M02 or M2	Program end

The standard interpreter understands the following codes.

F	Feed rate
S	Spindle speed
N	Block number
T or TOOL CALL	Tool

X	Code for the X-axis
Y	Code for the Y-axis
Z	Code for the Z-axis
I	Incremental distance to the center in the X-axis
J	Incremental distance to the center in the Y-axis
K	Incremental distance to the center in the Z-axis
R or Z	Arc <180 degrees
R- or Z-	Arc >180 degrees
P	Overlapping in pocket milling
L	Subprogram block number
U	Subprogram repetitions

($ SetPartSize(200,150,30,2)) : Determin the dimensions for the workpiece .

($Message("Press Esc to return")) : to write a message .