Electric Power / Controls

Programmable Logic Controller

Basic Principles Using the Programming Software

Courseware Sample

88270-F0

Lab-Volt®

ELECTRIC POWER / CONTROLS

PROGRAMMABLE
LOGIC CONTROLLER
BASIC PRINCIPLES USING
THE PROGRAMMING SOFTWARE

by
the Staff
of
Lab-Volt Ltd.

Printed in Canada
October 2011

Table of Contents

Table of Contents

Sample Exercise

Extracted from

Student Manual

Exercise 1

Familiarization with the PLC Trainer and RSLogix 500

EXERCISE OBJECTIVES

- To become familiar with the Lab-Volt PLC Trainer
- To run the RSLogix 500 software.
- To enter the default project files path.
- To create and save a project file.

DISCUSSION

Introduction to the Lab-Volt PLC Trainer, Model 3240-4

> **Note:** *If you are using one of the following PLC trainer models: 3240-A, 3240-3, 3270-4, or 9066, skip this part of the DISCUSSION, which deals specifically with Model 3240-4, and refer to Appendix F through I of this manual for a detailed description of the PLC model you are using. Then, go back to Exercise 1 and proceed with the next DISCUSSION section, entitled The RSLogix 500 Software.*

Programmable logic controllers (PLC's) permit hardware control devices such as relays, timers, counters, and drum controllers (sequencers) to be replaced by programmable solid-state components and programmed instructions. To do so, a ladder program, consisting of a set of instructions representing the logic to be followed by the PLC, is developed, entered, and downloaded to the PLC. Once placed in the Run mode, the PLC follows this logic to interpret the input signals sent to it from input devices and operate its output devices accordingly.

The Lab-Volt PLC Trainer, Model 3240-4, features an Allen Bradley MicroLogix 1200 PLC. This PLC can be programmed by using the RSLogix 500 software from Rockwell Software. A direct communication link (DF1 full duplex) is used to connect the PLC to the computer that runs RSLogix 500, sparing the need for any interface between them.

The PLC has 14 numbered inputs, labeled 0 through 13, and 10 numbered outputs labeled 0 through 9. The trainer includes a built-in 24-VDC voltage for powering PLC output devices.

Figure 1-1 shows the front panel of the trainer.

- PLC inputs 0 through 13 are internally connected, through a PLC input interface, to 14 pairs of plug-in jacks mounted at the right top of the front panel. Each pair of jack permits activation of the corresponding PLC input using a 24-VDC voltage from an external PLC input device. Three momentary pushbutton switches and four toggle switches, labeled 1 through 8 and mounted on the front panel, can be used as PLC input devices: when connected to any of the PLC input jacks, they permit activation of the PLC inputs with a 24-VDC voltage provided by the built-in source of the trainer.

- PLC outputs 0 through 9 are internally connected, to 10 plug-in jacks mounted at the right middle of the front panel. The jacks each correspond to a PLC output, and permit connection of external PLC output devices, such as relay coils and actuators, that are energized or de-energized as the controller program is being executed. The PLC output jacks are hardwired to the built-in 24-VDC source and are used for energizing the PLC output devices. Above each jacks is a light indicating the status of the corresponding PLC output.

The features of the trainer front panel are described below (refer to Figure 1-1).

1. **Access door to the PLC input terminals**

2. **Memory module expansion compartment**: provides access to a 10-pin connector for installation of an optional memory module and/or real-time clock.

3. **RS-232 communication port** (Primary port, or Channel-0 port): used to connect the PLC to the computer that runs the RSLogix 500 software, using a serial cable (a 1761-CBL cable). The recommended protocol for this configuration is DF1 full duplex.

4. **24V DC Power Supply**: provides 24V DC to power the PLC outputs and the different pushbuttons and toggle switches.

5. **Trim pots**: permit modification of data in a register of the controller (the TPI register). Throughout the course, these potentiometers **must not be adjusted or tampered with**, as this will modify the content of the TPI register.

6. **Access door to the PLC output terminals**

7. **PLC output terminals**

8. **PLC output status indicators**: LED's indicating the current status (logic state 0 or 1) of the bits associated with PLC outputs 0 through 9 in the output data file of the PLC.

9. **PLC input status indicators**: LED's indicating the current status (logic state 0 or 1) of the bits associated with PLC inputs 0 through 13 in the input data file of the PLC.

10. **Analog Inputs and Outputs Expansion Card**: Part of Model 3244-40, this PLC expansion card provides analog inputs and outputs to the PLC trainer.

11. **I/O bus interface connector**: used to connect an expansion I/O module to the controller, through a flat ribbon cable.

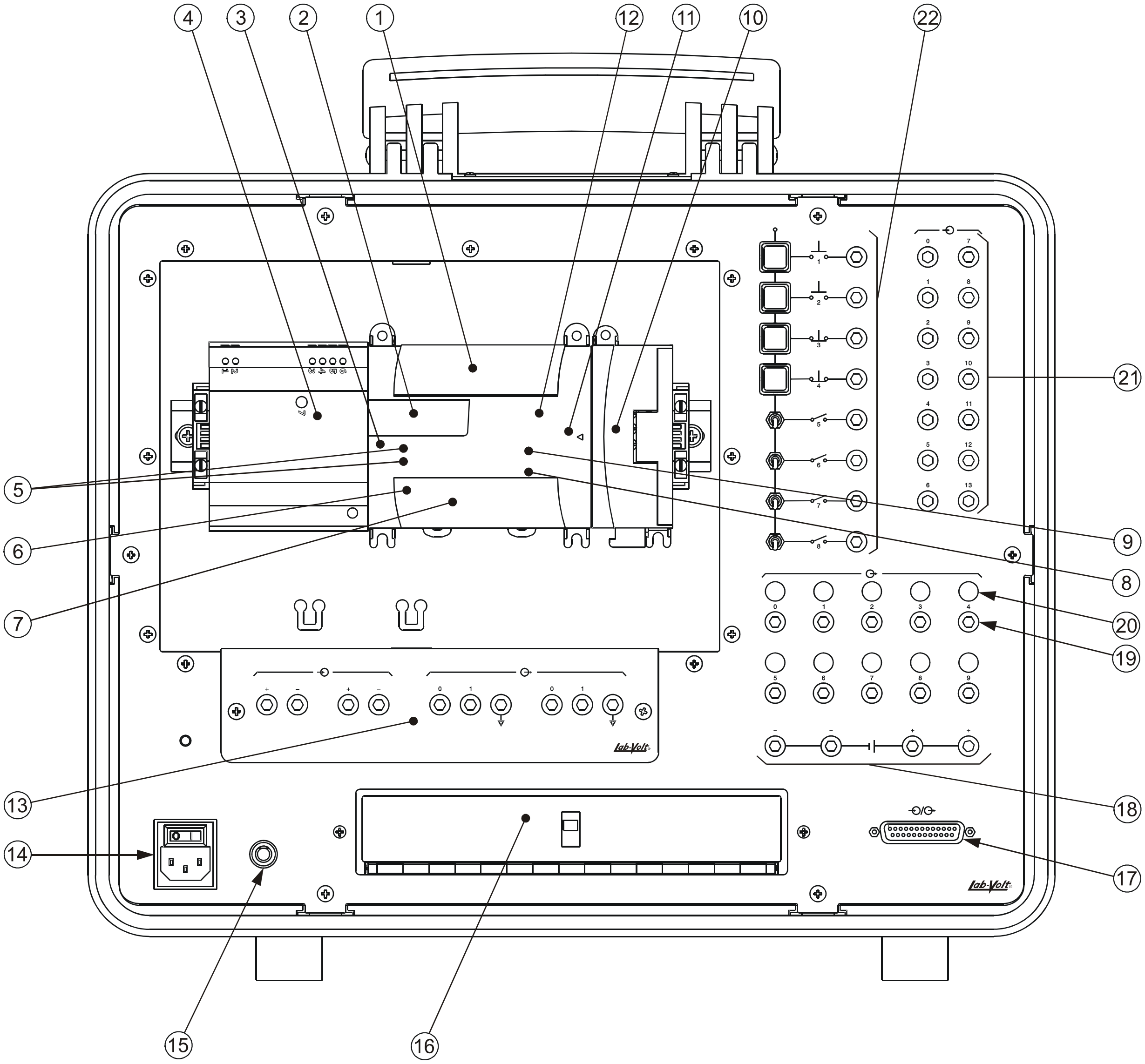


Figure 1-1. PLC Trainer, Model 3240-4 (front view).

12. **PLC status indicators**: LED's indicating the current status of the controller:

- **POWER**: this LED is on when the PLC is properly powered. It is off when there is no input power to the PLC or when a power error condition occurs.
- **RUN**: this LED is on when the PLC is executing a program in the Run mode. It is off when no program is being executed.
- **FAULT**: this LED is off when there is no fault. It is on when the controller hardware is faulty. It flashes when a major hardware or software fault has been detected.

- **FORCE**: this LED is on when one or more PLC inputs or outputs are forced on or off. It is off when no forces are installed.

- **COMM 0**: This LED is off when the controller is not transmitting data via the PLC communication port (channel-0 port). It is on when the controller is transmitting data via this port.

- **DCOMM**: This LED is on when the controller is in the default communication mode. It is off when the controller is in the user-configured communication mode.

13. **Analog Inputs and Outputs Expansion Panel**: provides connection to the PLC expansion card (included with Model 3244-40).

14. **Power Inlet and Switch**: Connect the PLC Trainer from that inlet using the appropriate power cord (included) to a standard wall AC outlet. The inlet also includes the power switch to turn on and off the trainer.

15. **AC-line voltage RESET button**: used to reset the breaker of the built-in AC-line voltage source of the trainer.

16. **Fault Panel**: twelve fault switches are located behind the fault panel door. These switches creates, when turned on, electrical connections problems which permits the student to troubleshoot the trainer.

17. **P-SIM-to-PLC Interface connector**: used to connect the PLC Trainer to the P-SIM to PLC Interface, Model 3243, through a DB-25 flat cable. The interface, which converts RS-232 signals into PLC signals, and vice-versa, is required for the second level of the Lab-Volt PLC Training Program. It allows the PLC to control animated industrial processes on a computer with the P-SIM Simulations software, Model 91773.

18. **Jacks of the 24V-DC power supply (4)**: provides 24V-DC to external devices.

19. **PLC Outputs**: when PLC outputs are activated, a DC voltage of 24 V is applied by a relay (from the PLC) to the jack to which external PLC output devices, such as relay coils and motor drives can be connected.

20. **PLC Output Lamps**: these lamps are on when their PLC output is activated (that is, when the bit associated with PLC output in the PLC output data file is at logic 1, or when this bit is forced on).

21. **PLC Inputs**: permit activation of the PLC inputs upon DC voltage of 24 V. The voltage can come from one of the eight switches mounted on the trainer front panel, or from external PLC input devices rated at 24 V DC. Inputs 0 to 3 can be used as high-speed inputs (up to 20 kHz).

22. **Pushbuttons and Toggle Switches**: Two NO (normally open) pushbuttons, two NC (normally closed) pushbuttons and four toggle switches are connected, on one side, to the 24V-DC power supply and can be used to input this voltage to PLC inputs.

The RSLogix 500 Software

The RSLogix 500 software is used to program and control Allen-Bradley PLC's with a PC-type computer. This software allows you to create, edit, and monitor PLC ladder programs. It also allows you to document ladder programs, to store projects (ladder program files and all other associated files) on disk, and to print complete reports on a project.

Running RSLogix 500

RSLogix 500 runs under the Microsoft® Windows® environment. RSLogix 500 is started by selecting the corresponding command in the Rockwell Software program group. Figure 1-2 shows a view of the RSLogix 500 window upon creation of a new project. This window consists mainly of the following elements:

- A standard (main) toolbar which allows you to select a function from a series of menus.

- An online section with four scrollable bars that allow you to see the operational mode of the PLC and whether or not online forces are installed.

- An instruction-insertion section with instruction-category tabs permitting the selection of a category of instructions. When a category tab is selected, a toolbar listing all the instructions (symbol or mnemonic) pertaining to the selected category is displayed. An instruction is inserted in a rung of the ladder program by clicking its button in the toolbar.

- An Add-In button representing a portal to Visual Basic Applications (VBA).

- A Run Macro button used to display the Macro dialog box and execute, modify, or remove a macro.

- A status bar that prompts you to take an action while you are using the software, and that provides information relevant to the current file: file and rung location of the cursor, mode selected for the cursor entry, etc.

- A project tree, which contains all files associated with the current project. You can usually click an icon in this tree and then click the mouse right button for a functional menu that provides quicker access to relevant functional choices.

- A ladder view where you can observe the ladder logic file (program file) and edit the ladder logic.

- A results window that displays the results of search and verification procedures.

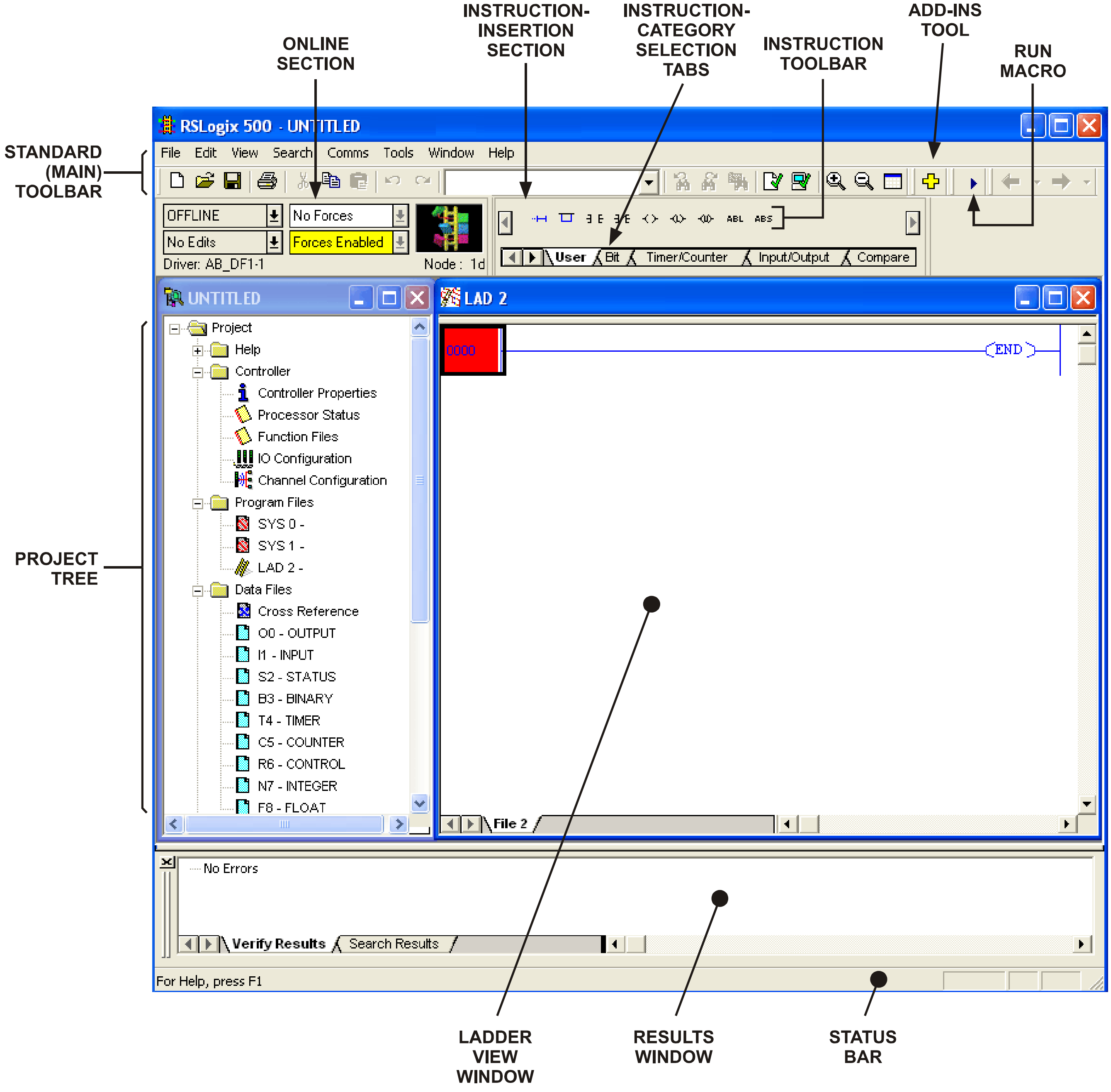
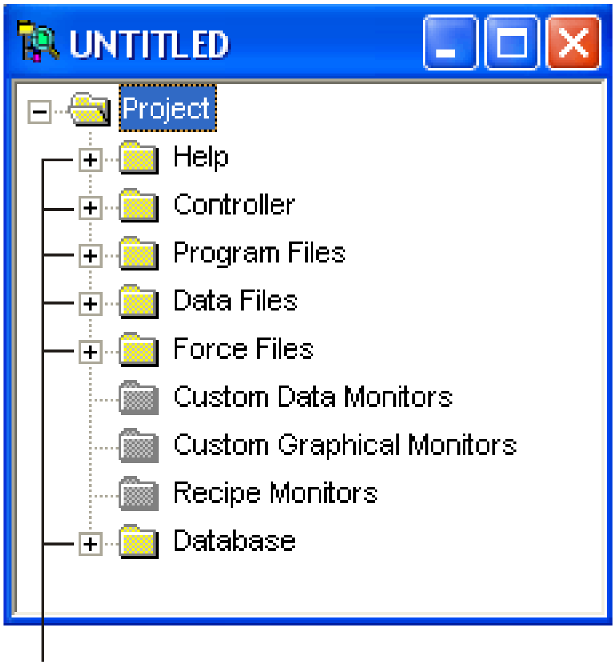


Figure 1-2. The main elements in the RSLogix 500 window.

Projects

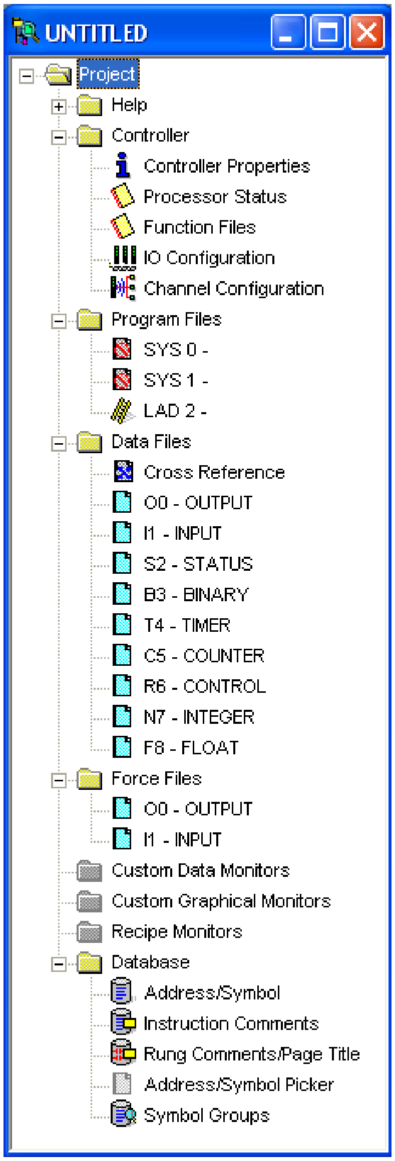
RSLogix 500 is based on projects. A project is a complete set of files associated with a logic program. To create a project, the **New** command in the **File** menu must be selected. This causes RSLogix 500 to prompt you to select the type of processor you will communicate with, and to create a project tree. This tree is the entry point to all files associated with the newly created project. Figure 1-3 shows an example of a project tree as it appears in RSLogix 500. As you can see, a project consists of several folders that contain files (controller files, program files, data files, etc.).



EACH OF THESE BUTTONS CAN BE CLICKED TO EXPAND THE CORRESPONDING FOLDER AND REVEAL THE FILES IT CONTAINS.

Figure 1-3. Example of a project tree in RSLogix 500.

Figure 1-4 is an expanded project tree that shows the files contained in the following three folders: **Controller**, **Program Files**, and **Data Files**.



UNTITLED

- Project
  - Help
  - Controller
    - Controller Properties
    - Processor Status
    - Function Files
    - IO Configuration
    - Channel Configuration
  - Program Files
    - SYS 0 -
    - SYS 1 -
    - LAD 2 -
  - Data Files
    - Cross Reference
    - O0 - OUTPUT
    - I1 - INPUT
    - S2 - STATUS
    - B3 - BINARY
    - T4 - TIMER
    - C5 - COUNTER
    - R6 - CONTROL
    - N7 - INTEGER
    - F8 - FLOAT
  - Force Files
    - O0 - OUTPUT
    - I1 - INPUT
  - Custom Data Monitors
  - Custom Graphical Monitors
  - Recipe Monitors
  - Database
    - Address/Symbol
    - Instruction Comments
    - Rung Comments/Page Title
    - Address/Symbol Picker
    - Symbol Groups

Figure 1-4. Project tree showing the files in the Controller, Program Files, and Data Files folders.

– The **Controller** folder consists of files that contain the controller properties, the processor status, the function files, the inputs/outputs (I/O) configuration, and the communication channel configuration.

– The **Program Files** folder can contain up to 256 program files. File SYS 0 (system program) is always included and contains the controller configuration. File SYS 1 is always included and is reserved for internal controller use. File LAD 2 is always included and is the main ladder program. Program files 3 to 255 are optional and used to store subroutine programs.

– The **Data Files** folder can contain up to 256 data files. The data files contain status information on all the instructions in the main ladder program and its subroutine(s), if any. There are several types of data files. The first ten data files have default types, as shown in Table 1-1. Other data files can be user-defined or will automatically be created by RSLogix 500 when additional data is to be stored. For example, you can create and define these files for the storage of:

- bits, timers, counters, control, or integer data; or
- programmable limit switch data (6-word elements); or
- double words, message word elements, and PID word files.

File Cross Reference	Stores the cross-reference report.
File O0 - (OUTPUT)	Stores the status of each PLC output.
File I1 - (INPUT)	Stores the status of each PLC input.
File S2 - (STATUS)	Stores information on PLC operation.
File B3 - (BINARY)	Stores binary data for internal relay logic.
File T4 - (TIMER)	Stores timer data (accumulated value, preset value, and timer status bits).
File C5 - (COUNTER)	Stores counter data (accumulated value, preset value, and counter status bits).
File R6 - (CONTROL)	Stores control data (length, pointer position, and status bits) for shift registers and sequencer instructions.
File N7 - (INTEGER)	Stores numeric values or bit information.
File F8 (FLOAT)	Stores a value with a range of 1.1754944e-38 to 3.40282347e+38.
Files 9-255	User-defined files

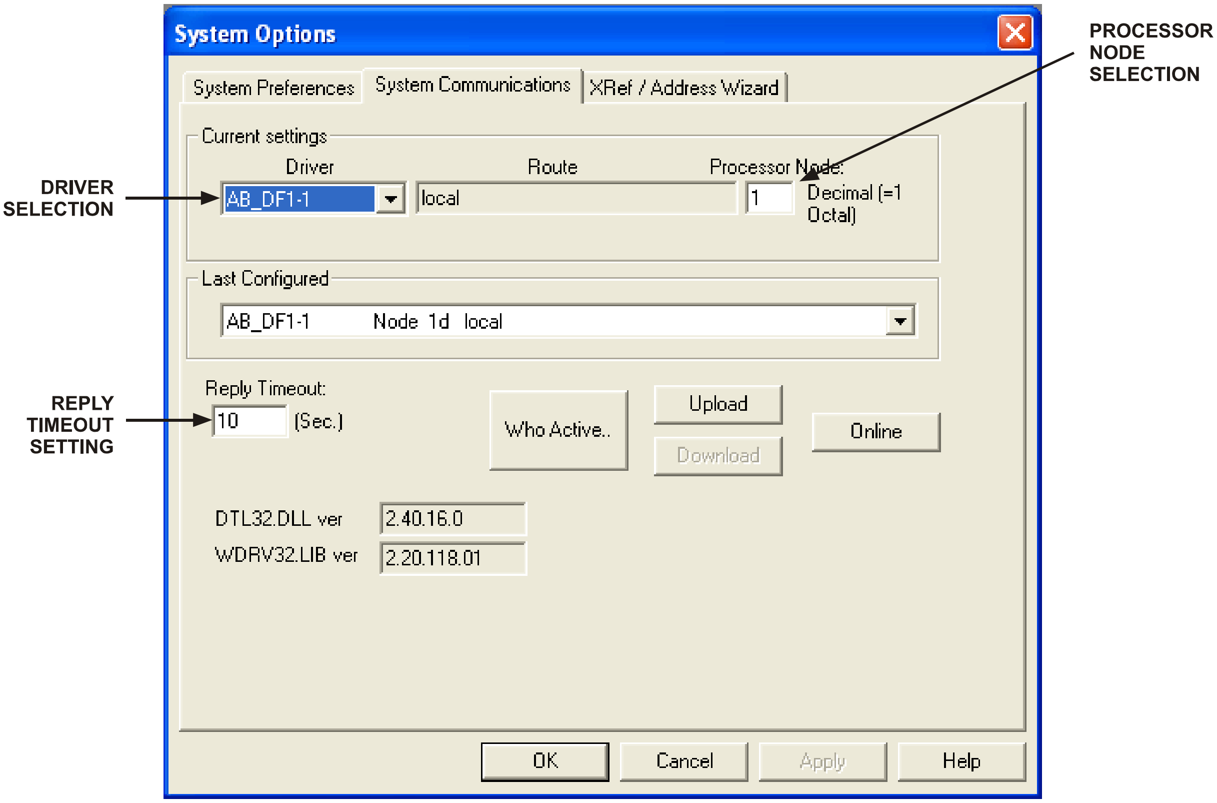
Table 1-1. Default type data files.

A project, that is, the complete set of files associated with a logic program, is saved by choosing either the **Save** or **Save As** command in the **File** menu.

Configuring System Communications

System communications, that is, the communication between a PLC and the computer station that runs RSLogix 500, should be configured before creating a new project. This is performed by choosing the **Options** command in the **Tools** menu. This opens the **System Options** dialog box. Clicking the **System Communications** tab places the corresponding folder on top of the dialog box. Figure 1-5 shows the **System Communications** folder of the **System Options** dialog box.

The driver settings should correspond to those made in RSLinx upon installation of RSLogix 500 by your instructor, unless they have been modified since then. To learn how to create an appropriate communication driver, ask your instructor (the procedure is in the Instructor Guide 36017-1)



Current settings

Driver: AB_DF1-1

Route: local

Processor Node: 1 Decimal (=1 Octal)

Last Configured: AB_DF1-1 Node 1d local

Reply Timeout: 10 (Sec.)

DTL32.DLL ver 2.40.16.0

WDRV32.LIB ver 2.20.118.01

DRIVER SELECTION

REPLY TIMEOUT SETTING

PROCESSOR NODE SELECTION

Figure 1-5. Configuring system communications.

The **Driver** drop-down list allows you to select the type of driver used to link the computer station to the PLC. The data field called **Processor Node** allows you to enter the processor node. The data field called **Reply Timeout** field allows you to change the value of the reply timeout. Once system communications are set as desired, the **System Options** dialog box is closed by clicking the **OK** button. The settings established in the **System Communications** folder of the **System Options** dialog box will be used upon creation of any new project, and will be applied when you attempt to download a ladder program to the PLC.

Editing System Preferences

System preferences can be edited by choosing the **Options** command in the **Tools** menu. This opens the **System Options** dialog box. Clicking the **System Preferences** tab places the corresponding folder on top of the dialog box. This folder provides various options that allow RSLogix 500 to be set according to your needs and/or preferences. Among these options is the field called **Project Files Search Path**. This field allows you to select the path where you want your projects to be stored.

Procedure Summary

In this exercise, you will familiarize yourself with the main elements of RSLogix 500. You will learn how to edit the project files path, configure system communications, create a new project, see that the project tree is the entry point to all files associated with a project, and you will save a project to a file.

EQUIPMENT REQUIRED

Refer to Appendix A of this manual to obtain the list of equipment required to perform this exercise.

PROCEDURE

Running RSLogix 500

☐ 1. Turn on the computer and start RSLogix 500.

☐ 2. Observe that there is no project tree nor is there any program file (ladder program) displayed in the RSLogix 500 window. This occurs because no project file has been created or opened so far.

Editing the Default Project Files Path

☐ 3. Choose the **Options** command in the **Tools** menu. This opens the **System Options** dialog box. Select the **System Preferences** folder, and observe that various options are available.

You can set the **Project Files Search Path** to the default path:

C:\PROGRAM FILES\ROCKWELL SOFTWARE\RSLOGIX 500 ENGLISH\PROJECT

You can enter another path if you want your project files be saved to another location on the hard disk of the computer (or any other memory media).

Configuring System Communications

□ 4. Select the **System Communications** folder. The **Current settings** section of this folder should indicate that the driver is an **AB_DF1-1** and the processor node is set to **1**. These are the normal settings to be used, unless otherwise specified by your instructor.

Set the **Reply Timeout** to **5 s**, if it is not already set to this value.

Click the **OK** button to save the system options and close the **System Options** dialog box.

Creating a New Project

□ 5. Choose the **New** command in the **File** menu to initiate the creation of a new project. This opens a dialog box that prompts you to type a processor name. Type **EXERC_1** as the processor name.

□ 6. In the processor list, select the processor type on PLC Trainer Model 3240-4, that is, **Bul. 1762 MicroLogix 1200 Series C**.

> **Note:** *For the standalone PLC Model 3270-4, select Bul. 1761 MicroLogix 1000;*
>
> *For PLC Trainer Model 3240-A, or the Programmable Logic Controller, Model 9066, select Bul. 1763 MicroLogix 1100 Series A.*
>
> *For PLC Trainer, Model 3240-3, select Bul. 1764 MicroLogix 1500 LRP Series C.*

□ 7. Observe that the bottom portion of the **Select Processor Type** dialog box indicates the communications settings. These settings are the same as those defined in the **System Communications** folder of the **System Options** dialog box.

□ 8. Click the **OK** button in the **Select Processor Type** dialog box. This will close this dialog box and create a new project in the computer memory.

The Project Tree

□ 9. Observe that a project tree and file LAD 2 are now displayed in the RSLogix 500 window.

Scroll through the EXERC_1 project tree to see all the files it contains.

Which file contains the main ladder program?

____

□ 10. Open data file **I1 - INPUT**. To do so, select it using the mouse, click the mouse right button to display the context-sensitive menu, and choose the **Open** command in this menu. A window will appear, showing the contents of data file **I1 - INPUT**. This file is used to store the logic state of each of the PLC inputs.

Close data file **I1 - INPUT** by clicking the **Close** button of the corresponding window.

Other files in the project tree can be opened using the same procedure.

The Instruction Toolbar

□ 11. Locate the instruction toolbar and the instruction-category selection tabs in the upper middle section of the RSLogix 500 window.

Click the **User** instruction-category selection-tab to select it, then place the mouse pointer on one of the instruction buttons in the instruction toolbar above this tab. Observe that a floating tooltip window appears and indicates which instruction is associated with the button.

□ 12. Click each other instruction-category selection tab while observing the corresponding list of instruction buttons displayed for each category. Describe what happens.

____

____

The Online Section

□ 13. Locate the **Online** section in the RSLogix 500 window. Observe that this section indicates that the operational mode is currently set to **Offline**. This implies that the project you are working on is in the computer memory, not in the PLC memory.

Observe that the **Online** section also indicates that no forces are applied, as well as the driver type and the processor node number.

Saving a Project to a File

□ 14. The project created in this exercise only exists in the computer memory. To save it to a file on the hard disk of the computer, choose the **Save** or **Save As** command in the **File** menu.

This opens the **Save Program As...** dialog box. This box prompts you to type a project filename (the processor name previously entered upon selection of the processor type is suggested). Keep the processor name the same as the project filename (EXERC_1).

Make the following observations:

- the path leading to the location where project files are to be stored is indicated at the top of the dialog box;
- the extension of project filenames is .RSS;
- you can include a revision note and a version number with the project file.

Click the **Save** button in the dialog box. This will cause project EXERC_1 to be saved in a file named EXERC_1.RSS.

☐ 15. Choose the **Close** command in the **File** menu to close project file EXERC_1.RSS. This will cause project EXERC_1 to be removed from the computer memory and the RSLogix 500 window.

☐ 16. Close RSLogix 500. Turn off the computer.

CONCLUSION

In this exercise, you became familiar with the main elements of RSLogix 500. You saw how to configure the system communications and preferences. You learned how to edit the path leading to the location where project files are saved. You created a new project. You learned that a project is the complete set of files associated with a logic program. You saw that the project tree is the entry point to all files associated with a project. You saved a project to a file.

REVIEW QUESTIONS

1. Name four types of hardware control devices that programmable logic controllers (PLC's) permit to replace.

   ____

   ____

2. Before a ladder logic program can be edited, what must be done first?

   ____

3. What is the point of entry to all files associated to a project?

   ____

4. What are the three program files that are automatically generated upon creation of a new project ?

   ____

5. Which program file contains the main ladder program?

   ____

Exercise 2

Programming Basics

EXERCISE OBJECTIVES

- To create a project and edit a PLC ladder program.
- To document a PLC ladder program.
- To print a report on a project.

DISCUSSION

PLC Ladder Program

Figure 2-1 shows a hardwired ladder diagram used to control the turning on and turning off of a pump motor. The horizontal lines similar to the steps of a ladder are called rungs. The two vertical lines at the rung extremities are called power rails. The left side rail is the line (L) power rail; it is connected to the hot side of the power supply. The right side rail is the neutral (N) power rail; it is connected to the common side of the power supply.

In each rung of the ladder diagram, **electrical continuity** exists when there is an uninterrupted electrical path between the L and N power rails. In this condition, the electrical contacts that are in the closed state on the rung form a path that permits electrical current to flow from the L power rail to the N power rail. This causes the output device on this rung to be energized. Examples of electrical contacts are switch contacts and relay contacts. Examples of output devices are relay coils and pump motors.

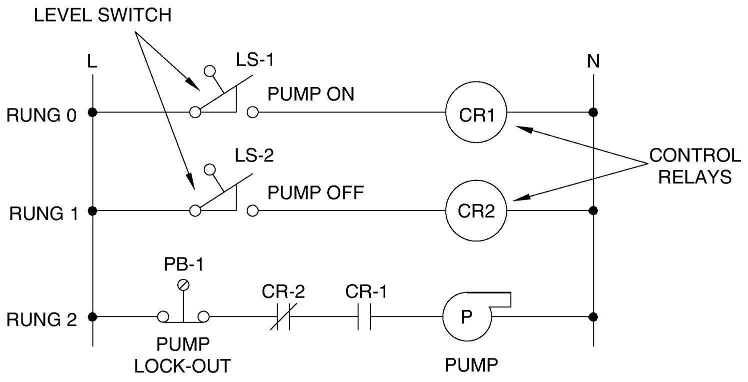


Figure 2-1. Hardwired ladder diagram.

Figure 2-2 shows a PLC ladder program that accomplishes the same control function as the ladder diagram of Figure 2-1. A PLC ladder program is a planned set of instructions resembling a hardwired ladder diagram. It consists of a line (L) power rail and a neutral (N) power rail between which one or more rungs are inserted.

Each individual rung contains one or more input instructions on its left-hand (L power rail) side, and a single output instruction or several output instructions placed in parallel on its right-hand (N power rail) side. In Figure 2-2, for example, the instructions Examine If Closed (XIC) and Examine If Open (XIO) are input instructions analogous to relay contacts. On the other hand, the instruction Output Energize (OTE) is an output instruction analogous to a relay coil.

Note: *"OTE" stands for output energize.*

The PLC ladder program is the main component of the project you download to a PLC. The PLC uses this program to interpret the signals present at its inputs and operate its outputs accordingly.

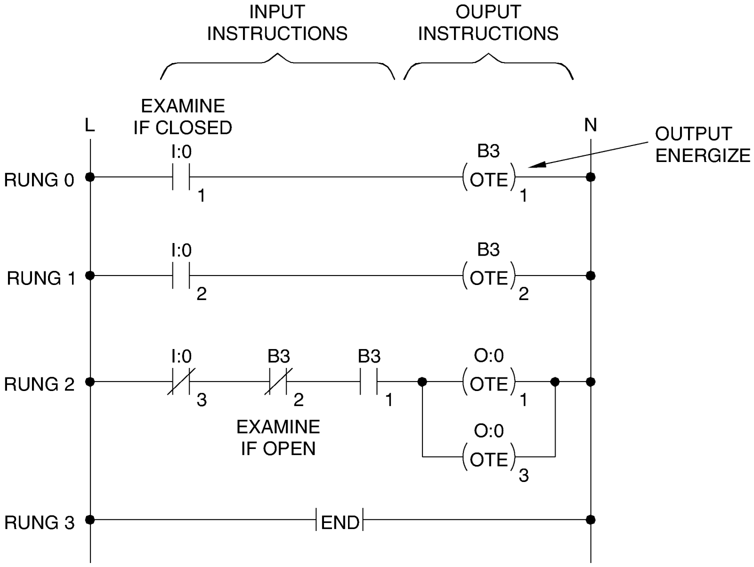


Figure 2-2. Equivalent PLC ladder program.

Logical Continuity

During PLC operation, the processor reads the status of the signals applied to the PLC inputs, through the PLC internal input interface, to determine whether these PLC inputs are activated or deactivated. The processor then updates the input data file (data file I1) bits accordingly. The processor then evaluates each rung of the ladder program individually, updates the timer, binary status, counter, and control data, and then modifies the output data file (data file O0) bits accordingly. The output data file bits are used to energize or deenergize relays in the PLC internal output interface, causing these relays to apply or remove power to/from the devices connected to the PLC output interface terminals.

To evaluate a rung, that is, to determine if the rung is true or false, the processor verifies if a continuous left-to-right path of true input instructions exists between the line (L) and neutral (N) power rails.

- When a continuous path of true input instructions exists, the rung is evaluated as true and the output instruction on this rung is true;

- When there is no continuous path of true input instructions on the rung, the rung is evaluated as false and the output instruction on this rung is false.

The status of a rung instruction (true or false) depends on the logic state of the data file bit this instruction is addressed to. Figure 2-3, for example, indicates the status of the instructions Examine If Closed (XIC) and Examine If Open (XIO), according to the logic state of the corresponding data file bit. From this figure, we can see that:

- the Examine If Closed (XIC) instruction is true when its associated bit is at logic state 1;

- conversely, the Examine If Open (XIO) instruction is true when its associated bit is at logic state 0.

IF THE CORRESPONDING DATA FILE BIT IS	THE STATUS OF THE INSTRUCTION IS	
	XIC EXAMINE IF CLOSED —| |—	XIO EXAMINE IF OPEN —|/|—
LOGIC 0	FALSE	TRUE
LOGIC 1	TRUE	FALSE

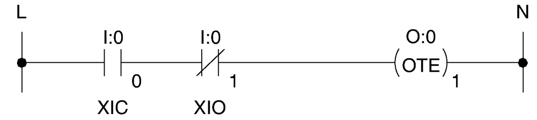
Figure 2-3. Truth table for the XIC and XIO input instructions.

The XIC and XIO instructions may represent the external input devices connected to the PLC. In that case, the status of these instructions depends on the logic state of their corresponding bit in the input data file (data file I1) of the PLC.

Figure 2-4, for example, shows a ladder rung having two input instructions and one output instruction. In this rung, instruction XIC I:0/0 represents the normally open contacts of a pushbutton switch, while instruction XIO I:0/1 represents the normally closed contacts of a pushbutton switch.

When an unbroken path of true instructions (logical continuity) exists, from left to right, on rung 0, this rung is evaluated as true and instruction Output Energize (OTE) O:0/1 is true. This implies that instructions XIC I:0/0 and XIO I:0/1 must both be true in order for instruction OTE O:0/1 to be true.

From the truth table in Figure 2-3, we can see that input data file bit I:0/0 must be at logic state 1 and input data file bit I:0/1 must be at logic state 0 for logical continuity to exist on rung 0. Consequently, PLC input 0 must be activated and PLC input 1 must be deactivated in order for rung 0 to be evaluated as true and instruction OTE O:0/1 to be true. This situation occurs when the pushbutton represented by instruction XIC I:0/0 is depressed (pushbutton contacts are closed) and when the pushbutton represented by instruction XIO I:0/1 is not depressed (pushbutton contacts are open).



XIC = EXAMINE IF CLOSED, ADDRESS I:0/0
XIO = EXAMINE IF OPEN, ADDRESS I:0/1
OTE = OUPUT ENERGIZE, ADDRESS O:0/1

Figure 2-4. Single rung containing input and output instructions.

Series (AND) and Parallel (OR) Logics

The ladder rung in Figure 2-4 was an example of series (AND) logic. Series logic means that all the instructions in the rung (XIC **I**:0/0 **AND** XIO **I**:0/1) must be true in order for output instruction OTE O:0/1 to be true.

The ladder rung in Figure 2-5 is an example of parallel (OR) logic. Parallel logic means that one or another path of true instructions must exist on the rung in order for the output instruction to be true. In Figure 2-5, either input instructions XIC **I**:0/1 **OR** XIC **I**:0/2 must be true in order for instruction OTE O:0/1 to be true. Parallel logic is programmed by branching instructions in a ladder rung.

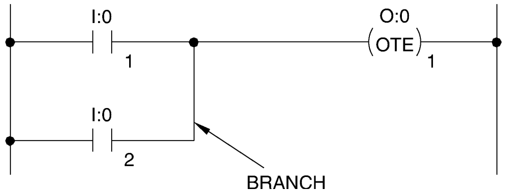


Figure 2-5. Combination of series and parallel logics.

Figure 2-6 shows a ladder rung that uses a combination of series and parallel logic. As this figure shows, branches can be inserted for both input and output instructions on a rung. **However, it is not possible to put two output instructions in series**; output instructions can only be placed in parallel. When two output instructions are placed in parallel, both are activated when logical continuity exists on the rung. For example, OTE instructions O:0/1 and O:0/2 in Figure 2-6 are true when XIC instructions (**I**:0/1 **OR** **I**:0/2 **OR** **I**:0/3) **AND** **I**:0/4 **AND** (**I**:0/5 **OR** **I**:0/6) are true.

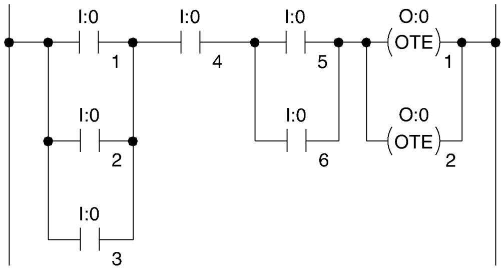


Figure 2-6. Combination of series and parallel logics.

Documenting a Ladder Program

You can document a ladder program by inserting rung comments, instruction descriptions, and address descriptions. This allows you to keep notes on:

- how your ladder program works;
- the purpose of an instruction or a rung;
- the type of input or output device (pilot lamp, pushbutton, limit switch, etc.) associated with each address;
- the conditions required for a rung to be true.

You can insert your comments and descriptions while you enter a ladder program or after you have entered it. The three types of comments and descriptions which can be inserted are described below.

- The **rung comment**: normally used to determine what the rung is meant to do. It is displayed just over the rung in the ladder view window.

- The **instruction description**: used to determine what the instruction is meant to do or the conditions required for the instruction to be true. This description specifies the type and address of the instruction. All instructions of the same type that have a common address will automatically have the same instruction description. The instruction description is displayed over each instruction in the ladder view window.

- The **address description**: used to identify the type of input or output device associated with an address. All instructions having the same address will automatically have the same address description. Note that address descriptions associated with instructions that are provided with an instruction description are not displayed in the ladder view. However, all address descriptions can be observed by opening the **Cross Reference** data file.

Creating and Printing Reports

Once you have finished a project, you may wish to keep a hard copy of the project. RSLogix 500 can generate many different reports on a project. The **Report** folder in the **Reports Options** dialog box allows you to select the reports to be printed. Some of the most important reports available in RSLogix 500 are described below.

- **General**: this report contains processor information (type of processor, processor name, number of program files and data files contained in the project, cross-reference information), alphabetical list of all the addresses and their description used in the program, as well as input/output configuration and channel configuration.

- **Data File List**: this report lists all the data files contained in the project.

- **Data Files**: this report provides the contents of each data file contained in the project.

- **Program File List**: this report lists all program files in the project.

- **Program Files**: this report contains the main ladder program as viewed in the RSLogix 500 window. It also contains all other program files (if any) in the project. This report is very useful when revising the program logic.

Once the reports to be printed are selected, printing is started by clicking the **Print** button in the **Report Options** dialog box.

Procedure Summary

In this exercise, you will create a new project and enter instructions in the main ladder program (program file LAD 2). You will insert rung comments, instruction descriptions, and address descriptions in the ladder program. You will create and print a project report.

EQUIPMENT REQUIRED

Refer to the Equipment Utilization Chart, in Appendix A of this manual, to obtain the list of equipment required to perform this exercise.

PROCEDURE

Starting a New Project

☐ 1. Turn on the computer and start RSLogix 500.

☐ 2. Create a new project having the following processor name: EXERC_2. You can refer to Exercise 1 of this manual for a recall of how to create a new project using RSLogix 500.

☐ 3. The project tree of processor EXERC_2 and program file LAD 2 are displayed in the RSLogix 500 window. Program file LAD 2 contains the main ladder program.

Notice that, at the beginning of a project, there is only rung 0 (the end rung) in program file LAD 2. Also note that the rung number is highlighted and enclosed in a box to indicate that this particular rung is the item currently selected in RSLogix 500.

Editing the Main Ladder Program

☐ 4. Figure 2-7 shows the main ladder program for processor EXERC_2. The next steps of this procedure show how to enter this ladder program using RSLogix 500.

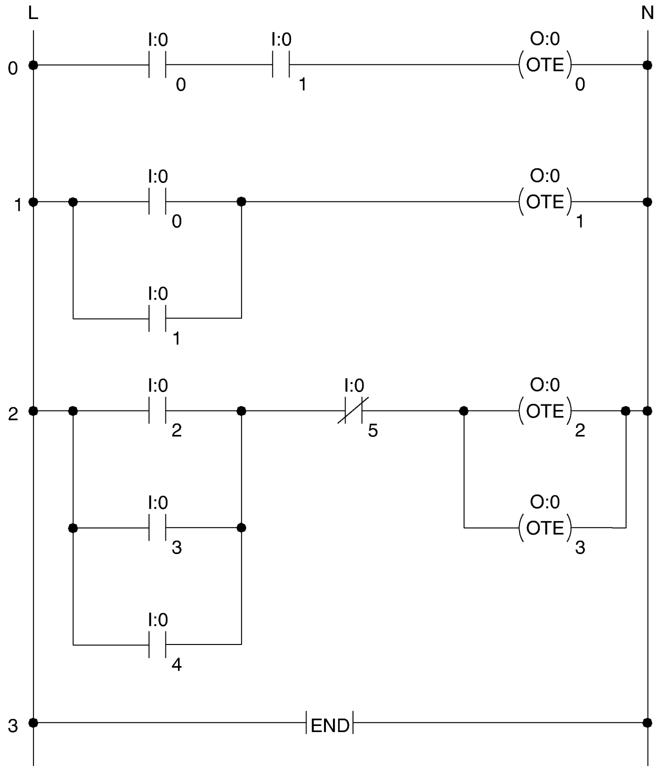


Figure 2-7. Main ladder program for processor EXERC_2.

☐ 5. Select the **User** category of instructions by clicking the corresponding instruction category selection tab.

Click the **New Rung** button on the instruction toolbar just above the category selection tabs. Notice that a new rung has been inserted into program file LAD 2 and that the end rung is now numbered 1. Also note that a column of e's appears within the highlighted box to the left (L) power rail to indicate that rung 0 is in the edit mode.

> **Note:** *If you make a mistake, go to the* ***Edit*** *menu and choose the* ***Undo*** *command to undo your last action.*

□ 6. Click the **Examine if Closed** button on the instruction toolbar to insert this instruction into rung 0. Notice that the XIC instruction is enclosed in a box to indicate that it is currently selected.

Type the following instruction address using the keyboard: **I:0/0**, then click the mouse left button to enter the address. The instruction address is now displayed in the ladder view. You have entered the first instruction of rung 0 shown in the ladder program of Figure 2-7.

> **Note:** *If, after typing the instruction address, you press ENTER instead of clicking the mouse left button, the Edit Description Type dialog box will appear. In that case, just click Cancel to close the box and enter the instruction.*

□ 7. Repeat the previous step to enter instruction XIC I:0/1 into rung 0.

□ 8. Click the **Output Energize** button on the instruction toolbar to insert this instruction into rung 0. Notice that the OTE instruction is on the right-hand side of rung 0 and enclosed in a box to indicate that it is currently selected. Type the following instruction address using the keyboard: **O:0/0**, then click the mouse left button to enter the address. The instruction address is now displayed in the ladder view.

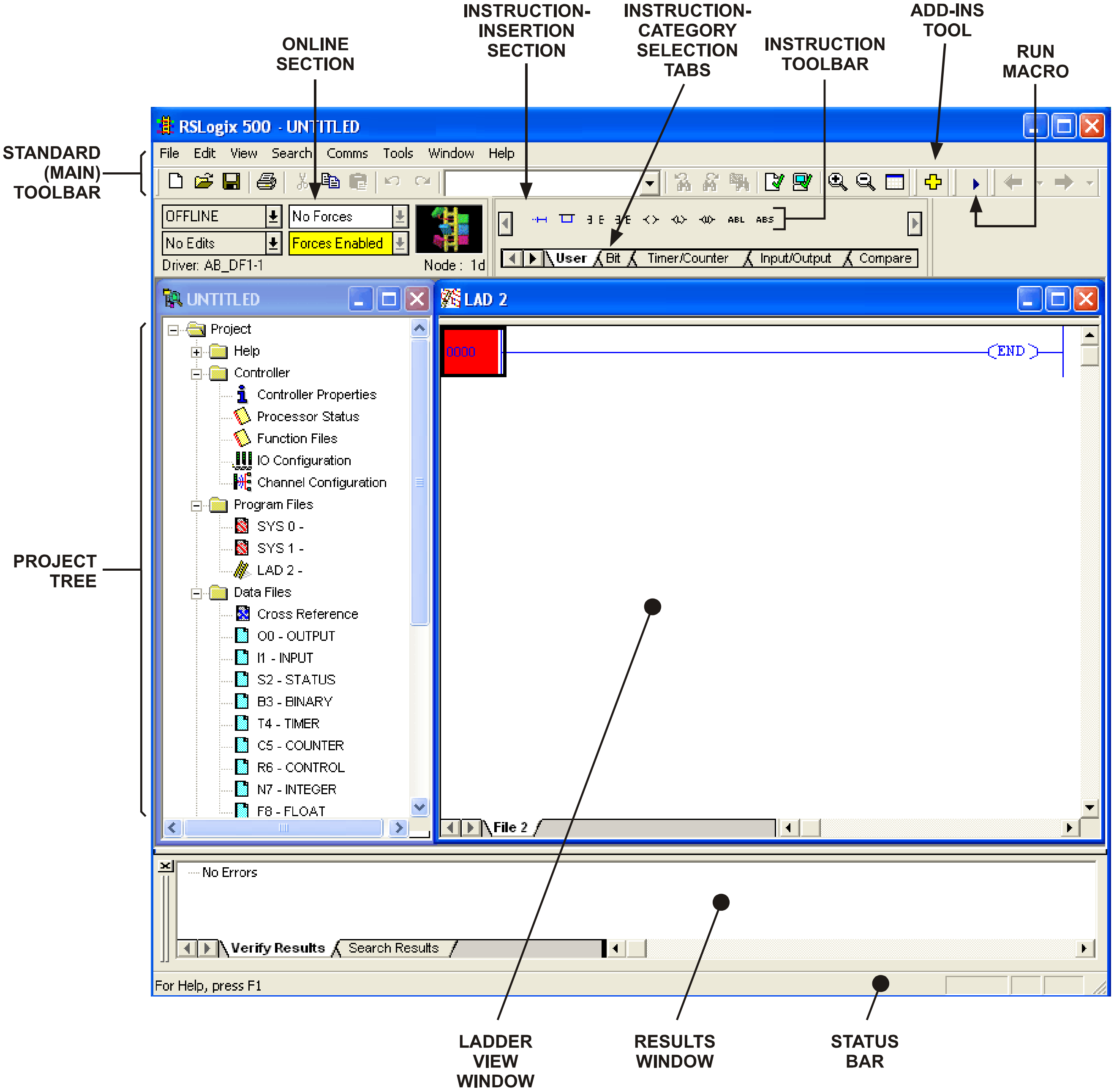
> **Note:** *The "OTE" letters which stand for "Output Energize" in Figure 2-7, do not appear within the parenthesis of the symbol for this instruction in the displayed ladder diagram.*

□ 9. You have now finished entering the instructions of rung 0 shown in the ladder program of Figure 2-7. However, rung 0 is still in the edit mode and has not been verified, since a column of e's still appears to the left of this rung.

To verify rung 0, click on "0000" at the left of rung 0 to select it, and then click the mouse right button. This opens a context-sensitive menu. Choose the **Verify Rung** command in this menu.

If no error is detected, the rung is accepted and the column of e’s is removed to indicate that rung 0 is no longer in the edit mode. Moreover, the processor type selected upon creation of project EXERC_2 appears below each instruction in rung 0 of the ladder view.

If an error is detected, an error window as shown in Figure 2-8 will appear and rung 0 remains in the edit mode. In this example, the error window indicates that the address of the first instruction in rung 0 is unconfigured. This error can be corrected by selecting the instruction, entering the correct address, and verifying the rung once again.



OFFLINE

No Forces

No Edits

Forces Enabled

Driver: AB_DF1-1

Node : 1d

- Project
  - Help
  - Controller
    - Controller Properties
    - Processor Status
    - Function Files
    - IO Configuration
    - Channel Configuration
  - Program Files
    - SYS 0 -
    - SYS 1 -
    - LAD 2 -
  - Data Files
    - Cross Reference
    - O0 - OUTPUT
    - I1 - INPUT
    - S2 - STATUS
    - B3 - BINARY
    - T4 - TIMER
    - C5 - COUNTER
    - R6 - CONTROL
    - N7 - INTEGER
    - F8 - FLOAT

0000 —————— END

No Errors

For Help, press F1

Figure 2-8. Error window that appears after a "verify rung" command that failed.

☐ 10. Click the **New Rung** button on the instruction toolbar to insert a new rung in the ladder program. The new rung is numbered 1 and the end rung becomes rung 2. A column of e’s appears next to the left power rail to indicate that rung 1 is in the edit mode.

☐ 11. You are now ready to enter the instructions of rung 1 shown in the ladder program of Figure 2-7. First, enter instructions XIC I:0/0 and OTE O:0/1 in rung 1.

☐ 12. In the ladder program of Figure 2-7, a branch is required to enter instruction XIC I:0/1 in parallel with instruction XIC I:0/0 in rung 1. To do so, click instruction XIC I:0/0 in rung 1 to select it, click the mouse right button to open the context-sensitive menu, and choose the **Insert New Branch** command.

A branch will appear in rung 1. One leg of the branch is highlighted to indicate that it can be dragged using the mouse. Drag this branch leg slightly towards instruction XIC I:0/0 in rung 1 and notice that this causes small highlighted boxes to appear on rung 1. These boxes show possible branching points. Drag the branch leg just past instruction XIC I:0/0 in rung 1 then release the mouse left button to complete the branch insertion.

☐ 13. Click the lower corner of the left leg of the branch newly created in rung 1 in order to select this leg. Once selected, the corner of the leg is highlighted.

Click the **Examine if Closed** button on the instruction toolbar to insert this instruction in the branch of rung 1. Type the following address: **I:0/1**, then click the mouse left button to enter this address.

☐ 14. All instructions of rung 1 are now entered as per Figure 2-7. Verify rung 1 and correct the errors, if any.

☐ 15. Insert a new rung in the ladder program. The new rung is numbered 2 and the end rung becomes rung 3. A column of e’s appears next to the left power rail to indicate that rung 2 is in the edit mode.

☐ 16. Enter all instructions of rung 2 shown in the ladder program of Figure 2-7.

To do so, first enter the three series input and output instructions (I:0/2, I:0/5, and O:0/2), and then enter the remaining branch instructions.

Verify rung 2 and correct the errors, if any.

The ladder program displayed in RSLogix 500 should be identical to that shown in Figure 2-7.

Documenting a Ladder Program

☐ 17. To document a ladder program, you can insert rung comments, instruction descriptions, and address descriptions.

To insert a comment in rung 0, select this rung by clicking on "0000" at the left of this rung, click the mouse right button to open the context-sensitive menu, and choose the **Edit Comment** command. This opens a dialog box where you can enter a comment.

Type the following comment: **This rung is TRUE when PLC inputs 0 AND 1 are activated.**

Click **OK** to close the dialog box. Observe that the comment has been inserted above rung 0 of the ladder view and that it is highlighted.

☐ 18. Insert the following comments in rungs 1 and 2.

Rung 1: **This rung is TRUE when PLC input 0 OR 1 is activated.**

Rung 2: **This rung is TRUE when PLC input 2, 3, OR 4 is activated, AND PLC input 5 is deactivated.**

☐ 19. Add a description to instruction XIC I:0/0 in rung 0. To do so, select this instruction, click the mouse right button to open the context-sensitive menu, and choose the **Edit Description** command. This opens a dialog box where you can enter an instruction description or an address description.

Check the **Instruction** button, then click in the edit zone of the dialog box, and type the following instruction description: **Examine if Input File Bit 0 is at logic state 1.**

Click the **OK** button to close the dialog box. Notice that the instruction description has been added over all instructions XIC I:0/0 in the main ladder program (that is, those in rungs 0 and 1).

☐ 20. Add the following instruction descriptions to instructions XIC I:0/1, OTE O:0/0, and OTE O:0/1.

XIC I:0/1: **Examine if Input File Bit 1 is at logic state 1.**

OTE O:0/0: **Energize PLC Output 0.**

OTE O:0/1: **Energize PLC Output 1.**

You can add instruction descriptions to other instructions in the main ladder program if desired.

☐ 21. Add a description to address I:0/0. To do so, select any instruction in the main ladder program which contains address I:0/0, click the mouse right button to open the context-sensitive menu, and choose the **Edit Description** command. This opens the dialog box where you can enter an instruction description or an address description.

Check the **Address** button, click in the edit zone of the dialog box, and type the following address description: **Input File Bit 0.**

Click the **OK** button to close the dialog box. Observe that the address description is not displayed in the main ladder program. This is normal because instruction XIC I:0/0 in the main ladder program is already displayed with an instruction description. However, the address description is not lost, it is stored in the cross reference report (**Cross Reference** data file).

☐ 22. Open the **Cross Reference** data file by double-clicking this file in the project tree and observe that the description of address I:0/0 is: **Input File Bit 0.**

Close the **Cross Reference** data file.

You can add other address descriptions if desired.

☐ 23. You have now finished a ladder program with comments and descriptions. Save your project as EXERC_2.

Creating and Printing a Project Report

☐ 24. To create and print a project report, choose the **Report Options** command in the **File** menu to open the **Report Options** dialog box. Select the **Configuration** folder, then make sure the following elements of the **Reports** section are selected:

- Title Page
- Processor Information
- Data File List
- Data Files
- Program File List
- Program Files

Click the **Print** button to close the **Report Options** dialog box. This also opens the **Print** dialog box. Change the print settings as desired then click the **OK** button to close the dialog box and start printing.

☐ 25. Close RSLogix 500. Turn off the computer.

CONCLUSION

In this exercise, you created a new project and learned how to enter instructions in a ladder program. You also learned how to insert rung comments, instruction descriptions, and address descriptions in a ladder program. You saw how to create and print a project report.

REVIEW QUESTIONS

1. During PLC operation, what condition is required for a rung to be evaluated as true by the processor?

   ____

   ____

2. In Figure 2-4 of this exercise, what must be the logic state of input file bits I:0/0 and I:0/1 in order for logical continuity to exist on rung 0?

   ____

   ____

3. Briefly explain how to insert a rung comment in a ladder program.

   ____

   ____

   ____

4. What is the difference between a hardwired ladder rung and a PLC ladder rung?

   ____

   ____

   ____

   ____

5. Which file must be opened to observe the address descriptions?

   ____

Exercise 7

Sequencer Instructions

EXERCISE OBJECTIVE

- To program and test PLC ladder programs that use sequencer instructions.

DISCUSSION

Introduction

PLC sequencer instructions are output instructions used to control sequential operations. They are employed in systems where devices must be turned on and off during definite periods of time, and in systems that perform a sequence of successive operations.

Sequencer Instructions of the Trainer PLC

The PLC on your trainer includes the following sequencer instructions:

- the sequencer output (**SQO**) instruction;

- the sequencer compare (**SQC**) instruction.

The SQO instruction **transfers data** from a programmed sequencer file, through a mask, to a destination file. This instruction is used for the sequential control of various devices in process or control operations.

The SQC instruction **compares the data** from a source, through a mask, against the data in a programmed sequencer file for equality. This instruction is often used to monitor machine operating conditions or for diagnostic purposes.

To understand how the SQO and SQC instructions work, one must first consider the parameters associated with these instructions. These parameters are described below.

Parameters of the Sequencer Instructions of the Trainer PLC

When entering a sequencer instruction (SQO or SQC) with the trainer PLC, the following parameters must be programmed:

- **File** (SQO, SQC): address of the sequencer file. With an SQO instruction, the sequencer file stores data to be transferred to the destination file. With an SQC instruction, the sequencer file stores reference data that is used for comparison to the data at the source address. The data contained in the sequencer file can be stored in a binary (Bx) data file or in an integer (Nx) data file.

- **Mask** (SQO, SQC): hexadecimal (h) code or the address of a word or file through which the sequencer instruction transfers (SQO) or compares (SQC) data. Mask bits that are set to logic state 0 will mask data. Mask bits that are set to logic state 1 will pass data.

- **Source** (SQC): address of the input word or file where the SQC instruction takes data for comparison to the data in its sequencer file (reference).

- **Destination** (SQO): address of the output word or file where the SQO instruction transfers data from its sequencer file.

- **Control** (SQO, SQC): 3-word register (R data file) that stores the status bits of the sequencer instruction, the length of its sequencer file, and the position (step) of the sequencer instruction within the sequencer file, as shown in Table 7-1.

Word	B_{15}	B_{14}	B_{13}	B_{12}	B_{11}	B_{10}	B_9	B_8	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
0	EN		DN		ER			FD								
1	Length of sequencer file															
2	Position															

Table 7-1. R data file structure.

Status bits (word 0 of the control data file)

The status bits provide information on the sequential process. They can be used in a ladder program to control relay-type instructions.

– Enable [EN] – (bit number 15 of word 0): The EN bit pertains to both the SQO and SQC instructions. This bit is set to logic state 1 when the rung containing the sequencer instruction is true, indicating that this instruction is true. It is set to logic state 0 when the rung is false.

– Done [DN] – (bit number 13 of word 0): The Done (DN) bit pertains to both the SQO and SQC instructions. It is set to logic state 1 when the sequencer instruction steps to the last word of data in its sequencer file, causing this word to be transferred (SQO) or compared (SQC). The DN bit is reset to logic state 0 on the next false-to-true transition of the sequencer rung.

– Error [ER] – (bit number 11 of word 0): The ER bit pertains to both the SQO and SQC instructions. The ER bit is set to logic state 1 when the processor detects a negative position value, or a negative or zero length value.

– Found [FD] – (bit number 8 of word 0): The FD bit pertains to the SQC instruction only. This bit is set to logic state 1 when all the non-masked bits in the word or file at the source address match those of the corresponding reference word in the sequencer file. When the bits do not match, the FD bit is set to logic state 0. The FD bit is updated each time the processor evaluates the SQC instruction while the rung is true.

Length of the sequencer file (word 1 of the control data file)

This is the number of steps contained in the sequencer file, starting at position 1. The sequencer instruction automatically returns (wraps) to position 1 upon completion of each cycle.

Upon startup, that is, when the PLC is switched from the Program mode to the Run mode, the sequencer instruction is set to **position 0** of its sequencer file. If the rung containing the sequencer instruction is **true**, this instruction transfers (SQO) or compares (SQC) the data present at position 0 of the sequencer file. If the rung containing the sequencer instruction is **false** upon startup, this instruction waits until the rung becomes true and then transfers (SQO) or compares (SQC) the data present at **position 1** of the sequencer file.

Position (word 2 of the control data file)

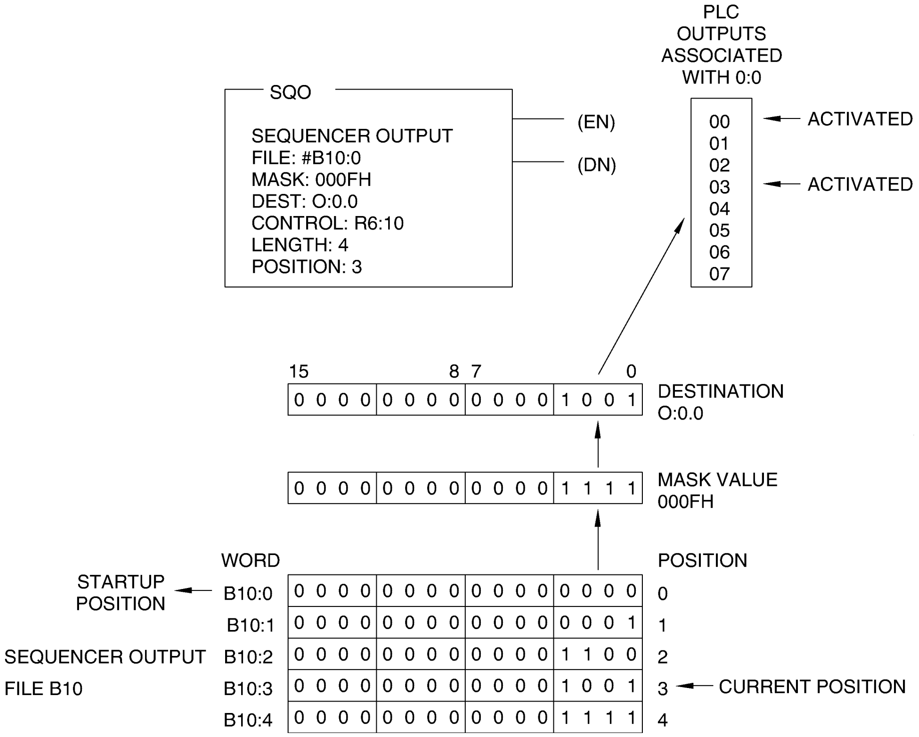
This is the position (step) where the sequencer instruction currently is within its sequencer file.

The Reset (RES) instruction can be used to reset a sequencer. To do so, the Reset instruction must be programmed with the address of the R data file (control register) of the sequencer instruction. When made true, the Reset instruction resets the sequencer instruction to **position zero** of its sequencer file, and it resets all the sequencer status bits (except the FD bit) to logic state 0.

Operation of the SQO Instruction

Figure 7-1 shows how the SQO instruction works. When the rung containing the SQO instruction goes from false to true, this instruction steps to the next position in its sequencer file (file B10). The bits in the sequencer file are programmed to control PLC outputs 0 through 3. They are transferred, through the mask value (000FH), to the destination address (O:0.0).

In Figure 7-1, for example, the SQO instruction is at position 3 of its sequencer file. Consequently, the word of data stored at address B10:3 is transferred to destination address O:0.0. Since bits 0 through 3 of the mask value are set to logic state 1, bits 0 through 3 of the word of data (1001) are allowed to pass to destination address O:0.0. This causes the output data file bits at addresses O:0/0 and O:0/3 to be set to logic state 1, causing PLC outputs 0 and 3 to be activated.



	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1	DESTINATION O:0.0
	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	MASK VALUE 000FH
B10:0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
B10:1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	1
B10:2	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0	2
B10:3	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1	3
B10:4	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	4

Figure 7-1. Operation of the SQO instruction.

When the rung containing the SQO instruction becomes false, this instruction remains at position 3 of its sequencer file, so that PLC outputs 0 and 3 remains activated. When the rung becomes true again, the SQO instruction steps to position 4 of its sequencer file, which is the last position of this file. This causes the word of data stored at address B10:4 to be transferred to destination address O:0.0 through the mask value. This in turn causes PLC outputs 0 through 3 to become all activated.

On the next false-to-true transition of the rung containing the SQO instruction, this instruction automatically returns to position 1 of its sequencer file to start a new cycle. This causes the word of data stored at address B10:1 to be transferred to destination address O:0.0 through the mask value. This in turn causes PLC output 0 to be activated, and PLC outputs 1, 2, and 3 to be deactivated.

If the Reset instruction associated with the SQO instruction is made true at some point of the cycle, the SQO instruction will automatically return to position 0 of its sequencer file, and the following will occur:

- If the rung of the SQO instruction is true at the moment when the Reset instruction is made true, the word of data stored at position 1 of sequencer file B10 will be transferred to destination address O:0.0. This same word will be transferred on the next false-to-true rung transition, as the SQO instruction will step to position 1 of its sequencer file.

- If the rung of the SQO instruction is false at the moment when the Reset instruction is made true, the PLC output status will remain unchanged. When this rung becomes true again, the SQO instruction will step to position 1 of sequencer file B10, causing the word of data stored at this position to be transferred to destination address O:0.0.

Operation of the SQC Instruction

Figure 7-2 shows how the SQC instruction works. When the rung containing the SQC instruction goes from false to true, this instruction steps to the next position in its sequencer file (file B12). The bits in the sequencer file are programmed to monitor PLC inputs 0 through 3. They are compared, through the mask value (000FH), against the bits at the source address (I:0.0) for equality. When all the non-masked bits at the source address match the bits in the reference word of the sequencer file, the Found (FD) bit of the sequencer instruction is set to logic state 1. When the bits do not match, the FD bit is set to logic state 0.

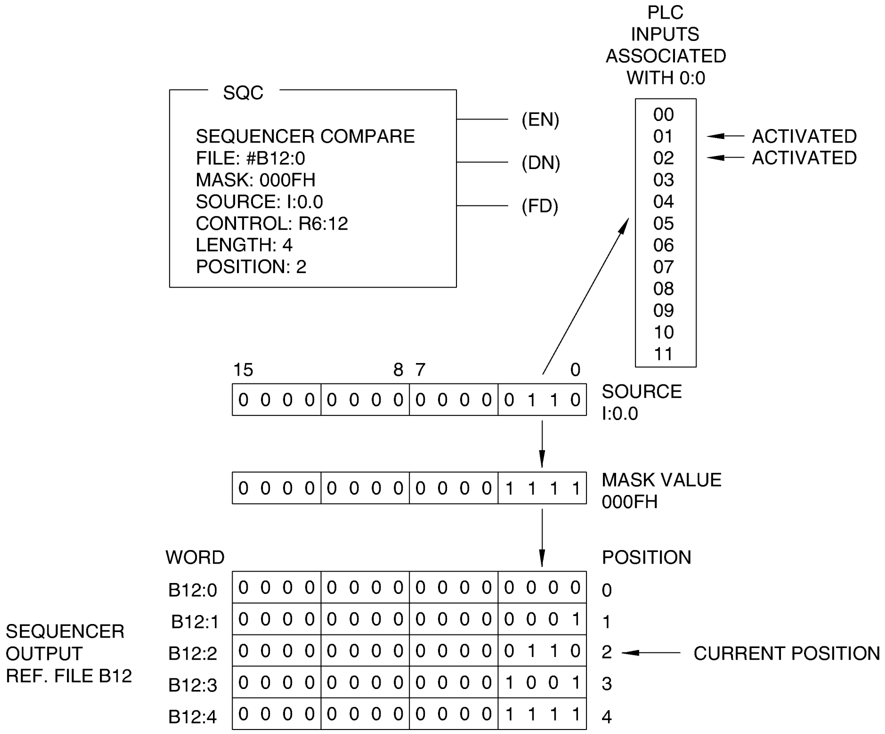


Figure 7-2. Operation of the SQC instruction.

In Figure 7-2, for example, the SQC instruction is at position 2 of its sequencer file. Consequently, the word of data stored at address B12:2 is compared, through the mask value, against the bits at source address I:0.0. Since PLC inputs 1 and 2 are activated, the bits at addresses I:0/1 and I:0/2 are set to logic state 1. Since bits 0 through 3 of the mask value are set to logic state 1, bits 0 through 3 at source address I:0.0 are non-masked. Consequently, all the bits at source address I:0.0

match those of the reference word at address B12:2, causing the FD bit to be set to logic state 1.

During all the time that the rung of the SQC sequencer remains true, the comparison is performed for every scan of the processor. This implies that, if PLC input 1 or 2 becomes deactivated while the rung is still true, the FD bit will be reset to logic state 0.

When the rung containing the SQC instruction becomes false, the FD bit is automatically reset to logic state 0. When the rung becomes true again, the SQC instruction steps to position 3 of its sequencer file and compares the word of data stored at address B12:3, through the mask value, against the bits at source address I:0.0. If PLC inputs 0 and 3 are activated, the FD will be set to logic state 1, otherwise it will be set to logic state 0.

On the next false-to-true transition of the rung containing the SQC instruction, the SQC instruction steps to position 4 and compares the word of data stored at address B12:4, through the mask value, against the bits at source address I:0.0. If PLC inputs 0 through 3 are all activated, the FD will be set to logic state 1, otherwise it will be set to logic state 0.

On the next false-to-true transition of the rung containing the SQC instruction, this instruction automatically returns to position 1 of its sequencer file to start a new cycle.

Procedure Summary

In this exercise, you will program and test two ladder programs: one that uses a sequencer output (SQO) instruction, and one that uses a sequencer compare (SQC) instruction. This will allow you to understand how each of these instructions and their status bits work. You will then use the acquired knowledge to create and test your own ladder program.

> **Note:** *As earlier mentioned, the manual applies specifically to PLC Trainer Model 3240-4. If you are using another model, follow the exercise procedure by adapting it as described below.*
>
> - *Model 3240-A or Model 3240-3: Same way as Model 3240-4.*
>
> - *Model 3270-4: Connect an external 24-VDC source to the supply jacks intended for this purpose at the left bottom of the trainer front panel. When asked to activate a PLC input, connect the corresponding PLC input jack to the positive terminal of the 24-VDC source, using a connection lead. When asked to deactivate a PLC input, remove the lead between the corresponding PLC input jack and the 24-VDC source. To determine whether or not a PLC output is activated, observe the PLC output status indicators on the PLC module.*

- *Model 9066, connect the DC COM jacks (2) to the DC SUPPLY OUTPUT negative terminal. To activate a PLC input, connect the corresponding PLC input jack to the positive terminal of the DC SUPPLY OUTPUT. To determine whether or not a PLC output is activated, you can see it on the LCD display default page (if not on this page, press ESC and select I/O status). To act on a physical device, connect VAC/VDC jacks to the positive terminal of the DC SUPPLY OUTPUT if you are are using the two first outputs or connect DC 24 + and DC 24 V - to the corresponding terminals of the DC SUPPLY OUTPUT to use last four outputs.*

EQUIPMENT REQUIRED

Refer to the Equipment Utilization Chart, in Appendix A of this manual, to obtain the list of equipment required to perform this exercise.

PROCEDURE

Setting Up the Equipment

☐ 1. Connect the RS-232 serial port of the computer station to the communications port of the PLC on the PLC Trainer, using a 1761-CBL-PM02 cable.

☐ 2. Turn on the computer and start RSLogix 500.

Turn on the PLC Trainer.

☐ 3. In this exercise, you will study the operation of PLC instructions through observation of the ladder program view, the data files of the processor, as well as the status of the lamps next to the PLC output jacks on the trainer front panel.

Note: *Since trainer models 3270-4 and 9066 do not have output lamps, observe the PLC output status LED's on the PLC module instead.*

The SQO Instruction

☐ 4. Create a new project having the following processor name: EXERC_7.

The project tree of processor EXERC_7 and program file LAD 2 should be displayed in the RSLogix 500 window. Program file LAD 2 contains the main ladder program.

The next steps of this procedure consist in entering the ladder program of Figure 7-3.

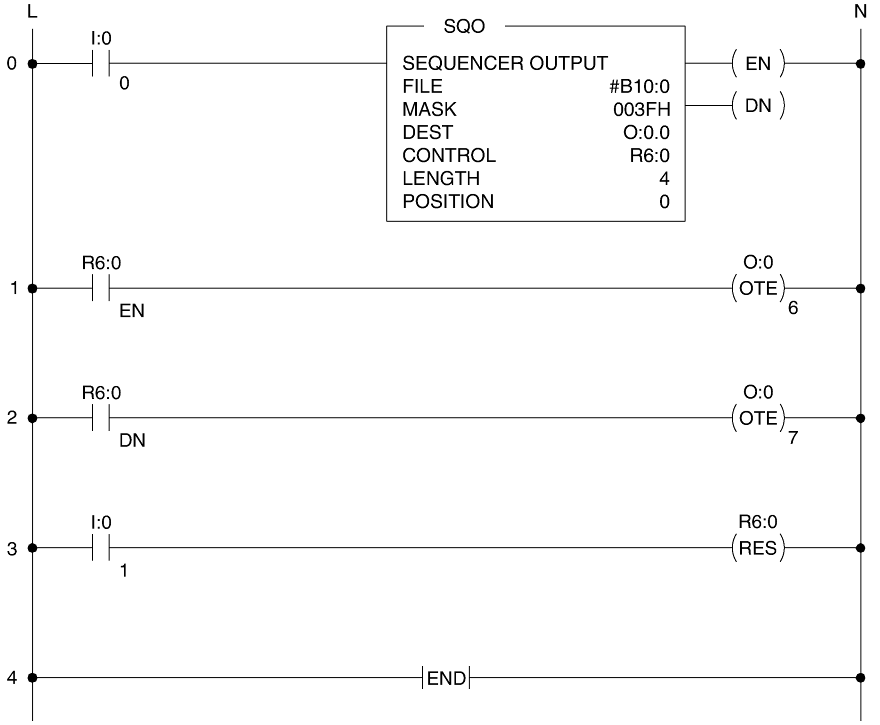


Figure 7-3. The sequencer output (SQO) instruction.

☐ 5. Select the **User** category of instructions by clicking the corresponding instruction category selection tab.

Insert a new rung into program file LAD 2. In this rung, enter instruction XIC I:0/0. Then, enter instruction SQO R6:0 by performing the following steps:

– Select the **File Shift/Sequencer** category of instructions by clicking the corresponding instruction category selection tab.
– Click the **Sequencer Output** button on the instruction toolbar to insert this instruction in the rung, type: **B10:0**, then press the mouse left button to accept this address. (The file indicator "#" will automatically be added to the front of this address).
– Double-click **Mask** within the SQO instruction, type: **003FH**, then press the mouse left button to accept value.
– Double-click **Dest** within the SQO instruction, type: **O:0.0**, then press the mouse left button to accept this address.
– Double-click **Control** within the SQO instruction, type: **R6:0**, then press the mouse left button to accept this address.
– Double-click **Length** within the SQO instruction, type: **4**, then press the mouse left button to accept this value.
– Leave the **Position** value of the SQO instruction set to **0**. This completes the parameter setting for instruction SQO R6:0.

You have now finished entering the instructions of rung 0 in the ladder program of Figure 7-3.

Note: *The letter "H" that you entered after the mask value 003F specifies that this value is a hexadecimal code. This hexadecimal code corresponds to the following binary code: 0000 0000 0011 1111.*

☐ 6. Select the **User** category of instructions by clicking the corresponding instruction category selection tab.

Enter a new rung into program file LAD 2. In this rung, enter instruction XIC R6:0/EN. To do so, click the **Examine If Closed** button on the instruction toolbar, type: **R6:0/EN**, then click the mouse left button to enter this address.

Enter instruction OTE O:0/6.

You have now finished entering the instructions of rung 1 in the ladder program of Figure 7-3.

☐ 7. Enter a new rung into program file LAD 2. In this rung, enter instruction XIC R6:0/DN, then enter instruction OTE O:0/7.

You have now finished entering the instructions of rung 2 in the ladder program of Figure 7-3.

☐ 8. Enter a new rung into program file LAD 2. In this rung, enter instruction XIC I:0/1. Then, enter instruction RES R6:0. To do so, select the Timer/Counter category of instructions by clicking the corresponding instruction category selection tab. Click the **Reset** button on the instruction toolbar to insert this instruction in the rung, type: **R6:0**, then press the mouse left button to accept this address.

You have now finished entering the instructions of rung 3 in the ladder program of Figure 7-3.

☐ 9. Using the **Verify Rung** command, verify the rungs that have been edited. Correct the errors, if any.

The main ladder program in RSLogix 500 should be identical to that shown in Figure 7-3.

☐ 10. In the **Data Files** folder of processor EXERC_7, observe that a new file named B10 appears below File F8 - FLOAT. This file has been created because you entered B10:0 as the **File** address for instruction SQO R6:0. Data file **B10** is the sequencer file for this instruction.

Open data file **B10**. Observe that the corresponding window displays the logic state of the bits in the words at addresses B10:0 through B10:4. Each word corresponds to a sequencer position.

Enter 1’s at the proper bit locations of data file **B10** so that it contains the same data as Table 7-2 below. When you have finished, close data file **B10**.

> **Note:** *To enter a 1 at a bit location, double-click the bit location, type: 1 and then press the* ***Enter*** *key.*

SEQUENCER POSITION	WORD	BINARY DATA			
		15	8	7	0
0	B10:0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	B10:1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
2	B10:2	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 1
3	B10:3	0 0 0 0	0 0 0 0	0 0 1 0	1 0 1 0
4	B10:4	0 0 0 0	0 0 0 0	0 0 1 1	1 1 1 1

Table 7-2. Data table for instruction SQO R6:0.

☐ 11. Save the project in a project file named EXERC_7.RSS.

☐ 12. Make sure the system communications are properly configured.

☐ 13. Download project EXERC_7 to the PLC.

Go online and place the PLC in the Run mode.

☐ 14. Open data file **B10** to display the logic state of the bits in the words at addresses B10:0 through B10:4.

Place the **B10** data file window at the bottom of the RSLogix 500 window.

☐ 15. Open data file **R6 - CONTROL**. Observe that the corresponding window displays the status bits of instruction SQO R6:0 (bits EN, EU, DN, ...), the length of sequencer file B10 (4), and the position where instruction SQO R6:0 is within sequencer file B10 (position 0).

What is the logic state of the EN status bit of instruction SQO R6:0? Why?

____

____

What is the current position of instruction SQO R6:0 within sequencer file B10? Why?

Place the **R6 - CONTROL** data file window at the bottom of the RSLogix 500 window.

□ 16. Activate PLC input 0, using one of the toggle switches of the trainer. What happens to the EN status bit of instruction SQO R6:0 and to PLC output 6? Why?

What is the current position of instruction SQO R6:0 within sequencer file B10, according to the status of PLC output lamps 0 and 1? Why?

□ 17. Which bits in the word at address B10:1 are set to logic state 1?

□ 18. According to the current status of PLC outputs 0 and 1, is the word of data at address B10:1 transferred to destination address O:0.0, through the mask value of instruction SQO R6:0? Explain.

□ 19. Deactivate PLC input 0. What happens to the EN status bit of instruction SQO R6:0 and to PLC output 6? Why?

Is instruction SQO R6:0 still at position 1 of sequencer file B10, causing PLC outputs 0 and 1 to be activated?

□ Yes □ No

□ 20. Activate PLC input 0. What is the current position of instruction SQO R6:0 within sequencer file B10, according to the status of the PLC output lamps? Why?

□ 21. Which bits in the word at address B10:2 are set to logic state 1?

□ 22. According to the current status of PLC outputs 0, 1, and 2, is the word of data at address B10:2 transferred to destination address O:0.0, through the mask? Explain.

□ 23. Deactivate and then activate PLC input 0. Are PLC outputs 1, 3, and 5 activated? Why?

☐ 24. Deactivate and then activate PLC input 0. What happens to the DN status bit of instruction SQO R6:0 and to PLC output 7? Why?

____

____

____

Are PLC outputs 0 through 5 activated? Why?

____

____

____

☐ 25. Deactivate and then activate PLC input 0. What happens to the DN status bit of instruction SQO R6:0 and to PLC output 7? Why?

____

____

____

____

Are PLC outputs 0 and 1 activated, indicating that instruction SQO R6:0 has returned to position 1 of sequencer file B10 to initiate a new cycle?

☐ Yes ☐ No

☐ 26. Deactivate and then activate PLC input 0. Instruction SQO R6:0 is now at position 2 of sequencer file B10, causing PLC outputs 0, 1, and 2 to be activated.

Deactivate and then activate PLC input 0. Instruction SQO R6:0 is now at position 3 of sequencer file B10, causing PLC outputs 1, 3, and 5 to be activated.

To reset the SQO instruction, activate PLC input 1, using another toggle switch of the trainer. Are PLC outputs 0 and 1 activated? Why?

____

____

____

____

____

☐ 27. Deactivate PLC input 1. Does this cause instruction SQO R6:0 to step to position 1 of sequencer file B10, causing PLC outputs 0 and 1 to remain activated?

☐ Yes ☐ No

☐ 28. If the mask value of instruction SQO R6:0 were changed to "00F0H", what would be the status of PLC outputs 0 through 5 when this instruction is at positions 1, 2, 3, and 4 of sequencer file B10? Explain.

____

____

____

____

☐ 29. On the PLC Trainer, make sure that all PLC inputs are deactivated (remove all the switch connection leads and place all the switch toggles downward).

☐ 30. Place the PLC in the **Program** mode and go offline.

The SQC Instruction

☐ 31. Modify the existing main ladder program of project EXERC_7 as indicated below in order to obtain the ladder program shown in Figure 7-4:

a. In rung 0, double-click the address of instruction XIC I:0/0, type the following address: **I:0/8**, then press the mouse left button to accept the new address.

b. In rung 0, replace instruction SQO R6:0 with a SQC (sequencer compare) instruction having the same address. To do so, select instruction SQO R6:0 and choose the **Change Instruction Type** command in the context-sensitive menu. Type **SQC** using the keyboard and press the **Enter** key.

Modify the parameter setting of the SQC instruction as follows:

– Double-click **File**, type **B12:0**, then press the mouse left button to accept this address.
– Double-click **Mask**, type: **00FFH**, then press the mouse left button to accept this value.
– Double-click **Source**, type: **I:0.0**, then press the mouse left button to accept this value.
– Leave the **Control** parameter set to **R6:0**, the **Length** parameter set to **4**, and the **Positio**n parameter set to **0**.

c. In rung 1, double-click the address of instruction OTE O:0/6, type the following address: **O:0/0**, then press the mouse left button to accept the new address.

d. In rung 2, double-click the address of instruction OTE O:0/7, type the following address: **O:0/1**, then press the mouse left button to accept the new address.

e. Insert a new rung below rung 2. In the newly created rung (rung 3), enter instruction XIC R6:0/FD. Then, enter instruction OTE O:0/2.

f. In rung 4, double-click the address of instruction XIC I:0/1, type the following address: **I:0/9**, then press the mouse left button to accept the new address.

Using the **Verify Rung** command, verify the rungs that have been edited. Correct any errors.

The main ladder program in RSLogix 500 should be identical to that shown in Figure 7-4.

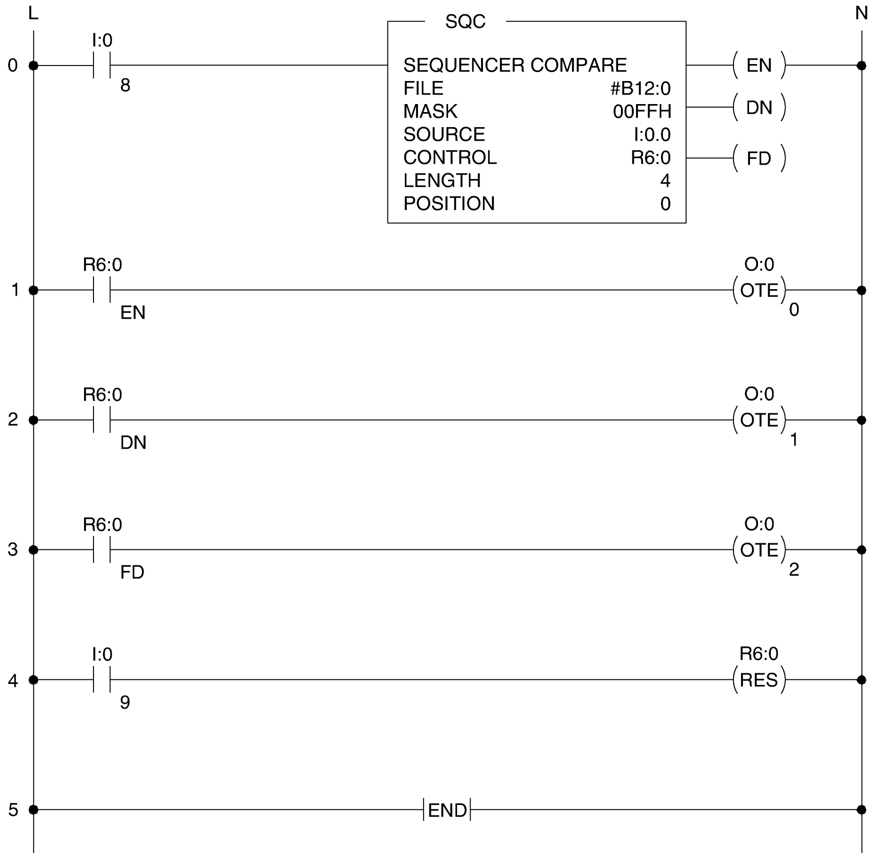


Figure 7-4. The SQC instruction.

□ 32. In the **Data Files** folder of processor EXERC_7, observe that a new file named **B12** appears below File B10. File B12 has been created because you entered B12:0 as the **File** address for instruction SQC R6:0. File B12 is the sequencer file for this instruction.

Open data file **B12**. Observe that the corresponding window displays the logic state of the bits in the words at addresses B12:0 through B12:4. Each word corresponds to a sequencer position.

Enter 1's at the proper bit locations of data file **B12** so that it contains the same data as Table 7-3 below. When you have finished, close data file **B12**.

SEQUENCER POSITION	WORD	BINARY DATA			
		15	8	7	0
0	B12:0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	B12:1	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 1
2	B12:2	0 0 0 0	0 0 0 0	0 1 0 1	0 1 0 1
3	B12:3	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1
4	B12:4	0 0 0 0	0 0 0 0	1 0 1 0	1 0 1 0

Table 7-3. Data table for instruction SQC R6:0.

□ 33. Using the **Save As** command in the **File** menu, save the new main ladder program in a project file named EXERC_7a.RSS.

> **Note:** *Do not forget to change the processor name while you are in the Save Program As dialog box.*

□ 34. Skip this step if you are not using PLC Trainer Model 3240-4.

On the PLC Trainer Model 3240-4, connect switches 1, 4, 5, 6, and 7 of the front panel to the 24-VDC PLC input jacks 0 through 9 as indicated in Table 7-4. Figure 7-5 shows the detail of the switch connections to make. This will allow you to activate PLC inputs 0 through 9 for testing the newly entered project (EXERC_7a).

TRAINER SWITCH	CONNECT TO PLC INPUT JACK(S)
1	9
4	8
5	0 AND 2
6	4 AND 6
7	1, 3, 5, AND 7

Table 7-4. Trainer switch connections to make.

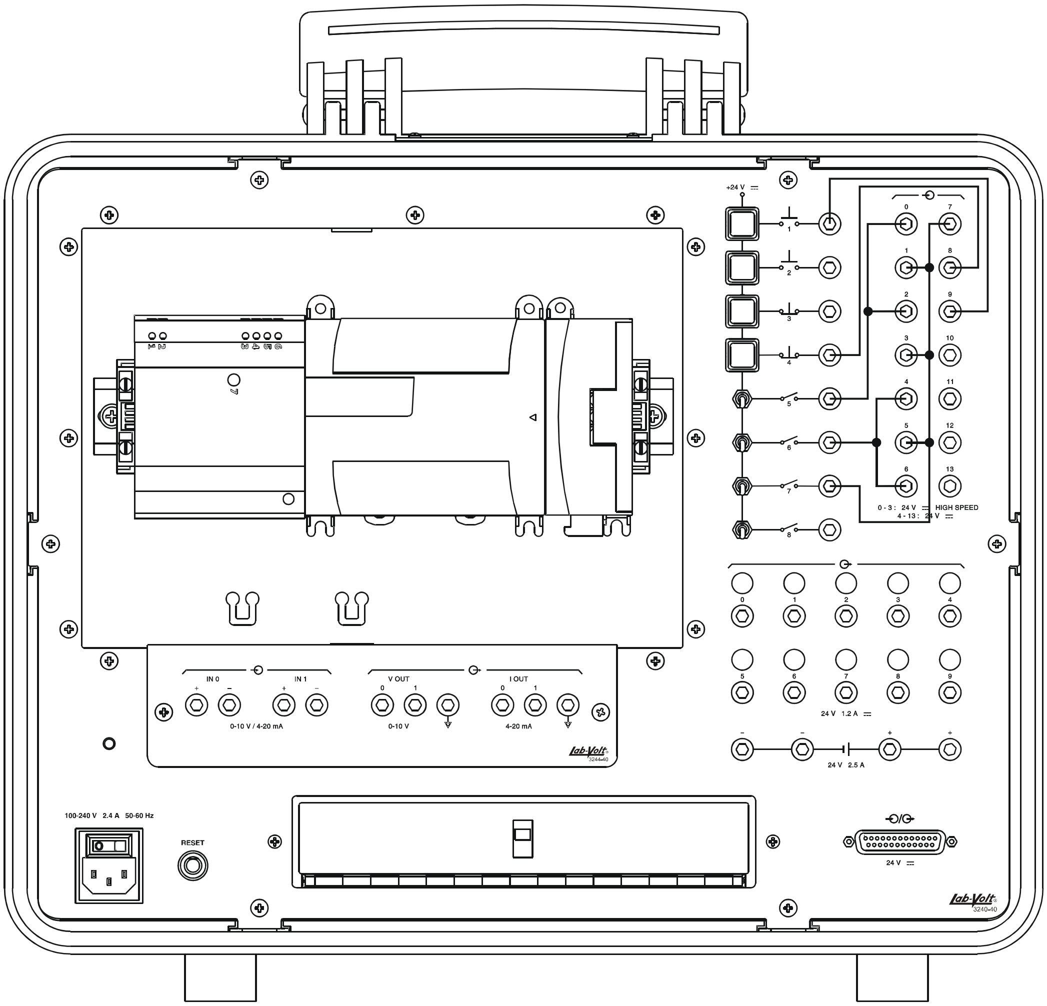


Figure 7-5. Trainer switch connections to make.

☐ 35. Download project EXERC_7a to the PLC of the PLC Trainer. Go online and place the PLC in the **Run** mode.

☐ 36. Open data file **B12** to display the logic state of the bits in the words at addresses B12:0 through B12:4.

Place the **B12** data file window at the bottom of the RSLogix 500 window.

☐ 37. Open data file **R6 - CONTROL**. Observe that the corresponding window displays the status bits of instruction SQC R6:0, the length of sequencer file B12, and the position where instruction SQC R6:0 is within sequencer file B12.

What is the logic state of the EN status bit of instruction SQC R6:0 ? Why?

What is the current position of instruction SQC R6:0 within sequencer file B12? Why?

☐ 38. Activate PLC input 8. To do so, set the toggle of trainer switch 4 upward. What happens to the EN status bit of instruction SQC R6:0 and to PLC output 0? Why?

What is the current position of instruction SQC R6:0 within sequencer file B12? Why?

☐ 39. Which bits in the word at address B12:1 are set to logic state 1?

☐ 40. Activate PLC inputs 0 and 2. To do so, set the toggle of trainer switch 5 upward. What happens to the FD status bit of instruction SQC R6:0 and to PLC output 2? Why?

☐ 41. Deactivate PLC input 8. To do so, set the toggle of trainer switch 4 downward. What happens to the EN and FD status bits of instruction SQC R6:0, and to PLC outputs 0 and 2? Why?

____

____

____

____

____

____

☐ 42. Activate PLC input 8. To do so, set the toggle of trainer switch 4 upward. Does this cause instruction SQC R6:0 to step to position 2 in sequencer file B12? Why?

____

____

☐ 43. Leave PLC inputs 0 and 2 activated (toggle of trainer switch 5 in the upward position). Activate PLC inputs 4 and 6 by setting the toggle of trainer switch 6 upward. Does the FD status bit of instruction SQC R6:0 go to logic state 1? Why?

____

____

____

____

☐ 44. Deactivate PLC inputs 4 and 6 by setting the toggle of trainer switch 6 downward. What happens to the EN and FD status bits of instruction SQC R6:0? Why?

____

____

____

□ 45. Deactivate and then activate PLC input 8 by setting the toggle of trainer switch 4 downward and then upward. Instruction SQC R6:0 is now at position 3 of sequencer file B12. Which PLC inputs must now be activated in order for the FD status bit to be set to logic state 1? Explain why and verify your answer by activating the proper PLC inputs (toggles of trainer switches 5, 6, and 7 upward).

____

____

____

□ 46. Again deactivate and then activate PLC input 8. What happens to the DN status bit of instruction SQC R6:0 and to PLC output 1? Why?

____

____

____

____

Which PLC inputs must now be activated in order for the FD status bit to be set to logic state 1? Explain why and verify your answer by activating the proper PLC inputs (toggles of trainer switches 5 and 6 downward, toggle of trainer switch 7 upward).

____

____

____

□ 47. Deactivate and then activate PLC input 8. What happens to the sequencer position? To the DN status bit of instruction SQC R6:0? Explain.

____

____

____

□ 48. Deactivate and activate PLC input 8 until instruction SQC R6:0 is at position 3 of sequencer file B12.

Activate PLC inputs 0 through 7 to make the FD status bit of instruction SQC R6:0 go to logic state 1 (toggles of trainer switches 5, 6, and 7 upward).

Activate PLC input 9. To do so, press the pushbutton of trainer switch 1 and keep it pressed. What happens to the sequencer position? To the FD status bit? Why?

____

____

____

□ 49. Deactivate PLC input 9 by releasing the pushbutton of trainer switch 1. Does this cause instruction SQC R6:0 to step to position 1 of sequencer file B12?

□ Yes □ No

□ 50. If the mask value of instruction SQC R6:0 were changed to "00F0H", what PLC inputs would you need to activate in order for the FD status bit of instruction SQC R6:0 to be set to logic state 1 when this instruction is at positions 1, 2, 3, and 4 of sequencer file B12? Explain.

____

____

____

____

□ 51. On the PLC Trainer, make sure that all PLC inputs are deactivated (remove all the switch connection leads and place all the switch toggles downward).

□ 52. Place the PLC in the **Program** mode. Go offline and close project EXERC_7a.RSS.

Creating a New Ladder Program

□ 53. Create (on paper) a ladder diagram that will control the activation of PLC outputs 0 through 7 in the following way:

– Upon startup, all the PLC outputs are deactivated.
– When PLC input 0 is activated, PLC output 0 is activated (step 1);
– 5 seconds later, PLC outputs 0 through 3 are activated (step 2);
– 5 seconds later, PLC outputs 0 through 3 are deactivated, while PLC outputs 4 through 7 are activated (step 3);
– 5 seconds later, PLC outputs 0 through 7 are activated (step 4);
– 5 seconds later, the cycle automatically repeats, starting from step 1, causing PLC output 0 to be activated.
– If PLC input 0 is deactivated, the PLC outputs that are activated at that moment remain activated. When PLC input 0 is reactivated, the cycle resumes from the step following the one at which it was interrupted.

Hints: use a sequencer output (SQO) instruction and a timer-on-delay (TON) instruction in your program. Use timer status bits to create transitions on the rung containing the SQO instruction and to reset the TON instruction.

☐ 54. Create a new project having the following processor name: EXERC_7b. The project tree of processor EXERC_7b and program file LAD 2 should be displayed in the RSLogix 500 window.

Enter your ladder program in program file LAD 2.

Verify each rung, then save the project in a project file named EXERC_7b.RSS.

Download project EXERC_7b to the PLC of the PLC Trainer.

Go online and place the PLC in the **Run** mode.

☐ 55. Test program operation and, if required, modify your program so that it operates properly. Once the program has been found operational, have the instructor check your work.

☐ 56. When you have finished, place the PLC in the **Program** mode and clear the PLC memory.

☐ 57. Close RSLogix 500. Turn off the computer.

☐ 58. On the PLC Trainer, make sure that all PLC inputs are deactivated. Turn off the PLC Trainer. Remove all the switch connection leads, set all the switch toggles downward, and return all the equipment.

CONCLUSION

In this exercise, you familiarized yourself with the following sequencer instructions of the trainer PLC: the sequencer output (SQO) instruction and the sequencer compare (SQC) instruction. You learned that these instructions have an R6 (control) file, as well as a sequencer file that stores data to be transferred (SQO) or compared (SQC)

– The SQO instruction transfers data from its sequencer file, through a mask, to a destination file on each false-to-true rung transition. Once the SQO instruction has reached the last position in its sequencer file, it automatically returns to position 1 on the next false-to-true rung transition.

– The SQC instruction compares the data from a source file, through a mask, against the data in its sequencer file for equality. If there is equality, the Found (FD) status bit is set to logic state 1. When a false-to-true rung transition occurs, the SQC instruction steps to the next position in its sequencer file. Once the

SQO instruction has reached the last position in its sequencer file, it automatically returns to position 1 on the next false-to-true rung transition.

REVIEW QUESTIONS

1. What is the sequencer file of an SQO or SQC instruction?

____

____

____

2. What is the mask value of an SQO or SQC instruction?

____

____

____

3. Briefly describe how an SQO instruction works.

____

____

____

____

4. When is the Found (FD) bit associated with the SQC instruction set to logic state 1?

____

____

____

____

5. Once an SQO instruction has reached the last position in its sequencer file, what happens to the sequencer position and to the Done (DN) bit on the next false-to-true transition of this instruction's rung?

____

____

Sample

Extracted from

Instructor Guide

EXERCISE 1 FAMILIARIZATION WITH THE PLC TRAINER AND RSLOGIX 500

ANSWERS TO PROCEDURE QUESTIONS

☐ 9. File LAD 2 in the Program Files folder contains the main ladder program.

☐ 12. The displayed list of buttons in the instruction toolbar changes to reflect the newly selected category of instructions.

ANSWERS TO REVIEW QUESTIONS

1. PLC’s permit the following hardware control devices to be replaced: relays, timers, counters, and drum controllers.

2. Before a ladder logic program can be edited, a project must be created or opened.

3. The project tree.

4. Program files SYS 0, SYS 1, and LAD 2.

5. Program file LAD 2.

EXERCISE 2 PROGRAMING BASICS

ANSWERS TO REVIEW QUESTIONS

1. A continuous path of true instructions must exist between the line and neutral power rails. For example, all instructions on a series logic rung must be true.

2. Input file bit I:0/0 must be at logic state 1 and input file bit I:0/1 must be at logic state 0.

3. To insert a rung comment, select a rung, click the mouse right button to open the context-sensitive menu, choose the **Edit Comment** command, type the comment in the dialog box, and click the mouse left button while the mouse pointer is outside the dialog box.

4. With a hardwired ladder rung, electrical continuity is required on the rung in order for the output device on this rung to be energized.

   With a PLC ladder rung, logical continuity is required on the rung in order for the output instruction on this rung to be true.

5. The Cross Reference data file.

EXERCISE 3 ONLINE OPERATIONS

ANSWERS TO PROCEDURE QUESTIONS

□ 8. The PLC mode of operation must be changed to **Run**.

□ 13. Input file bit 0 returns to logic state 0, while instructions XIC I:0/0 in rungs 0 and 1 of the ladder program are dehighlighted. This occurs because the voltage is removed from terminal 0 of the PLC input interface, thereby deactivating PLC input 0.

□ 14.

PLC INPUT 0	PLC INPUT 1	OTE O:0/0
DEACTIVATED	DEACTIVATED	FALSE
ACTIVATED	DEACTIVATED	FALSE
DEACTIVATED	ACTIVATED	FALSE
ACTIVATED	ACTIVATED	TRUE

Table 3-5. Truth table of rung 0 of the main ladder program.

□ 15.

PLC INPUT 0	PLC INPUT 1	OTE O:0/1
DEACTIVATED	DEACTIVATED	FALSE
ACTIVATED	DEACTIVATED	TRUE
DEACTIVATED	ACTIVATED	TRUE
ACTIVATED	ACTIVATED	TRUE

Table 3-6. Truth table of rung 1 of the main ladder program.

□ 16. No, since instruction XIO I:0/5 is false, due to PLC input 5 being activated.

2. Counter instructions are used in conjunction with Reset instructions to permit resetting of their accumulated value to zero. This is necessary because counter instructions are retentive instructions that continue to increase or decrease their accumulated value once the preset value has been reached.

3. The CTU instruction increases its accumulated value by one count on each false-to-true transition of the rung in which it is contained. When the accumulated value becomes equal to the preset value, the Done (DN) bit is set to logic state 1. When the associated Reset instruction is made true, the accumulated value is reset to zero and the DN bit is reset to logic state 0.

4. False. The CTD instruction decreases its accumulated value by one count on each false-to-true rung transition. When the accumulated value becomes lower than the preset value, the DN bit is set to logic state 0. When the associated Reset instruction is made true, the accumulated value is reset to zero and the DN bit is reset to logic state 1.

5. The Done (DN) bit is set to logic state 0 when the accumulated value of the CTU or CTD instruction is lower than the preset value. It is set to logic state 1 when the accumulated value of the CTU or CTD instruction is equal to or greater than the preset value.

EXERCISE 7 SEQUENCER INSTRUCTIONS

ANSWERS TO PROCEDURE QUESTIONS

□ 15. The EN status bit is at logic state 0 because the rung containing instruction SQO R6:0 (rung 0) is false.

Instruction SQO R6:0 is at position 0 of sequencer file B10. This occurs because immediately after startup, that is, when the PLC is switched from the Program mode to the Run mode, the SQO instruction is set to position 0 of its sequencer file.

□ 16. When PLC input 0 is activated, the EN status bit of instruction SQO R6:0 is set to logic state 1, since the rung containing this instruction (rung 0) becomes true. This causes instruction XIC R6:0/EN in rung 1 to become true, making instruction OTE O:0/6 in this rung true. This causes PLC output 6 to become activated.

Since PLC output lamps 0 and 1 are on, instruction SQO R6:0 is at position 1 of sequencer file B10. This occurs because the rung containing this instruction (rung 0) has gone from false to true, causing this instruction to step to the next position in sequencer file B10.

□ 17. Bits B10:1/0 and B10:1/1 are set to logic state 1.

□ 18. Yes, since PLC outputs 0 and 1 are activated. This occurs because the mask value of instruction SQO R6:0 is set to 003FH (0000 0000 0011 1111 in binary), thereby allowing bits 0 through 5 of the word at address B10:1 to pass to destination address O:0.0. Since bits 0 and 1 in this word are set to logic state 1, the bits at addresses O:0/0 and O:0/1 are set to logic state 1, causing PLC outputs 0 and 1 to be activated.

□ 19. When PLC input 0 is deactivated, the EN status bit of instruction SQO R6:0 is set to logic state 0, since the rung containing this instruction (rung 0) becomes false. This causes instruction XIC R6:0/EN in rung 1 to become false, making instruction OTE O:0/6 in this rung false. This causes PLC output 6 to become deactivated.

Yes.

□ 20. Since PLC output lamps 0, 1, and 2 are on, instruction SQO R6:0 is at position 2 of sequencer file B10. This occurs because the rung containing this instruction (rung 0) has gone from false to true, causing this instruction to step to the next position in its sequencer file.

□ 21. Bits B10:2/0, B10:2/1, and B10:2/2 are set to logic state 1.

□ 22. Yes, since PLC outputs 0, 1, and 2 are activated. The reason why this occurs is that bits B10:2/0, B10:2/1, and B10:2/2 in the word at address B10:2 are set to logic state 1. Since these bits are allowed to pass to destination address O:0.0 through the mask, the bits at addresses O:0/0, O:0/1, and O:0/2 are set to logic state 1. This causes PLC outputs 0, 1, and 2 to be activated.

□ 23. Yes. This occurs because the rung containing instruction SQO R6:0 (rung 0) has gone from false to true, causing this instruction to step to position 3 of sequencer file B10. Since bits B10:3/1, B10:3/3, and B10:3/5 in the word at address B10:3 are set to logic state 1, and since these bits are allowed to pass to destination address O:0.0 through the mask, the bits at addresses O:0/1, O:0/3, and O:0/5 are set to logic state 1. This causes PLC outputs 1, 3, and 5 to be activated.

□ 24. When PLC input 0 is deactivated and then activated, the DN status bit of instruction SQO R6:0 is set to logic state 1 since this instruction steps to the last position in sequencer file B10 (position 4).This causes instruction XIC R6:0/DN in rung 2 to become true, making instruction OTE O:0/7 in this rung true. This causes PLC output 7 to become activated.

Yes. The reason why this occurs is that bits B10:4/0 through B10:4/5 in the word at address B10:4 are all set to logic state 1. Since these bits are allowed to pass to destination address O:0.0 through the mask, the bits at addresses O:0/0 through O:0/5 are all set to logic state 1. This causes PLC outputs 0 through 5 to be activated.

□ 25. The DN status bit is reset to logic state 0, because the rung containing instruction SQO R6:0 (rung 0) goes from false to true. This causes instruction XIC R6:0/DN in rung 2 to become false, making instruction OTE O:0/7 in this rung false. This causes PLC output 7 to become deactivated.

Yes.

□ 26. Yes. This occurs because activation of PLC input 1 causes instruction XIC I:0/1 in rung 3 to become true, making instruction RES R6:0 in this rung true. This causes instruction SQO R6:0 to return to position 0 of sequencer file B10. Since the rung of instruction SQO R6:0 is true, the word of data stored at position 1 of sequencer file B10 is transferred to destination address O:0.0 through the mask. As a result, PLC outputs 0 and 1 are activated.

□ 27. Yes.

□ 28. If the mask value of instruction SQO R6:0 were changed to "00F0H" (0000 0000 1111 0000 in binary), bits 0 through 3 of the words in sequencer file B10 would be masked. Consequently, the status of PLC outputs 0 through 5 for each sequencer position would be as follows:

Position 1: PLC outputs 0 through 5 deactivated;
Position 2: PLC outputs 0 through 5 deactivated;
Position 3: PLC outputs 0 through 4 deactivated, PLC output 5 activated;
Position 4: PLC outputs 0 through 3 deactivated, PLC outputs 4 and 5 activated.

□ 37. The EN status bit is at logic state 0 because the rung containing instruction SQC R6:0 (rung 0) is false.

Instruction SQC R6:0 is at position 0 of sequencer file B12. This occurs because immediately after startup, that is, when the PLC is switched from the **Program** mode to the **Run** mode, the SQC instruction is set to position 0 of its sequencer file.

□ 38. When PLC input 8 is activated, the EN status bit of instruction SQC R6:0 is set to logic state 1, since the rung containing this instruction (rung 0) becomes true. This causes instruction XIC R6:0/EN in rung 1 to become true, making instruction OTE O:0/0 in this rung true. This causes PLC output 0 to become activated.

Instruction SQC R6:0 is at position 1 of sequencer file B12. This occurs because the rung containing this instruction (rung 0) has gone from false to true, causing this instruction to step to the next position in sequencer file B12.

□ 39. Bits B12:1/0 and B12:1/2 are set to logic state 1.

□ 40. When PLC inputs 0 and 2 are activated, the FD status bit of instruction SQC R6:0 is set to logic state 1. This makes instruction XIC R6:0/FD in rung 3 true, making instruction OTE O:0/2 in this rung true. This causes PLC output 2 to become activated.

The reason why the FD status bit is set to logic state 1 is that all the non-masked bits at source address I:0.0 match the bits in the reference word at address B12:1. Thus, the bits at addresses I:0/0 and I:0/2 are set to logic state 1 due to PLC inputs 0 and 2 being activated. These bits are non-masked since the mask value of instruction SQC R6:0 is 00FFH (0000 0000 1111 1111 in binary). Consequently, all non-masked bits at source address I:0.0 match those in the reference word at address B12:1, whose bits B12:1/0 and B12:1/2 are set to logic state 1.

□ 41. When PLC input 8 is deactivated, the EN and FD status bits of instruction SQC R6:0 are set to logic state 0, since the rung containing this instruction (rung 0) becomes false.

The fact that the EN bit goes to logic state 0 makes instruction XIC R6:0/EN in rung 1 false, thereby making instruction OTE O:0/0 in this rung false and causing PLC output 0 to become deactivated.

The fact that the FD bit goes to logic state 0 makes instruction XIC R6:0/FD in rung 3 false, thereby making instruction OTE O:0/2 in this rung false and causing PLC output 2 to become deactivated.

□ 42. Yes. This occurs because activation of PLC input 8 causes the rung containing instruction SQC R6:0 (rung 0) to go from false to true, causing this instruction to step to the next position in its sequencer file.

□ 43. Yes. The reason why the FD status bit is set to logic state 1 is that the bits at addresses I:0/0, I:0/2, I:0/4, and I:0/6 are set to logic state 1, due to PLC inputs 0, 2, 4, and 6 being activated. Consequently, the status of all the non-masked bits at source address I:0.0 match the bits in the reference word at address B12:2, whose bits B12:2/0, B12:2/2, B12:2/4, and B12:2/6 are set to logic state 1.

□ 44. The EN status bit remains set to logic state 1 since the rung containing instruction SQC R6:0 remains true. However, the FD bit is set to logic state 0 since the non-masked bits at source address I:0.0 no longer match the bits in the reference word at address B12:2, due to bits I:0/4 and I:0/6 being set to logic state 0.

□ 45. PLC inputs 0 through 7 must be activated in order for the FD status bit of instruction SQC R6:0 to be set to logic state 1. This occurs because bits B12:3/0 through B12:3/7 in the reference word at address B12:3 are all set to logic state 1.

□ 46. When PLC input 8 is deactivated and then activated, the DN bit of instruction SQC R6:0 is set to logic state 1 since this instruction steps to the last position in sequencer file B12 (position 4). This makes instruction XIC R6:0/DN in rung 2 true, thereby making instruction OTE O:0/1 in this rung true and causing PLC output 1 to become activated.

PLC inputs 1, 3, 5, and 7 must be activated in order for the FD status bit of instruction SQC R6:0 to be set to logic state 1. This occurs because bits B12:4/1, B12:4/3, B12:4/5, and B12:4/7 in the reference word at address B12:4 are set to logic state 1.

□ 47. Instruction SQC R6:0 returns to position 1 of sequencer file B12 to initiate a new cycle. The DN status bit is reset to logic state 0 because the rung containing instruction SQC R6:0 (rung 0) goes from false to true.

□ 48. When PLC input 9 is activated, instruction SQC R6:0 returns to position 0 of sequencer file B12, while the FD status bit is reset to logic state 0. This occurs because activation of PLC input 9 makes instruction XIC I:0/9 in rung 4 true, making instruction RES R6:0 in this rung true.

□ 49. Yes.

□ 50. If the mask value of instruction SQC R6:0 were changed to "00F0H" (0000 0000 1111 0000 in binary), bits 0 through 3 of source address I:0.0 would be masked, so that you would need to activate the following PLC inputs in order for the FD status bit to be set to logic state 1:

Position 1: None
Position 2: PLC inputs 4 and 6
Position 3: PLC inputs 4 through 7
Position 4: PLC inputs 5 and 7

□ 53.

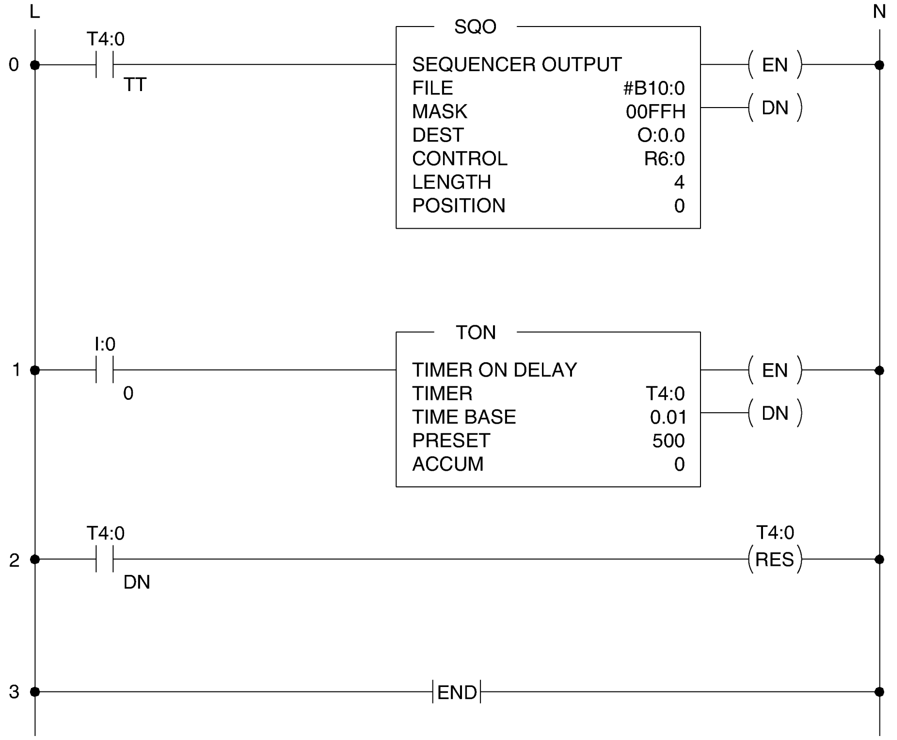


Figure 2. Suggested timer-driven SQO ladder program.

SEQUENCER POSITION	WORD	BINARY DATA			
		15	8	7	0
0	B10:0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	B10:1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
2	B10:2	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1
3	B10:3	0 0 0 0	0 0 0 0	1 1 1 1	0 0 0 0
4	B10:4	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1

Table 1. Data table for instruction SQO R6:0.

ANSWERS TO REVIEW QUESTIONS

1. The sequencer file is a file that stores data to be transferred to the destination file in the case of an SQO instruction, or reference data to be compared to the data at the source address in the case of an SQC instruction.

2. The mask value is a hexadecimal code or the address of a word or file through which the sequencer instruction transfers (SQO) or compares (SQC) data. Mask bits that are set to logic state 0 will mask data. Mask bits that are set to logic state 1 will pass data.

3. The SQO instruction transfers data from its sequencer file, through a mask, to a destination file on each false-to-true rung transition. Once the SQO instruction has reached the last position in its sequencer file, it automatically returns to position 1 on the next false-to-true rung transition.

4. The Found (FD) bit is set to logic state 1 when all the non-masked bits in the word or file at the source address match those of the corresponding reference word in the sequencer file. The FD bit is updated each time the processor evaluates the SQC instruction while the rung is true.

5. The Done (DN) bit passes from logic state 1 to logic state 0, and the SQO instruction returns to position 1 of its sequencer file to start a new cycle.

EXERCISE 8 COMPARISON INSTRUCTIONS

ANSWERS TO PROCEDURE QUESTIONS

□ 12. Yes. The reason why this occurs is that, when the accumulated value of instruction CTU C5:0 reaches 5, the value at Source A of the EQU instruction becomes equal to the value at Source B of this instruction (5), thereby making the EQU instruction true. This makes instruction OTE O:0/0 in rung 1 true, causing PLC output 0 to become activated.

□ 13. No. This occurs because the value at Source A of the EQU instruction, which is 6, is now unequal to the value at Source B of this instruction, which is 5. This makes the EQU instruction false, thereby making instruction OTE O:0/0 in rung 1 false and causing PLC output 0 to be deactivated.

□ 14. When PLC input 1 is activated, the accumulated value of instruction CTU C5:0 is reset to 0. This occurs because activation of PLC input 1 makes instruction XIC I:0/1 in rung 2 true, making instruction RES C5:0 in this rung true.

□ 20. Yes. The reason why this occurs is that the value at Source A of the NEQ instruction, which is 0, is not equal to the value at Source B of this instruction, which is 13. Because of this, the NEQ instruction is true, thereby making instruction OTE O:0/2 in rung 3 true and causing PLC output 2 to be activated.