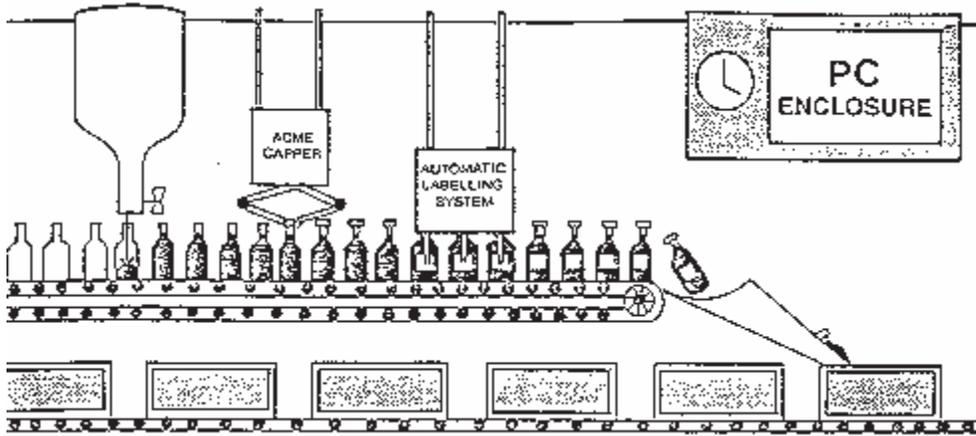


Unit 1, Overview of PLCs

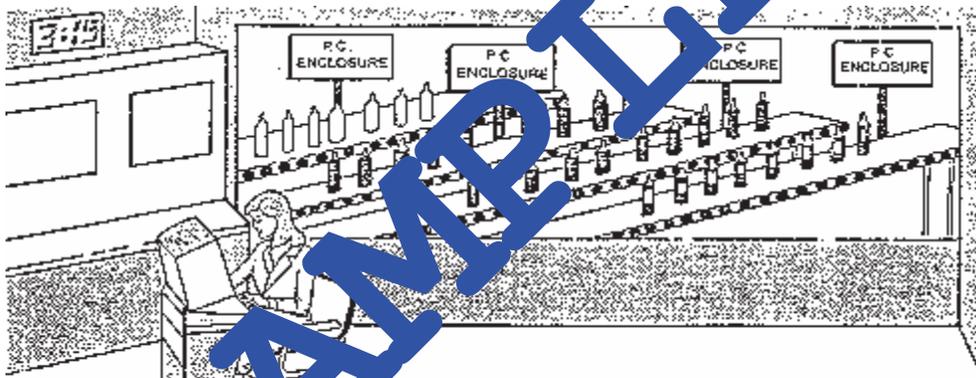
Learning Steps

Student Action	Where	Media
View	Unit 1, Introduction to PLCs	The PLCTrainer Tutorial
Complete	Unit 1, Introduction to PLCs	The PLCTrainer Lab Manual
Stop	Unit 1, Introduction to PLCs Assignment Sheet 1	Have Instructor Evaluate Assignment Sheet 1
Complete	Unit 1, Introduction to PLCs Assignment Sheet 2	The PLCTrainer Lab Manual
Stop	Unit 1, Introduction to PLCs Assignment Sheet 2	Have Instructor Evaluate Assignment Sheet 2
Complete	Unit 1, Practical Exercise 1	The PLCTrainer Lab Manual
Stop		Have Instructor Evaluate The PLCTrainer Lab Manual Practical Exercise 1
Complete	Unit 1, Practical Exercise 2	Have Instructor Evaluate The PLCTrainer Lab Manual Practical Exercise 2
Review	Unit 1, Introduction to PLCs	The PLCTrainer Tutorial
See Instructor		

SAMPLE



The high-speed demand and the computer age have together caused the evolution of a faster more precise process control system, the Programmable Logic Controller. This graphic shows the bottling process being more efficiently by a Programmable Logic Controller. The PLC offers greater flexibility and faster changes than the old relay control systems.



As technology continues to expand, so will our methods of automated control. Today it is possible to link many control systems together under the management of a central control system.

Functions of Programmable Controllers

PLCs make system control much more efficient than the electro-mechanical systems of the past. Additionally, PLC's are capable of performing many functions that the older systems just could not do. These might include:

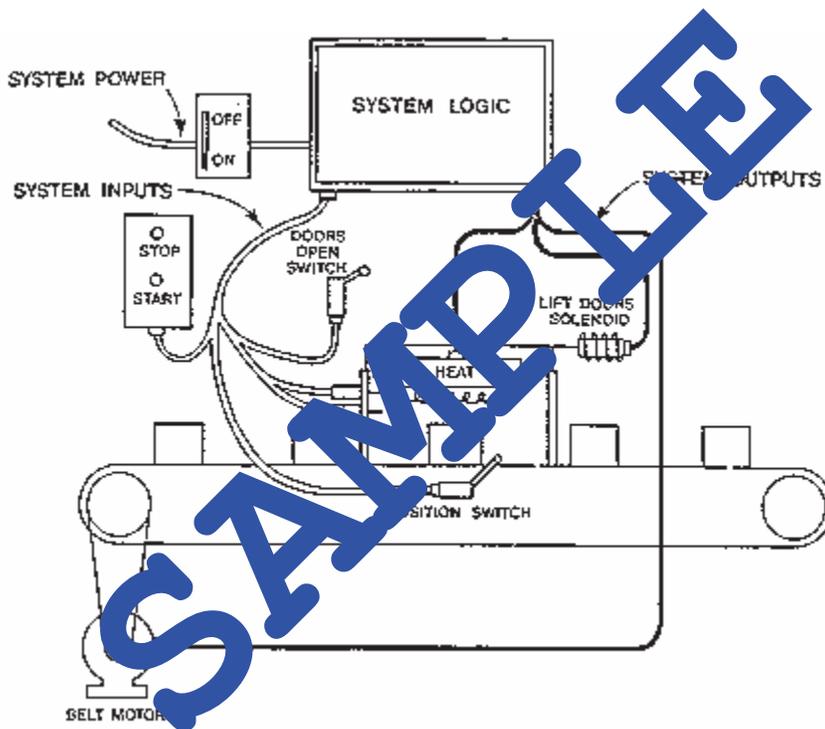
Practical Exercise 2

Directions:

Match the following input or output Devices with the statement that best explains their function in the system using your logical ability to determine what device causes what action. You will not find answers to this question within the tutorial.

When the system is started, the product moves into the 'oven'. There it stops and the doors will close. Once the target temperature is achieved for a specific period, the doors will open and the product moves out as another moves in place.

- | | |
|------------------------|--------------------|
| A. Temperature Sensors | B. Position Switch |
| C. Doors open switch | D. Start button |
| E. Heating Element | F. Drive Motor |



1. The _____ De-activates, signaling the Logic to begin heating.
2. The _____ will signal the Logic to start the belt moving.
3. The _____ will determine whether the heat is at the proper level.

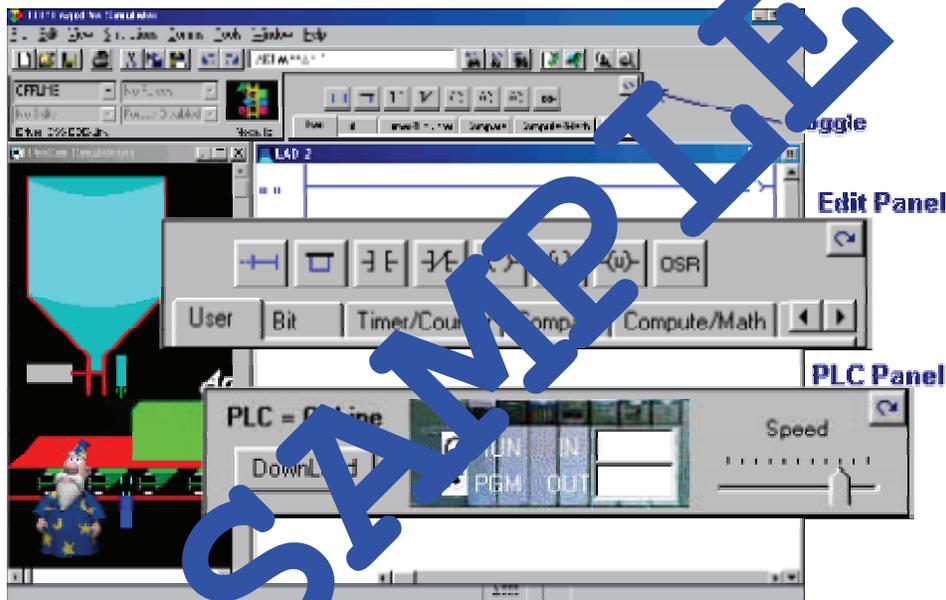
If you are New To PLCs then Please Read This.!

LogixPro allows you to practice and develop your RSLogix programming skills where and when you want. It replaces the PLC, ladder rung editor, and all the electrical components that have until now, been required to learn RSLogix. It doesn't however; replace instructors, texts, tutorials or PLC documentation manuals etc. which are so essential when learning about PLCs and RSLogix. Think of LogixPro as a Word Processor which will allow you to practice your literary skills **after** you are familiar with the meaning of words and how they are spelled

The LogixPro Screen

The most commonly used elements of LogixPro are displayed below. The **Edit Panel** provides easy access to all the RSLogix instructions and they may be simply dragged and dropped into your program.

Once your program is ready for testing, clicking on the "Toggle Button" of the Edit Panel will bring the **PLC Panel** into view. From the PLC Panel you can download your program to the "PLC" and then place it into the "RUN" mode. This will initiate the scanning of your program and the I/O of your chosen simulation.

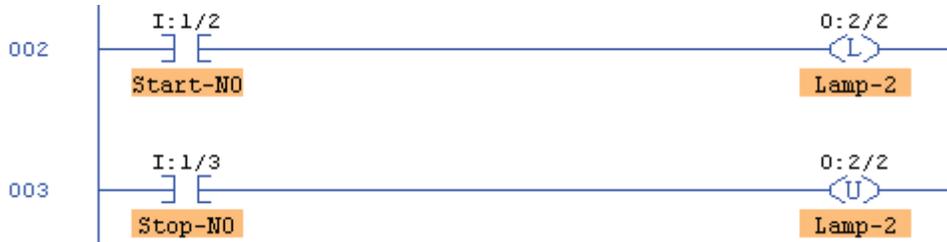


Editing Your Program

If you're familiar with Windows and how to use a mouse, then you are going to find editing a breeze. Both Instructions and Rungs are selected simply by clicking on them with the left mouse button. Deleting is then just a matter of hitting the Del key on your keyboard.

Double Clicking (2 quick clicks) with the left mouse button allows you to edit an instruction's address while right clicking (right mouse button) displays a pop-up menu of related editing commands.

Click on an Instruction or Rung with the left mouse button and keep it held down and you will be able to drag it wherever you please. Let go of it on any of the tiny locating boxes that you will see, and the Instruction or Rung will cling to its new home. Isn't Windows Grand!

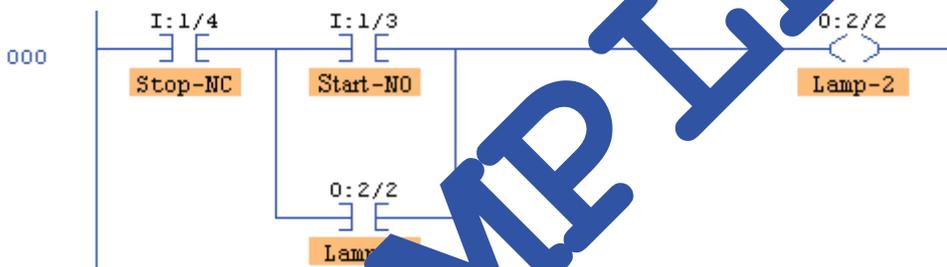


Activate the Start and Stop switches and ensure that the OTL and OTU output instructions are responding as outlined in your text. Once you have the lamp ON, could you turn it off if power was lost in the Stop Switch circuit?

Now modify your program so that it operates correctly when you substitute the N.O. Stop switch (I:1/03) with a Normally Closed Switch. If we now lost power on the N.C. Stop switch circuit, what would happen to the state of Lamp (O:2/02)?

Emulating Standard Stop/Start Control

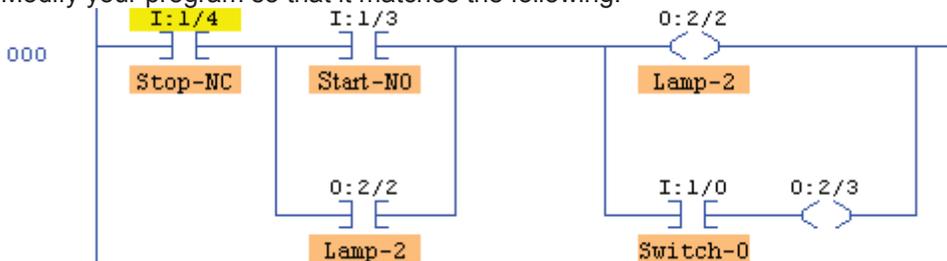
Erase your program by selecting "New" from the "File" menu selection at the top of the screen. When the dialog box appears just click on "OK" to select the default PLC type. Now enter the following program. To enter a branch, just drag the branch (but not coil) to the rung and then insert or drag instructions into the branch.



Before you download and run this program, take a careful look at our use of a XIC instruction to test the state of the N.C. Stop Switch. When someone presses the Stop Switch, will bit I:1/04 go True or False? Will the XIC instruction go True or False when the Switch is pressed? Is this the logic we are seeking in this circuit? Run the program and see if you're right! If we loose power in the Stop Switch circuit, what state will the lamp go to? Why do you think that most prefer this method rather than the OTL/OTU method of implementing Stop/Start Control?

Output Branching with RSLogix

Modify your program so that it matches the following.



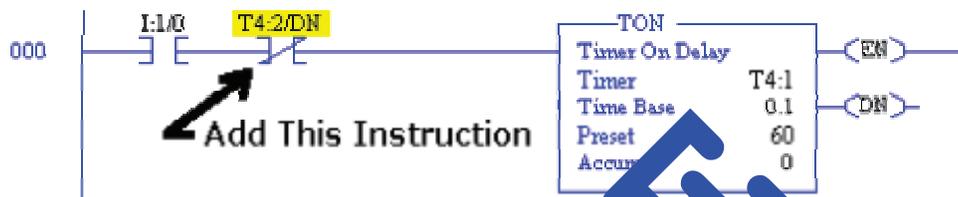
Download and Run the program. Operate the Stop and Start switches several times with Switch-0 open, and again with Switch-0 closed. Remove the XIC instruction from the Output branch and

- Run the timers through their timing sequence a number of times. Don't move on until you are satisfied that the timers are working as you would expect

In this exercise we have utilized just two timers, but there is nothing stopping us from sequencing as many timers as we wish. The only thing to remember is; to use the DN (done) bit of the previous timer to enable the next timer in the sequence. Obviously locating the timers on consecutive rungs, and employing consecutive numbering will make such a program much easier to read and trouble-shoot.

Self Resetting Timers

- Place the PLC into the PGM mode, and modify the first rung of your program as depicted below.



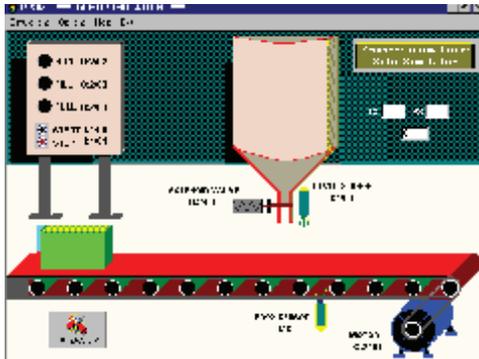
- Once you have modified your program, download it to the PLC and place the PLC into the RUN mode.
- Close switch I:1/0 and observe the operation of the timers. The timers should now be operating in a continuous loop with T4:1 starting timer2, and then when Timer2 is done, Timer1 is reset by Timer2's done bit. As before, when Timer1 is reset, it in turn resets Timer2 which causes Timer2's done bit to go low (T4:2/DN=0). Once Timer2's done bit is low, the sequence is back where it originally began, and the timing sequence will start over once again on the very next scan.
- Remove the first instruction (switch XIC I:1/0) from rung zero of your program.
- Download and Run the modified version of your program
- Does the timing operation continuously sequence as before? It should!
- Can you stop the timing sequence? Not without taking the PLC out of the RUN mode! In many applications there may never be a need to stop such a timing sequence, so a switch might not be used or needed.

In this exercise we cascaded two timers, but as before there is nothing to stop us from cascading as many timers as we wish. The thing to remember here is; utilize the DN (XIC or "NOT" done) bit of the last timer in the sequence to reset the first timer in the sequence. Once again, consecutive rungs, and numbering will make a program much easier to read and trouble-shoot.

The TOF Timer (Timer OFF Delay)

In Allen Bradley PLC programming, the TON timer is by far the most commonly used type of timer. Most people consider TON timers to be simple to use and understand. In comparison, many people find the operation of the Allen Bradley TOF (Timer OFF delay) timer to be less intuitive, but I'm going to let you decide for yourself.

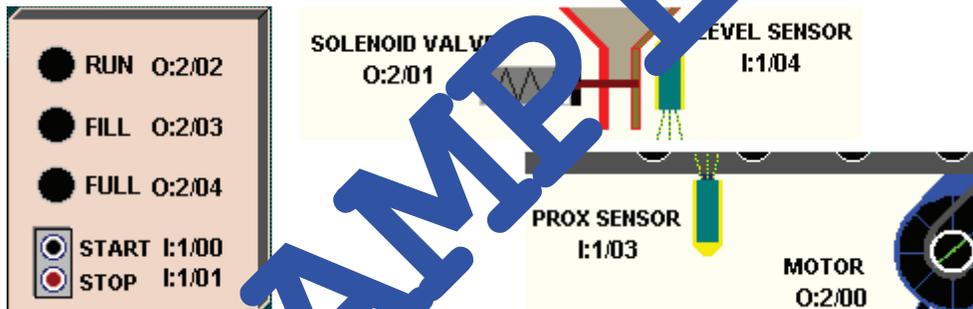
Scenario 1:



LogixPro
The Silo Lab
Utilizing Relay Logic

The ProSim-II Silo Simulation

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



Exercise #1 -- Continuous Operation

Completely design and debug 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.

Assignment Sheet 1:

- The terms table or file are used when referring to:
 - (A) Consecutive groups of words
 - (B) Two words
 - (C) A large metal cabinet
 - (D) Both A and C
- The earlier Get and Put instructions were replaced with the MOV instruction:
 - True
 - False
- The MOV instruction will:
 - (A) Copy from a source word. data.
 - (B) Write to a destination word.
 - (C) Move alphanumeric data.
 - (D) Both A and B.
- Data compare instructions do not include a Not Equal To instruction:
 - True
 - False.
- Which of the following is not a valid Data Compare instruction?
 - < >
 - < =
 - =
 - > = <

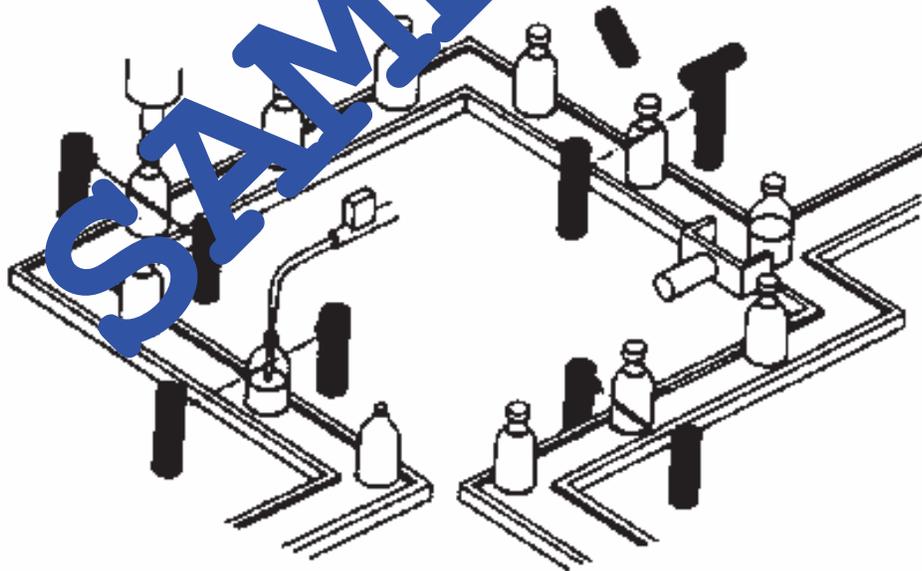
Scenario 1:

Design, program and test and document the following control system using Math Functions. (Use the I/O Simulator with LogixPro)

KWT Automation has secured the contract to alter the bottling line at a local sparkling water company. You are assigned the task of writing a program that will operate the system. The graphic below shows the layout of the line. A sensor has been added at the beginning of the line to count the bottles entering the line. There is also a sensor counting the bottles completing the process. The bottling operation should maintain a 94% success rate to meet projected profit margins. If the rate falls below 94% an alarm must sound to alert maintenance personnel. If it falls below 90% the alarm must sound and the line must be shut down.

The system must operate effectively no matter how many bottles have been processed. Remember, at any given time there are 10 bottles in the system. Also be careful that a division by zero does not occur while you are calculating the percentage. A division by zero will usually cause a CPU fault.

NOTE: The F8 files are not functional in LogixPro, therefore all data must be converted to whole numbers, and the equivalent values must be used.

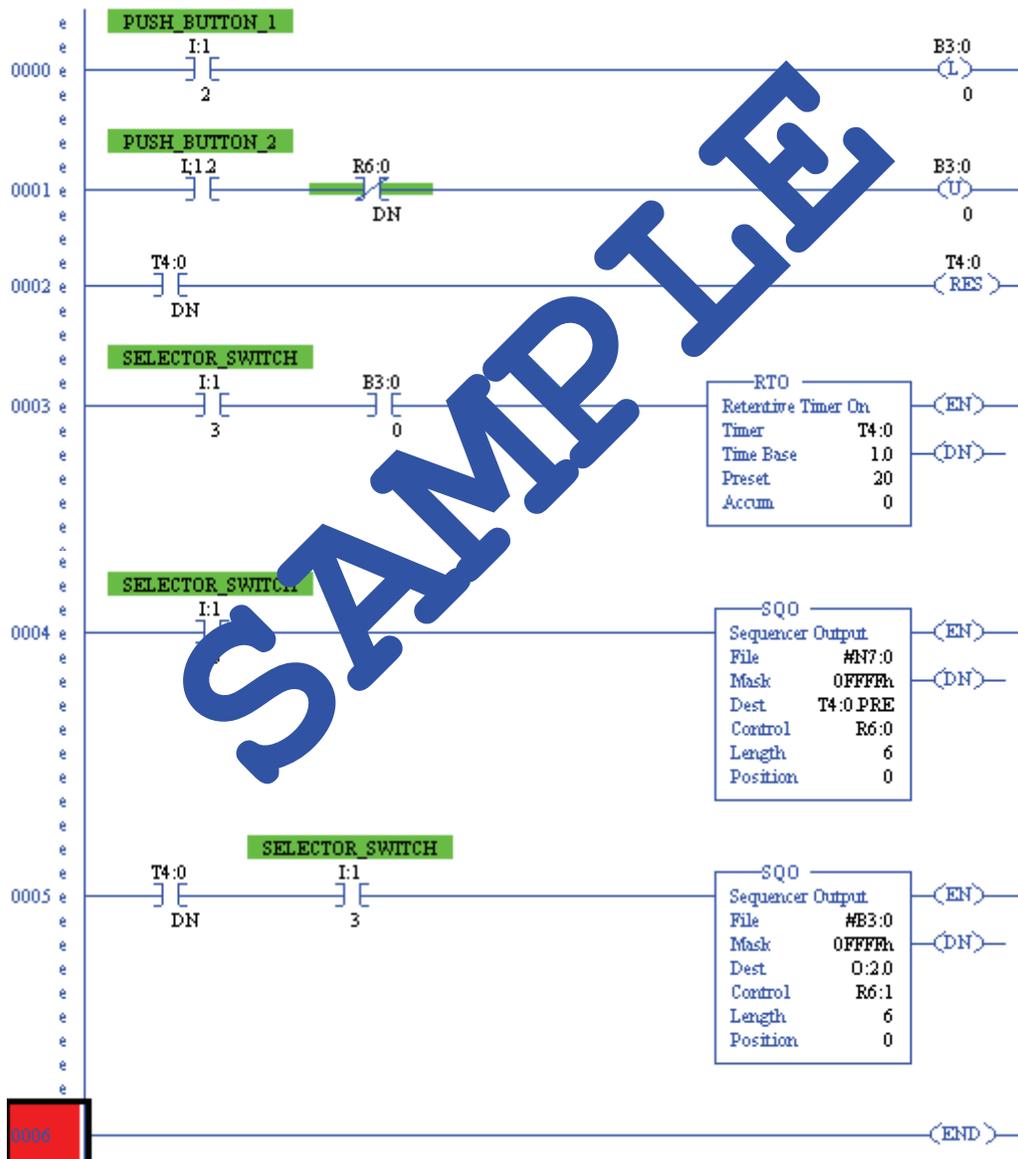


Practical Exercise:

This exercise will utilize a Master and Slave PLC. You must be running a DH485 network with at least 2 nodes in order to complete this exercise.

The program on the slave PLC will contain two sequencer instructions, SQO:0 and SQO:1. It will also contain one RTO instruction. As shown in the diagram below, the T4:0/DN bit will be used to forward the sequencers to the next step.

Each sequencer has six steps. The second sequencer will be called the Preset-Control Sequencer. It will be used for regulating the delay between steps of the first sequencer, the Event-Control Sequencer. The Preset-Control Sequencer transfers time delay information to the T4:0 Preset Register. The DN bit of timer T4:0 is used to reset the RTO.



Introduction:

HMIs are an essential part of many PLC managed processes. The competent technician will understand the uses and operation of Human Machine Interfaces.

Terminal Objective:

Upon completion of this unit the learner will be able to discuss HMIs.

Enabling Objectives:

1. Describe the use of HMIs.
2. Discuss HMI graphics.

SAMPLE

Assignment Sheet 1:

1. What are the things that all effective troubleshooting has in common?
 - Symptom recognition.
 - Problem isolation.
 - Corrective action.
 - All of the above.
2. When troubleshooting a PLC managed system, it is never necessary to question the operator.
 - True
 - False
3. If this LED is not on, the DC power supply is not functioning:
 - Processor Fault.
 - I/O Fault.
 - DC Power On.
 - Battery Low.
4. If this status light is on, a fault exists within the processor.
 - Processor Fault.
 - I/O Fault.
 - DC Power On.
 - Battery Low.
5. This status light will be on when a parity error exists between the memory module and the processor module;
 - Memory Fault.
 - I/O Fault.
 - DC Power On.
 - Processor Fault.

Appendix A

Status File Overview

The status file lets you monitor how your operating system works and lets you direct how you want it to work. This is done by using the status file to set up interrupts, load memory module programs, and monitor both hardware and software faults.

Note: Do not write to status file data unless the word or bit is listed as dynamic/static configuration in the descriptions that follow. If you intend writing to status file data, it is imperative that you first understand the function fully.

The status file **S:** contains the following words:

Word	Function applies to all processors
S:0	Arithmetic Flags
S:1	Processor Mode Status/Control
S:2	STI Bits / DH485 Comms
S:3L	Current Watchdog Scan Time
S:3H	Watchdog Scan Time
S:4	Free Running Clock
S:5	Minor Error Bits
S:6	Major Error Code
S:7,S:8	Suspend Code/Suspend File
S:9,S:10	Active Nodes (DH-485)
S:11,S:12	I/O Slot Enables
S:13,S:14	Math Register
S:15L	Node Address
S:15H	Baud Rate

Word	Function applies to SLC 5/02, SLC 5/03, SLC 5/04 processors
------	--

address of Bit file 3 is contained in address N10:2. Therefore, if N10:2 contains the value 5, B3:[N10:2] indirectly refers to address B3:5. Other examples include:

- N7:[N7:0], N[N7:0]:[N7:1], N7[T4:0.ACC], C5:[N7:0]

Indexed Indirect Addressing

- You may specify a combination of indirect and indexed addressing. The processor first resolves the indirect portion of the address and then adds the offset from the Index Register S:24 to come up with the final address. For example:
- #N7:[N10:3] where N10:3= 20 and S:24 =15 The actual address used by the instruction is N7:35.

Troubleshooting Faults

The major error fault code, indicates the probable cause of faults, and recommends corrective action. The following section describes the different ways to automatically clear a fault using your programming software.

Automatically Clearing Faults: SLC Processors

- Set the Fault Override at Power Up bit S:1/5 in the status file to clear the fault when power is cycled, assuming the user program is not corrupt.
- Set one of the auto load bits S:1/0, S:1/11, or S:1/12 in the status file of the program in an EEPROM to automatically transfer a new non-faulted program from the memory module to RAM when power is cycled.
- Refer to Status Bits for more information on bits S:1/13, S:1/8, S:1/10, S:1/11, S:1/12, S:5/0-7, and S:36/0-7. You can declare your own application specific major fault by writing your own unique value to S:6 and then setting bit S:1/13.

Unspecified voltages (relay contact for customer connections)	Gray
---	------

Voltages DC	Color
24v DC(+)	Brown
24v DC(-)	Blue
DC grounded circuits	White / Blue stripe
Foreign Voltage (Outside the panel)	Yellow
Temporary Jumpers	Purple (easy to see)
Oddball voltages (Speed Control potentiometer wiring)	Violet
Unspecified voltages (relay contact for customer connections)	Gray

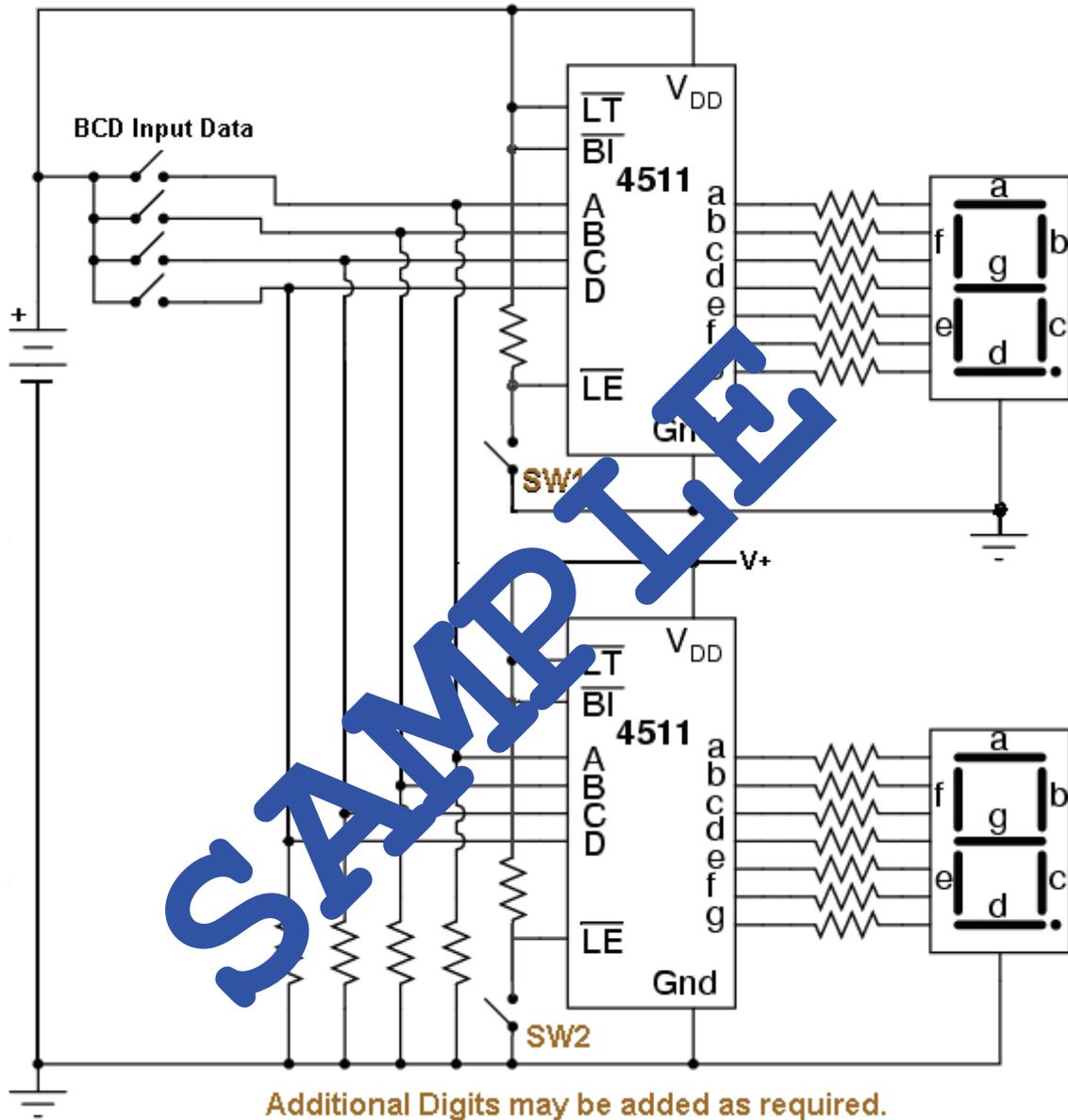
Inside the panel UL-508 allows the following
18 AWG = 7 Amps max
16 AWG = 10 Amps max
14 AWG = 20 Amps max
12 AWG = 25 Amps max
10 AWG = 40 Amps max

The allowable Current is Lower than the above stated values when the wire is in conduit, wireway or ductbank. Go to the NEC for these requirements.

- Most places use **14 AWG maximum for 120 VAC** field wiring, regardless of amps.
- Typically **16 AWG maximum for 24 VDC** field control circuits, and **18 AWG shielded twisted pair for 4-20 mA** wires (Belden 1032A or eq.)
- Avoid using 14 - 12 AWG shielded twisted pair for 4-20 mA wiring - it adds nothing in function, costs more, and is a pain to work with inside instrument enclosures.
- Do not allow wire nuts on 4-20 mA or other analog signal wires - they get loose, they pick up EMI/RFI, and are just no good. Field splices should only be made at terminal strips in junction boxes!
- It's important to note any **"foreign voltage"** on the panel, since "foreign" wires may still be hot after the panel is powered down and there isn't really a standard.

Multiplexed Digits

By making use of the 4-bit latches that are built into the 4511 IC, we can easily multiplex the digits if so desired... By properly controlling the state of each latch enable pin (LE) we can use the same input data lines (4 switches) to selectively write to each 7-Segment display independently... With just a minor modification to our circuit, we will be able to essentially treat each digit as a unique 4-bit memory location where BCD data of our choosing can be stored and retained...



In the above schematic diagram, each display may be written to separately... First the BCD equivalent of the desired numeral (0-9) is set using the 4 data switches... If SW1 is then closed, the current BCD input data will enter the latch of the upper 4511 IC, and will be passed on to the decoder causing the numeral to be displayed... if SW1 is then opened, the latch will retain the current data, but will now ignore any changes on its inputs... The desired numeral will continue to be displayed by the upper LED display until power is lost, or SW1 is again closed and new data is allowed to enter its latch...

The previous pages where just a sample of ...

PLC SCADA Certificate Course

<https://Feedforward.com.au/PLC-SCADA-System.htm>

By BIN95



To download this 2 course (PLC SCADA) bundle, please visit

<https://Feedforward.com.au/PLC-SCADA-System.htm>

Using Rockwell - Allen Bradley look & feel PLC training course and a Siemens SCADA basics course, you get a more blended and sound learning experience. A fitting bundle of courses, as the PLC hardware is most commonly used for RTU SCADA systems. The PLC SCADA System Basics course bundle is a synergetic combination, as PLC projects and applications sometimes include the maintenance tech working with SCADA system. Likewise, the SCADA Engineer can't do their job without knowing PLC basics. Also, the network security section in SCADA system basics course is helpful in maintaining the integrity of the SCADA network. Another thing left out of most online PLC SCADA course.

For many more courses, please browse

Feed Forward Publications

at ... <https://www.feedforward.com.au>

(Feed Forward Publications provides Industrial related Ebooks and Software and is a subsidiary of BIN95.com.)