SELF-TEACH MANUAL

Allen-Bradley Bulletin 1742
Modular Automation Controller

Price: $25.00 Per Copy

IMPORTANT INFORMATION

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Because of this, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment, or software described in this manual.

RELATED DOCUMENTATION

Application Considerations for Solid State Controls – Describes some important differences between solid state programmable controller products and hard-wired electromechanical devices. **Pub. SGI-1.1**

User's Manual – Refer to this manual for information on installation, start-up, maintenance/troubleshooting, and specifications. The User's Manual also includes the programming and operation information contained in the text portion of the Self-Teach Manual. **Pub. 1742-800**

Product Data – 120VAC Processor Module	**Pub. 1742-2.1**
Product Data – 120VAC Input Module	**Pub. 1742-2.2**
Product Data – 120VAC Output Module	**Pub. 1742-2.5**
Product Data – Operator Terminal	**Pub. 1742-2.9**
Product Data – EEPROM Memory Module	**Pub. 1742-2.10**
Product Data – Carrying Case	**Pub. 1742-2.11**
Quick Reference Guide	**Pub. MC-279**
Controller Mounting Templates	**Nos. 5116-200A, 5116-201A**

TABLE OF CONTENTS

Section 1
INTRODUCTION

PURPOSE

We developed this manual to help you become familiar with the programming procedures of the Modular Automation Controller.

The manual was written with the assumption that you are already familiar with industrial relay control techniques. In programming the controller, you will be using relay-type symbology to create the individual ladder rungs of your program. And you will still be using hard-wired control devices in the external I/O circuits of your controller.

A second assumption is that you have access to a controller or a factory demonstrator. A practice controller setup is described in Paragraph 3.3; the demonstrator is described in the appendix.

Emphasis is placed on practical knowledge. You will learn more about how the controller is *applied* than how it functions internally.

HOW TO USE THIS MANUAL

This is the text portion of the Self-Teach Manual. It is designed for use with the accompanying Study Guide.

Sections 2 and 3 of the text, entitled EQUIPMENT OVERVIEW and PROGRAMMING AND OPERATION are basically the same as Sections 2 and 3 of the User's Manual. We have divided these sections into the 13 study units indicated at the right.

The Study Guide consists of Question/Exercise Units corresponding to the 13 study units. Question/Exercise Units include questions, a summary of key terms and concepts, and in some cases additional programming details, examples, and exercises.

We suggest that you read the study units in numerical sequence. Read each study unit two times or more. Use a controller practice set-up or a factory demonstrator to familiarize yourself with programming procedures. Then refer to the corresponding Question/Exercise Unit to answer the questions and review the terms and concepts introduced in the study unit.

Set your own pace. Later, when you are satisfied that you have a good grasp of the entire text material, use the manual as a handy reference to refresh your memory from time to time.

STUDY UNITS

BULLETIN 1742 – MODULAR AUTOMATION CONTROLLER

The Bulletin 1742 Modular Automation Controller is a microprocessor-based programmable controller. It is a technologically advanced control system having the inherent flexibility and advantages characteristic of other programmable controllers–but with an important difference: Simplicity. It was designed with the first-time user in mind.

You will find that the controller is relatively easy to program, operate, and maintain, allowing you to take full advantage of its capabilites in the shortest possible time.

HARD-WIRED CONTROL VERSUS THE BULLETIN 1742

A comparison of a hard-wired relay logic system and a system using the Modular Automation Controller will show you the similarities which make the Bulletin 1742 controller so easy to apply.

Figure 1.1a shows a hard-wired relay control logic system. For purposes of comparison, it is divided into sections, consisting of:

Input Devices, including devices operated manually (push buttons) and devices operated automatically (limit switches) by the machine or process being controlled.

Relay Control Logic, consisting of relays interconnected to energize or de-energize output devices in response to the status of the input devices, and in accordance with the logic designed into the circuit.

Output Devices, consisting of motor starters, solenoids, etc. which control the machine or process.

Figure 1.1b shows a similar control system, using the Bulletin 1742 Modular Automation Controller. The only difference in this figure is that the relay control logic section is replaced by the Modular Automation Controller.

In place of hard-wired relay circuitry, the Modular Automation Controller contains programmed instructions which use **relay-type symbology**. Thus, with the Bulletin 1742 controller, you are still employing familiar relay logic methods–but in a new way, which provides a great deal more flexibility.

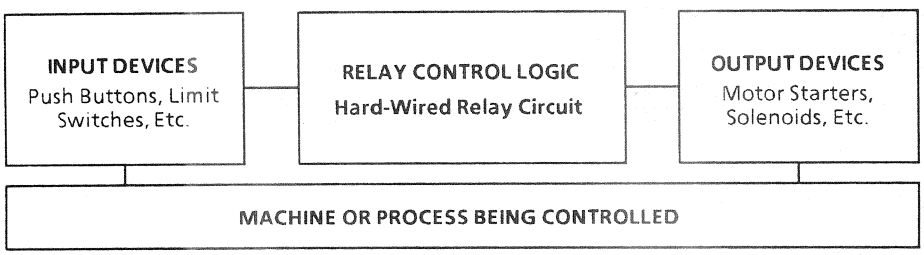


Figure 1.1a System using hard-wired relay control logic.

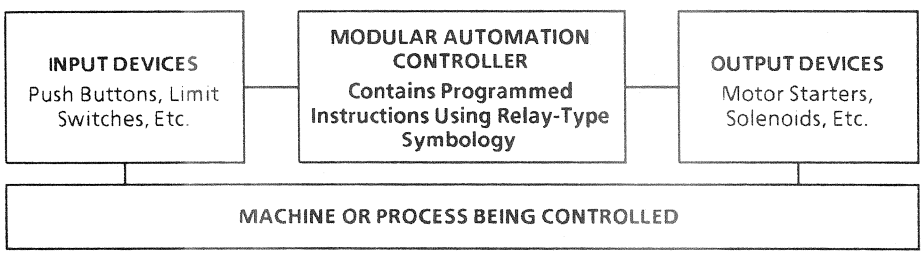
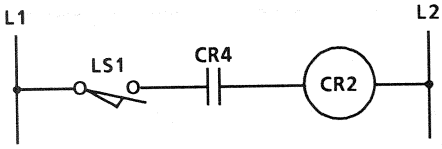
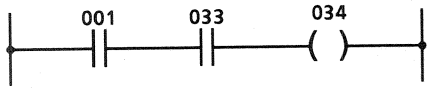


Figure 1.1b Similar system using the Bulletin 1742 Modular Automation Controller.

The simplicity of the ladder diagram programming format is illustrated in Figure 1.2, which shows a relay ladder rung as used in relay control systems and a similar ladder rung as used in programming the Modular Automation Controller.



Relay ladder rung consisting of a limit switch (LS1), a N.O. contact (CR4), and a coil (CR2). A continuous path is needed for **electrical** continuity.



Similar PC ladder rung. Instruction address numbers 001, 033, and 034 are identified with LS1, CR4, and CR2 respectively. A continuous path is needed for **logical** continuity.

Figure 1.2 Relay ladder rung and a similar ladder rung used in programming the Modular Automation Controller.

Section 2
EQUIPMENT OVERVIEW

2.0 GENERAL

The Modular Automation Controller includes the following modular components and accessories, easily assembled to meet your requirements.

- A processor module.
- Input and output modules, which plug into the processor module.
- A mounting plate to simplify assembly and installation.
- An operator terminal and cable to connect it to the processor module.
- An optional EEPROM memory module, which plugs into the processor module.

Figure 2.1 shows an assembled Modular Automation Controller in the maximum configuration, consisting of a processor module and eight input/output modules. The unit shown has four input modules and four output modules, but any other desired combination of input and output modules is possible (eight modules maximum).

Note that the modules are fastened to a mounting plate. This plate simplifies installation, since it eliminates the need to locate and drill mounting holes for the processor module and input/output modules.

The operator terminal and interconnection cable are shown in the foreground of the figure, together with an EEPROM memory module.

The operator terminal is a hand-held, portable device, which you connect to the processor module when it is necessary to enter or edit program instructions or to monitor operation. It is also used as a troubleshooting aid.

A single operator terminal can serve a number of separate controllers, since there is no need to keep the operator terminal connected during normal controller operation.

The memory module is an optional plug-in device, providing a convenient means of duplicating and storing the instructions you have entered in the processor module.

Figure 2.1 Assembled Modular Automation Controller, 32 I/O configuration. The operator terminal, interconnect cable, and EEPROM memory module are shown in the foreground.

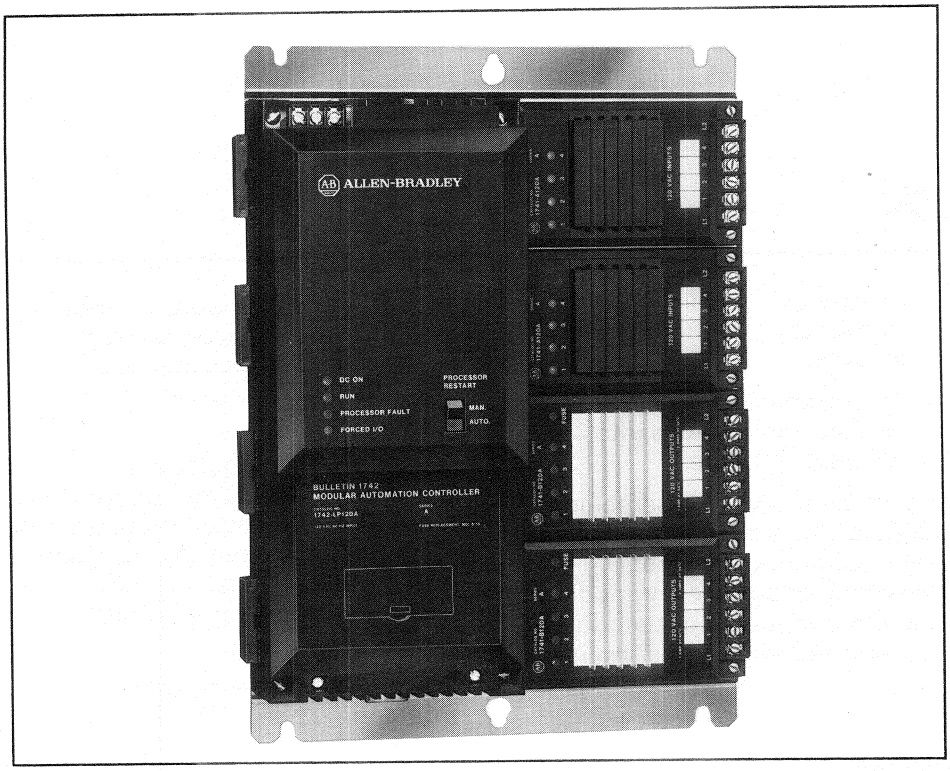


Figure 2.2 Assembled Modular Automation Controller, 16 I/O configuration.

Figure 2.2 shows another controller, consisting of a processor module, two input modules, and two output modules assembled on a smaller (16 I/O) mounting plate. If the application required inputs and outputs in a different combination, such as one input module and three output modules or vice versa, it could very easily be configured that way.

This modular design allows you to configure the controller as the application requires. The minimum controller configuration has one input module and one output module, while the maximum configuration has eight input/output modules–in the combination which best suits your application. Modularity also allows you to change the configuration, if the need arises.

2.1 COMPONENT IDENTIFICATION

Bulletin 1742 Processor Module	
Nominal Operating Voltage	**Catalog Number**
110/120 VAC, 50/60 Hz	1742-LP120A

Bulletin 1741 Mounting Plate	
Input/Output Capacity	**Catalog Number**
16 I/O (4 Modules) 32 I/O (8 Modules)	1741-MP16 1741-MP32

Bulletin 1741 EEPROM Memory Module
Catalog Number 1741-M1

Bulletin 1741 Input Module	
Nominal Operating Voltage	**Catalog Number**
110/120 VAC, 50/60 Hz 10-30 VDC	1741-A120A 1741-A30C

Bulletin 1741 Output Module	
Nominal Operating Voltage	**Catalog Number**
110/120 VAC, 50/60 Hz 10-50 VDC 240 VAC, 125 VDC Max.	1741-B120A 1741-B50C 1741-BCO

Bulletin 1740 Operator Terminal
Catalog Number 1740-PT1 (Includes Cat. No. 1741-C1 Cable)

2.2 COMPONENT FEATURES

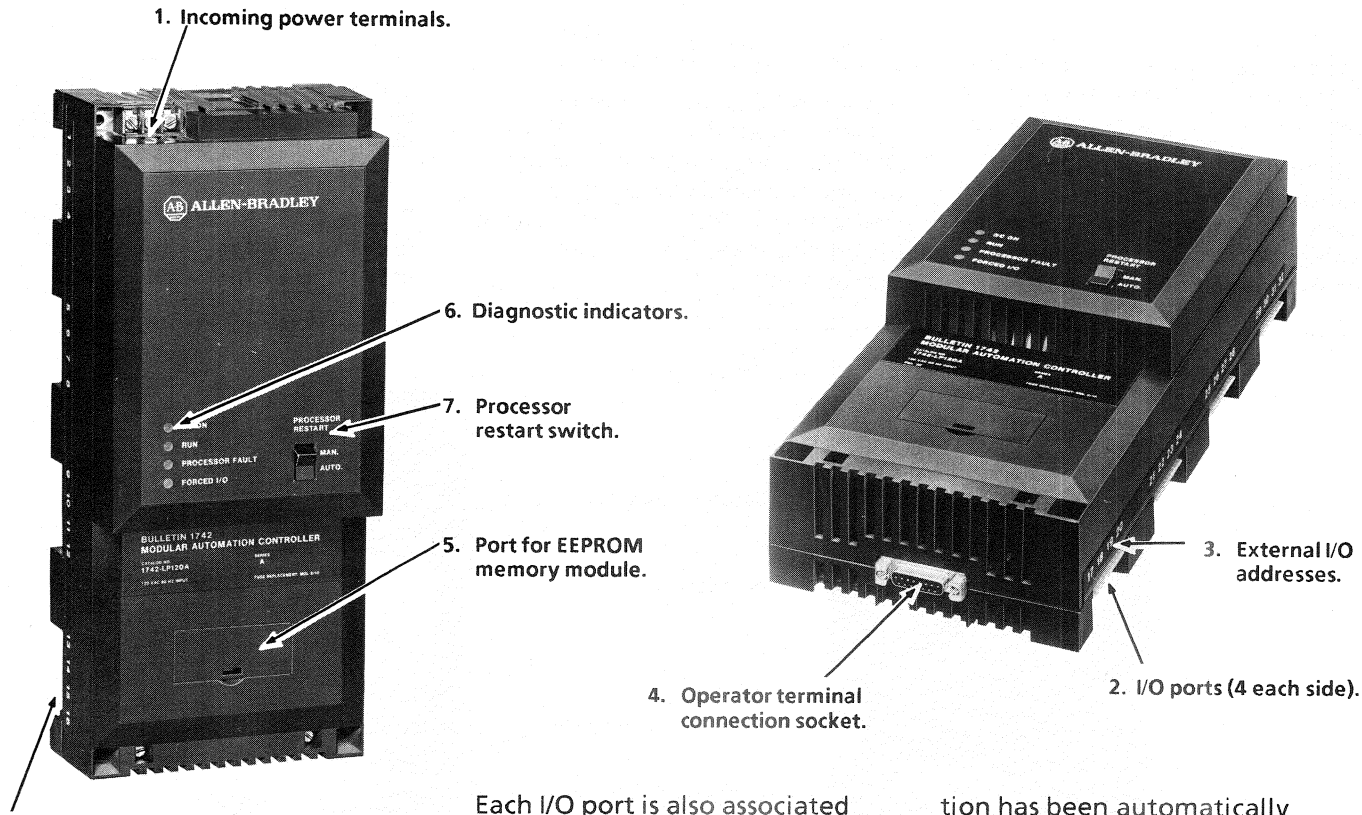
Significant features of the Modular Automation Controller components are pointed out in the following paragraphs.

2.2.1 Processor Module – The processor module houses solid state circuitry which processes and manipulates programmed information, causing output devices to be energized and de-energized in response to the ON/OFF status of input devices.

The programmed information is entered with the operator terminal, and stored in the processor CMOS RAM memory. A battery back-up is provided, so that the memory contents is retained when processor power is removed.

DC operating power for the processor module, the plug-in I/O modules, and the operator terminal is supplied by an internal power supply. The power supply fuse is accessed by removing the processor module cover. (Fuse replacement is discussed in Section 6 of the User's Manual.)

Figure 2.3 will show you some of the important external features of the processor module.



1. Incoming power terminals: Terminals at the top of the processor module are for incoming power connections. Refer to Section 4 of the User's Manual for wiring and grounding recommendations.

Terminals have self-lifting pressure plates for easy wire insertion and secure connections. One or two #14 AWG stranded wires can be attached to each terminal.

2. I/O ports (covers removed): The processor module has four I/O ports on each side. You can plug an input module or an output module into any port. Any logical arrangement of modules is suitable.

Both AC and DC I/O modules can be used in the same controller.

3. External I/O addresses: Four I/O address numbers appear directly above each I/O port, identifying the four circuits of the module assigned to the port. Each I/O port is also associated with a module group number. External I/O addresses and module group numbers are discussed later, in Section 3.

4. Operator terminal connection socket: Connect the operator terminal interconnect cable to this socket when you want to use the operator terminal to program, edit, or monitor operation.

5. Port for EEPROM memory module: Lift this access door to insert the EEPROM. The EEPROM module is illustrated on Page 2-7.

6. Diagnostic indicators: These four LED diagnostic indicators have the following functions:

DC ON – This is a green LED, which indicates that the processor module is supplying DC operating power.

RUN – This is a green LED, indicating that the processor is operating in the Run mode. Operating *modes* are discussed in Section 3.

PROCESSOR FAULT – A red LED, indicating that processor operation has been automatically stopped due to an error within the processor.

FORCED I/O – This is an amber LED, indicating that one or more inputs or outputs have been forced to an ON or OFF state. *Forcing* of inputs and outputs is discussed in Section 3.

7. Processor restart switch: This switch gives you the choice of re-entering the Run mode automatically or manually on power-up. (Processor must have been in the Run mode when power was removed.)

AUTO position – On power-up, the processor goes thru its normal diagnostic tests, then automatically enters the Run mode.

MAN position – Use this position when you do not want the processor to automatically enter the Run mode on power-up. In this case, the processor goes thru its diagnostic tests, but outputs are not enabled and the user program is not executed. To enter the Run mode, you must move the switch to AUTO.

Figure 2.3 External features of the processor module.

2.2.2 I/O Modules – Input and output module circuitry interfaces the logic level voltages of the controller and the voltage levels of the external input and output devices.

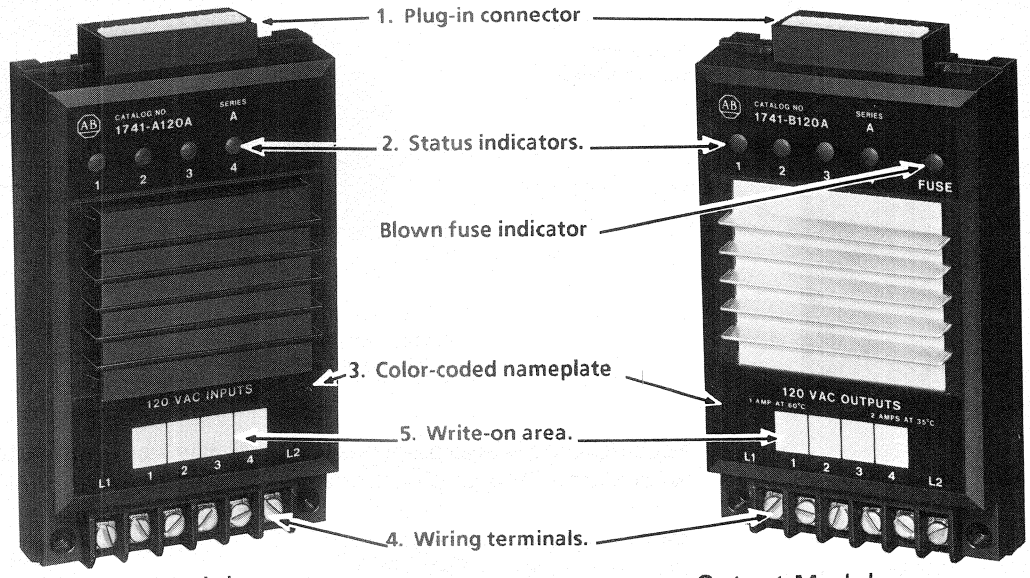
Input modules include optical isolation, as well as filtering and surge suppression to guard against possible damage by transients from external sources.

Output modules include optical isolation, surge suppression, and also short circuit protection.

Input modules sense whether external input devices are ON or OFF, continuously providing the processor module with status information. Typical input devices include limit switches, push buttons, etc.

Output modules control the ON or OFF status of external output devices, based on logic level commands received from the processor module. Typical output devices include solenoids, motor starters, etc.

Figure 2.4a indicates some of the important external features of input and output modules.



1. Plug-in connector: Simply plug the module into any of the processor module I/O ports.

2. Status indicators: Input modules are provided with LED status indicators. An indicator is lit when the corresponding input circuit is energized.

Output modules are provided with neon lamp status indicators. An indicator is lit when the corresponding output circuit is energized.

Output modules also have a blown fuse indicator, which will be lit when one of the four fuses is blown. Fuses are accessed by opening a cover on the front of the module. Fuse replacement is discussed in Section 6 of the User's Manual.

3. Color-coded nameplate: For easy module identification, nameplates are color-coded as follows:

Module	Volts	Color
Input	120VAC 10-30VDC	Red Blue
Output	120VAC 10-50VDC	Orange Green

Each module is supplied with a correspondingly color-coded identification strip. We suggest that you attach this self-adhesive strip to the processor module, directly above the address numbers of the assigned I/O port (see Figure 2.4b). Then, if you remove the module for any reason, the color of the strip will help you identify the type of module assigned to the port.

4. Wiring terminals: Connect external input and output devices to terminals 1-2-3-4. Connect the I/O circuit source voltage to the remaining two terminals. (Typical connections are shown in Figure 2.4b.)

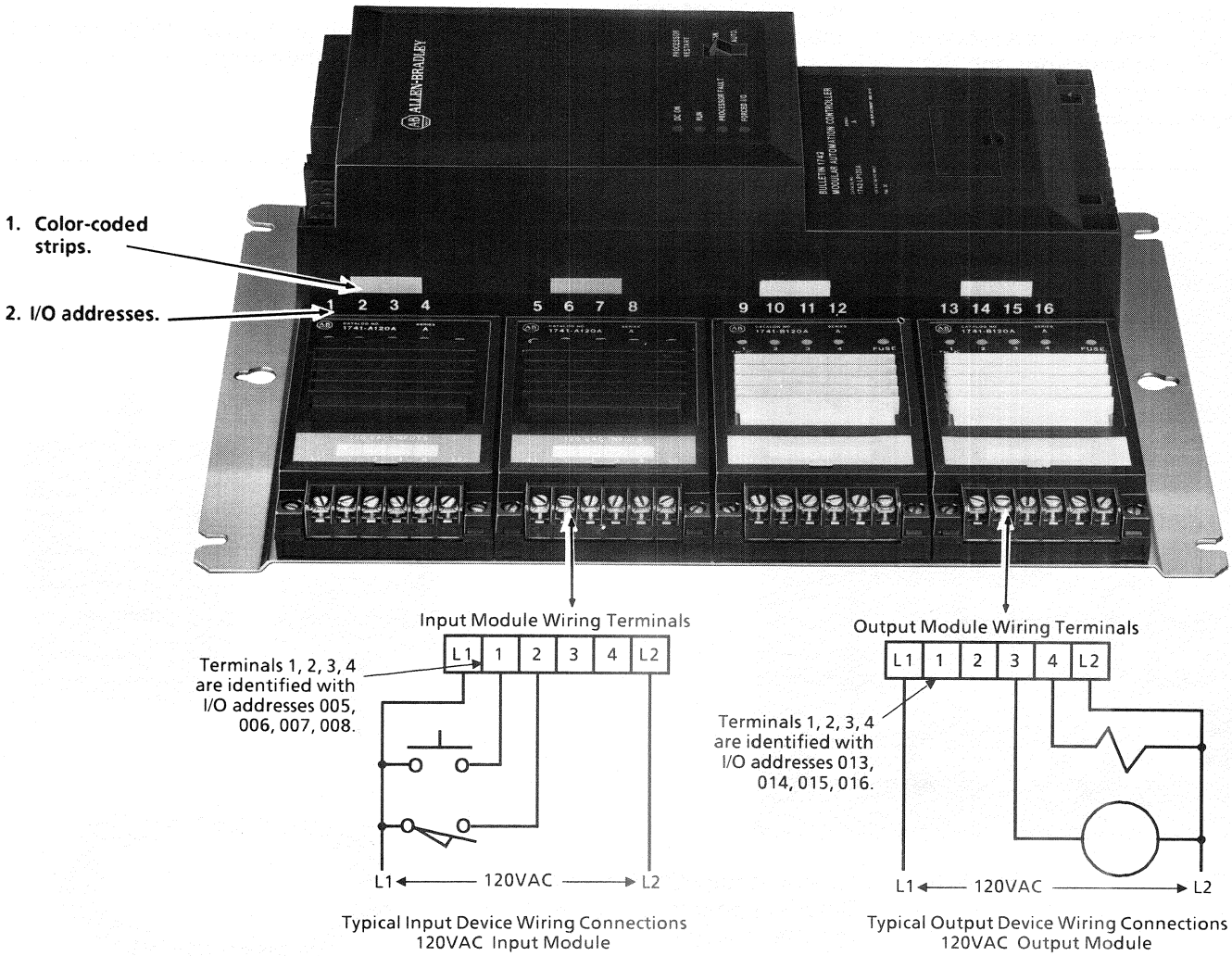
Terminals 1-2-3-4 (and associated status indicators) correspond to the four I/O address numbers appearing above the I/O port to which the module is assigned.

Terminals have self-lifting pressure plates for easy and reliable connections. One or two #14 AWG stranded wires can be attached to each terminal.

5. Write-on area: Use this write-on area to identify the individual external input and output devices.

Figure 2.4a External features of input and output modules.

Figure 2.4b shows a controller consisting of a processor module, two input modules, and two output modules secured to a mounting plate. The controller is shown in a horizontal position to more clearly illustrate the plug-in I/O modules. Typical wiring connections for 120VAC input and output modules are shown.



1. Color-coded strips: These identification strips are attached as suggested in Figure 2.4a. If you remove modules for any reason, the color-coded strips will help you identify the type of module assigned to each port.

2. I/O addresses: Four address numbers appear above each I/O port of the processor module. When you enter a program, you will be assigning these addresses to the instructions representing your external I/O devices. (Address numbers are generally indicated in 3-digit form; thus, the addresses 1, 2, 3, 4, etc. appearing on the processor module become addresses 001, 002, 003, 004, etc. when using the 3-digit form.)

Figure 2.4b Controller having two input modules and two output modules.

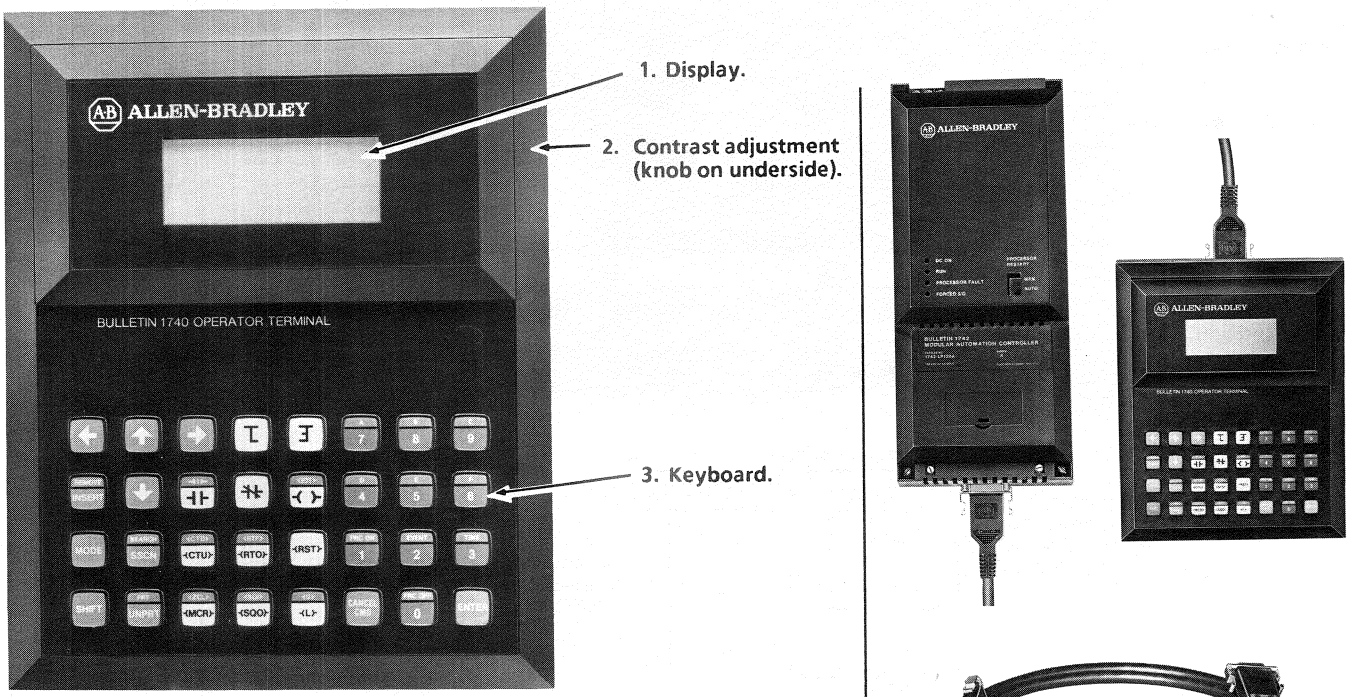
2.2.3 Operator Terminal – This is a hand-held, portable device used to program, edit, and monitor controller operation. It is easily connected by cable to the processor module when needed.

The operator terminal is your means of communicating with the controller. You use the *keyboard* to enter the instructions and data which make up your program; the *display* shows you these instructions and data as you enter them.

The operator terminal responds to your keyboard entries with prompting messages for the next step in entering instructions. Error codes appear when you forget something or do something wrong. (Error codes also appear if internal processor problems occur.)

A reference table appears on the back of the operator terminal (illustration on Page 3-7), providing you with quick reference information on modes, addresses, and error codes. You will become familiar with these subjects as you read Section 3.

Figure 2.5 indicates some of the important features of the operator terminal. You will find details on using the operator terminal in Section 3.



1. Display: This liquid crystal display (LCD) is comprised of a 28 x 80 dot matrix. It displays ladder rungs consisting of instruction symbols, and also various data and messages.

2. Contrast adjustment: An adjustment knob is provided for setting the display contrast to suit your viewing preference. It is located at the right edge of the operator terminal, accessible from the bottom. It is easily adjusted with your right forefinger while you view the display.

The contrast adjustment knob also functions as a reset, by turning it fully CCW, then CW. Details appear in Section 6 of the User's Manual.

3. Keyboard: The keyboard is the sealed membrane type, with tactile feedback. You will feel a distinct, positive reaction as you press each key.

The keyboard has 32 keys, color-coded according to function. Keys for entering the instruction set are white; keys for control, editing, and special functions are red; keys for entering numerics, forcing and sequencer data are blue. Upper-case functions, shown in grey, are implemented with the grey Shift key.

Cable connection: A 6.6 ft interconnect cable is included with the operator terminal. Connect one end to the socket on the top of the operator terminal and the other end to the socket on the bottom of the processor module. The cable connectors are keyed to guard against improper insertion.

When connecting the cable, be sure the spring latches are engaged to secure the cable.

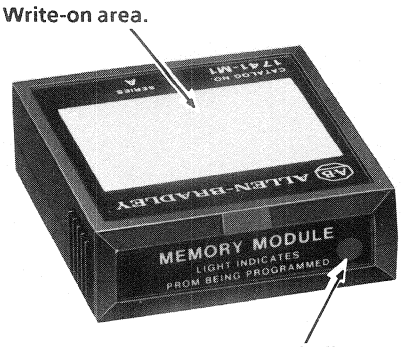
Figure 2.5 Operator terminal features.

2.2.4 EEPROM Memory Module – This is an optional device, easily plugged into the processor module. With it, you can:

- Duplicate the contents of the processor RAM memory, for storage purposes.
- Copy the contents of the EEPROM memory in the processor RAM. This is a time-saving convenience in applications where the same program is to be used in a number of processors.
- Operate directly from the EEPROM (the processor RAM contents will be ignored).

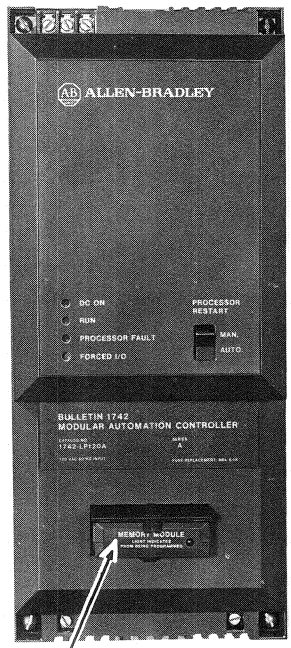
The EEPROM module is illustrated in Figure 2.6.

You will find detailed information on installing and using the EEPROM in Section 3.



1. Write-on area: A write-on area is provided on the memory module, for use in identifying its contents.

2. LED status indicator: The LED status indicator on the front of the memory module will be ON when a program is being loaded into the module.



The EEPROM module installed.

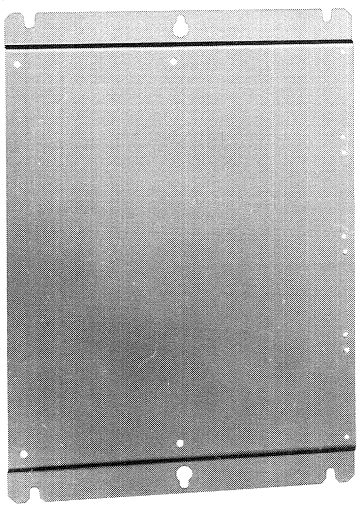
Figure 2.6 The EEPROM memory module.

2.2.5 Mounting Plates – The optional mounting plate simplifies installation, since it eliminates the need to locate and drill mounting holes for the processor module and I/O modules. You simply secure the processor module to the mounting plate, then plug in the I/O modules and secure them to the plate. Mounting screws are supplied with each module.

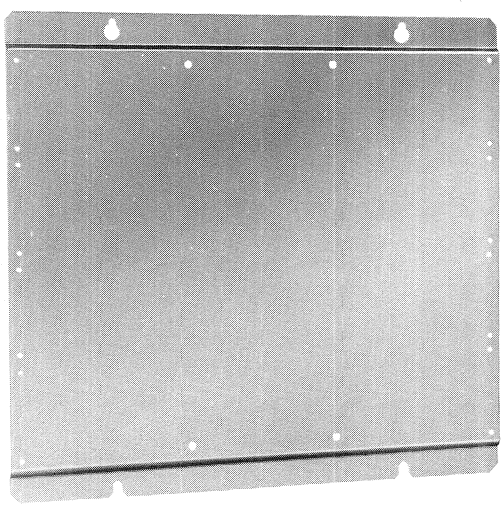
Two mounting plates are available. They are illustrated in Figure 2.7.

STOP

Re-read Section 2. Then answer the questions in Q/E Unit 1 of the Study Guide.



16 I/O mounting plate.



32 I/O mounting plate.

16 I/O mounting plate: Use up to four I/O modules with this plate. Note that there is a keyhole and two slots on both top and bottom of the plate. Thus, by inverting the plate you can mount the I/O modules on either the left or right side of the processor module, as you prefer.

32 I/O mounting plate: This plate allows you to mount up to eight I/O modules, four on each side of the processor module.

Figure 2.7 16 I/O and 32 I/O mounting plates.

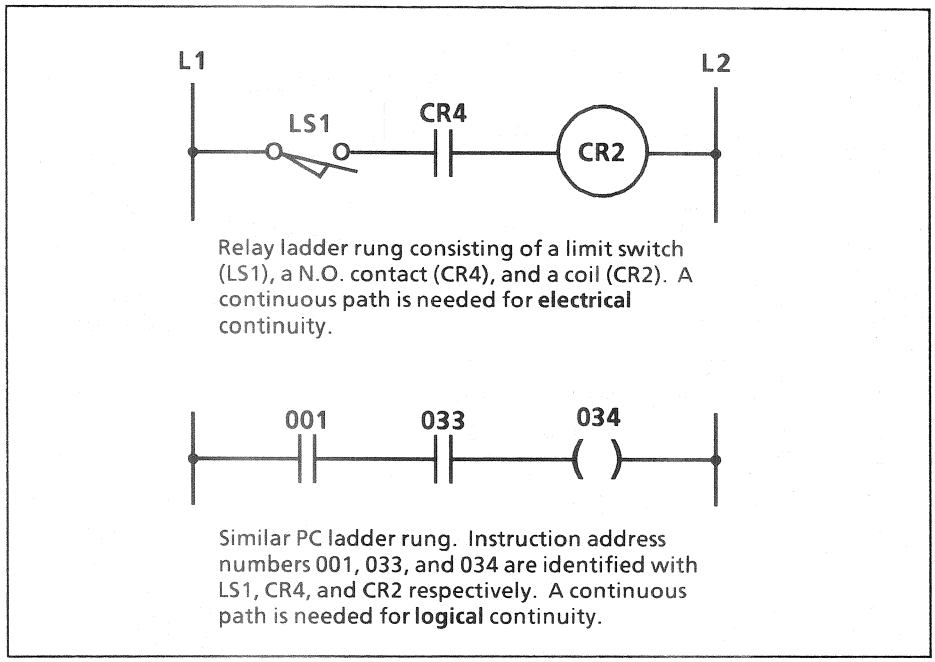
Section 3
PROGRAMMING AND OPERATION

3.0 GENERAL

This section covers the following topics:

- Programming Basics.
- The Operator Terminal.
- A Hands-On Introduction to Programming.
- Programming and Operation of the Instruction Set Groups –

  Relay Type Instructions.
  Timer Instructions.
  Counter Instructions.
  Sequencer Instructions.
  Special Instructions.
- Program Editing.
- On Line Data Control.
- Using the EEPROM Memory Module.



Relay ladder rung consisting of a limit switch (LS1), a N.O. contact (CR4), and a coil (CR2). A continuous path is needed for **electrical** continuity.

Similar PC ladder rung. Instruction address numbers 001, 033, and 034 are identified with LS1, CR4, and CR2 respectively. A continuous path is needed for **logical** continuity.

Figure 3.1 Relay ladder rung and a similar ladder rung used in programming the Modular Automation Controller.

3.1 PROGRAMMING BASICS

The following is a brief introduction to programming the Modular Automation Controller.

3.1.1 Ladder Diagram Format – The programming format for the Modular Automation Controller is the *ladder diagram*, which uses symbology similar to hard-wired relay ladder circuits. Figure 3.1 shows a relay ladder rung as used in hard-wired relay control systems and a similar PC ladder rung as used in programming the Modular Automation Controller. With the relay ladder rung, *electrical* continuity is required to energize the output, whereas in the PC ladder rung, *logical* continuity is required to energize the output.

3.1.2 Instructions – In the PC ladder rung of Figure 3.1, the individual symbols represent *instructions*; the numbers 001, 033, and 034 are the instruction *addresses*. When programming the controller, these instructions are entered one-by-one into the processor memory from the operator terminal keyboard. Instructions are stored in the *user program* portion of the processor memory.

Some of the instructions entered are used to represent the external input and output devices connected to the I/O modules of the controller; other instructions are "internal", used to establish the exact conditions under which the processor will energize or de-energize output devices in response to the status of input devices.

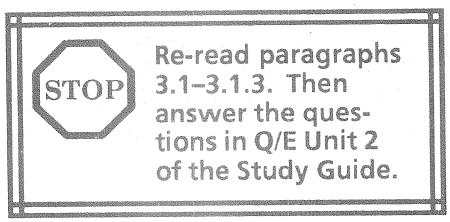
The memory storage unit for instructions is the *word*, with most instructions occupying one word of memory storage space. A total of 884 words are available in the user program for storing instructions.

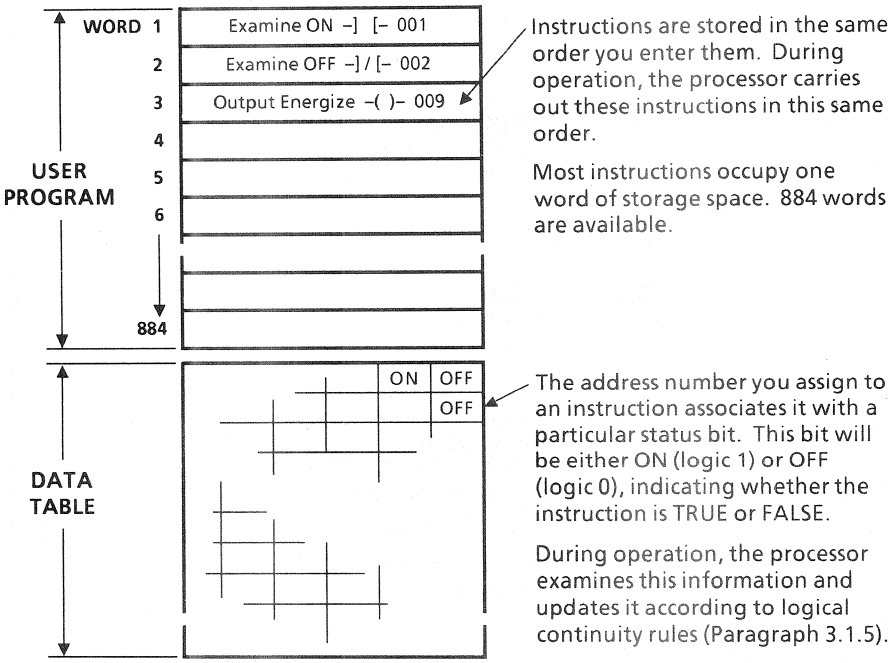
3.1.3 Instruction Addressing – To complete the entry of an instruction, you must assign an *address* number to it. This number identifies the function of an instruction and links it to a particular *status bit* in the *data table* portion of the memory.

The status bits in the data table can be either ON (logic 1 state) or OFF (logic 0 state), indicating the TRUE/FALSE status of the instructions they are associated with.

Figure 3.2 shows a simplified representation of the user program and data table areas of the processor memory.

Three-digit decimal numbers are used for instruction addresses. As indicated in Figure 3.3, the address number determines the function of the instruction. Thus, addresses 001 thru 032 are used for relay-type instructions associated with *external I/O devices* (paragraph 3.1.6). All other addresses are used *internally* for relay-type instructions, timer/counter/sequencer instructions, and associated status bits.





Instructions are stored in the same order you enter them. During operation, the processor carries out these instructions in this same order.

Most instructions occupy one word of storage space. 884 words are available.

The address number you assign to an instruction associates it with a particular status bit. This bit will be either ON (logic 1) or OFF (logic 0), indicating whether the instruction is TRUE or FALSE.

During operation, the processor examines this information and updates it according to logical continuity rules (Paragraph 3.1.5).

Figure 3.2 Simplified representation of the processor memory. The **User Program** is the memory area which stores the list of instructions you enter. The **Data Table** consists of status bits which indicate the TRUE/FALSE status of the instructions you enter. Numerical values associated with timer, counter, and sequencer instructions are also stored in the data table.

Address	Function
001-032	Relay-Type instructions used for external I/O points.
033-098	Relay-Type instructions used internally.
099	Program Initialization Instruction (used internally) – Used to initialize program to known state on power-up. This bit is always ON for the first scan.
901-932	Timer/Counter/Sequencer instructions – Used internally.
951-982	Status Instructions – Timer/Counter overflow/underflow and Sequencer completion. These internal addresses indicate when timer and counter values go over 9999 or below 0000, or when a sequencer has reached its last step. To determine the overflow/underflow/completion bit address, add 50 to the timer/counter/sequencer address. Thus, address 951 represents the overflow value of address 901; 952 represents the overflow address of 902, and so forth up to address 982, which represents the overflow of address 932.

Figure 3.3 Instruction address numbers for the Modular Automation Controller.

3.1.4 Instruction Classifications – Instructions are classified as *condition* instructions and *output* instructions. The instruction *set* for the Modular Automation Controller is shown in Figure 3.4.

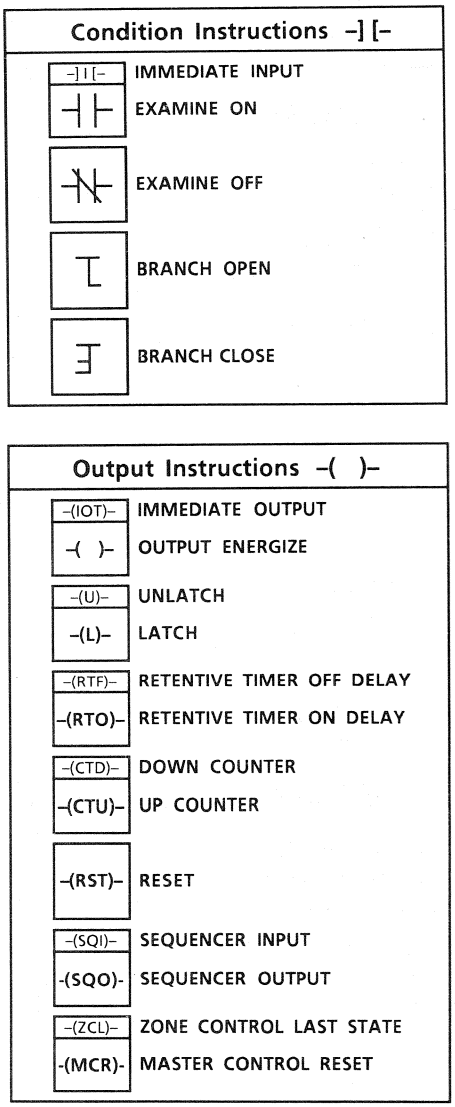


Figure 3.4 Instruction set of the Modular Automation Controller. Symbols are representations of the white keys of the operator terminal keyboard.

3.1.5 Logical Continuity – An example of how condition and output instructions are used is shown in the ladder diagram of Figure 3.5. The *Examine ON* and *Examine OFF* instructions (conditional) are analogous to relay contacts, while the *Output Energize* instruction is analogous to a relay coil. However, this diagram must be evaluated in terms of *logical* continuity rather than *electrical* continuity.

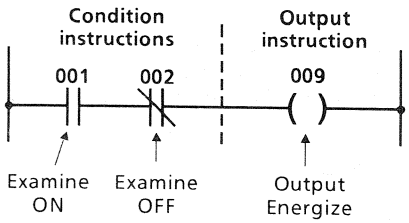


Figure 3.5 Ladder diagram rung. A continuous path of TRUE Condition instructions is required to make the Output Instruction TRUE.

As stated earlier, each instruction is linked to a status bit in the data table. The bit will be either ON or OFF to indicate the status of the instruction. Thus, with the Examine ON instruction, we are asking the controller to "examine the status bit for an ON condition" If the status bit is ON, then the instruction is considered TRUE; if the bit is OFF, then the instruction is FALSE.

Similarly, the Examine OFF instruction means "examine the status bit for an OFF condition". If the status bit is OFF, the instruction is considered TRUE; if the bit is ON, the instruction is FALSE.

The Output Energize instruction asks the controller to "set the status bit of the addressed Output Energize instruction to ON when rung conditions are TRUE".

Thus, in Figure 3.5, when both the Examine ON instruction is TRUE (status bit ON) and the Examine OFF instruction is TRUE (status bit OFF), the status bit of the Output Energize instruction will be set to ON.

In terms of continuity: When there is a continuous path of TRUE *conditional* instructions in a rung, logical continuity exists; accordingly, the output instruction is TRUE and its status bit will be set ON. If any conditional instruction in the continuity path goes FALSE, logical continuity is lost; the output instruction is then FALSE and its status bit will be set to OFF.

3.1.6 External I/O Devices – The user program always includes instructions to represent external devices connected to the controller I/O modules. The controller examines the status of these external devices indirectly.

Thus, in Figure 3.6a, the limit switch contact connected to input module terminal 3 causes a voltage to be present at the terminal when the switch is closed, and removes this voltage when the switch is opened. The input module senses these voltage levels, and the processor records it in its memory by setting the status bit of the instruction representing the limit switch to ON for the closed condition and OFF for the open condition.

If an Examine ON instruction is used to represent the limit switch contact, the open condition makes the instruction logically FALSE, and the closed condition makes it logically TRUE.

If an Examine OFF instruction is used to represent the limit switch contact, the open condition makes the instruction logically TRUE, and the closed condition makes it logically FALSE. Figure 3.6b summarizes both the Examine ON and Examine OFF cases.

Figure 3.6b also summarizes the Output Energize instruction as it applies to an external I/O address. In this case, if the status bit for the instruction is ON, voltage is applied to the external circuit, and the terminal is considered to be ON. If logical continuity of the rung is lost, the status bit is changed to OFF, and voltage is removed from the external circuit.

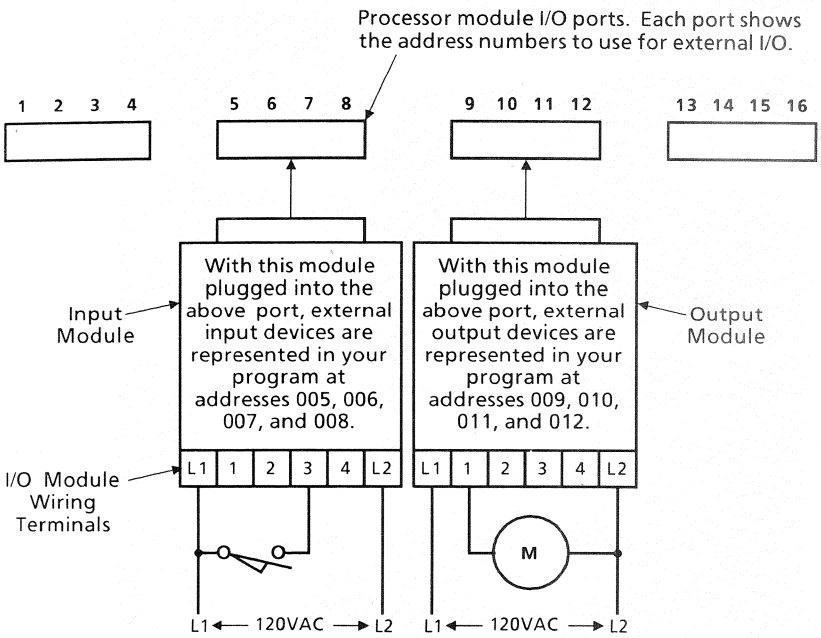


Figure 3.6a External input and output device connections. 120VAC I/O modules are used in this example. I/O ports with address numbers 1 thru 16 are shown. Address numbers 17 thru 32 appear at the I/O ports on the other side of the processor module.

External Input Device	Input Module Terminal	Examine ON Instruction	Examine OFF Instruction
OPEN	OFF	FALSE Status Bit OFF	TRUE Status Bit OFF
CLOSED	ON	TRUE Status Bit ON	FALSE Status Bit ON

Output Energize Instruction	Output Module Terminal	External Output Device
Status Bit ON TRUE Rung Conditions	ON	M ENERGIZED
Status Bit OFF FALSE Rung Conditions	OFF	M NOT ENERGIZED

Figure 3.6b Examine ON, Examine OFF, and Output Energize instructions, as they relate to external input and output devices.

3.1.7 Controller Operating Cycle – During each operating cycle, the controller examines the status of input devices, executes the user program, and changes outputs accordingly. This cycle is repeated about 40 times each second (varies with program length).

A single operating cycle or *scan* is illustrated in Figure 3.7. Note that it is divided into two distinct parts –the I/O scan and the program scan.

I/O scan: During this part of the cycle, data associated with external outputs is transferred from the data table to the corresponding output module terminals. (This data was updated during the preceding program scan.) In addition, input module terminals are examined, and the associated status bits are changed accordingly.

Program scan: The updated input device status information is applied to the user program during this part of the cycle. The processor executes the entire list of instructions in the same order they were entered. Status bits are updated according to logical continuity rules (Paragraph 3.1.5), as the program scan moves from instruction to instruction thru successive ladder rungs.

The I/O scan and program scan are separate, independent functions. Thus, any status changes occurring in external input devices during the program scan are not accounted for until the next I/O scan. Similarly, data changes associated with external outputs are not transferred to the output module terminals until the next I/O scan. Exception: Special instructions, Immediate Input and Immediate Output, Page 3-31.

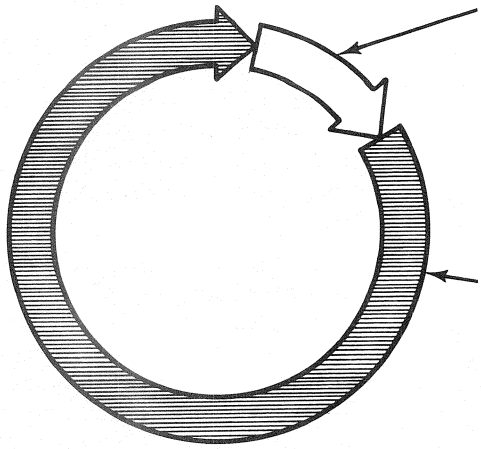


Figure 3.7 Operating cycle of the Modular Automation Controller. Typical cycle time is 25 milliseconds (depends on program content and length).

3.2 OPERATOR TERMINAL

Instructions are entered into the processor memory with the operator terminal (Figure 3.8). Becoming familiar with the features and basic operation of this device will prepare you for the specific programming procedures discussed in later paragraphs.

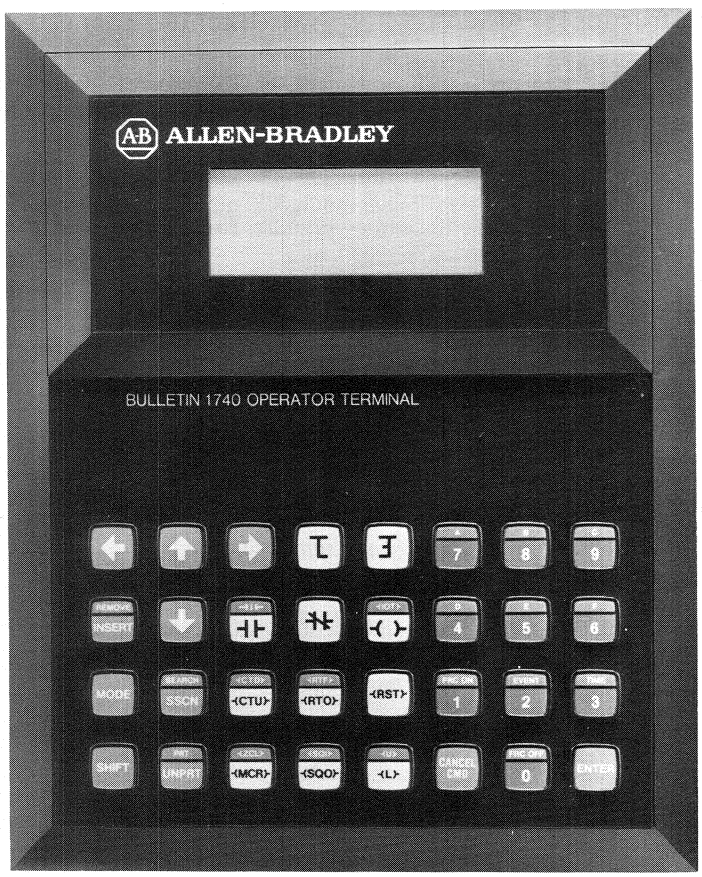


Figure 3.8 The operator terminal. Keys are color-coded according to function. The liquid crystal display features an adjustable contrast to suit your viewing preference.

STOP

Re-read paragraphs 3.1.4–3.1.7. Then answer the questions in Q/E Unit 3 of the Study Guide.

3.2.1 Keyboard – The operator terminal keyboard is shown in Figure 3.9. The keyboard includes 32 keys, which are color-coded according to function. These functional groups are pointed out in the figure. Key symbols and abbreviations are also briefly explained.

OPERATOR TERMINAL KEYBOARD

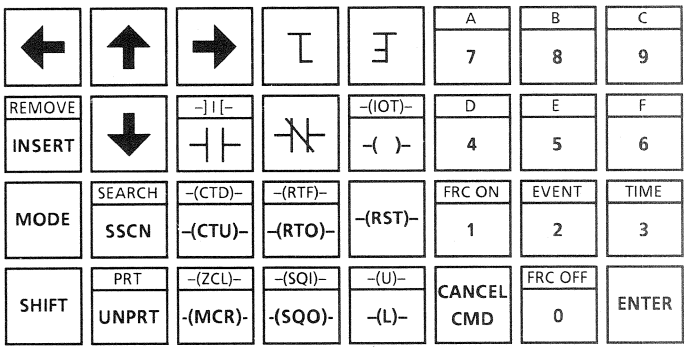
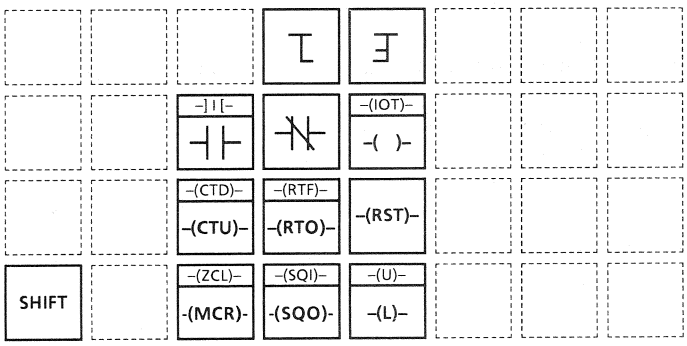
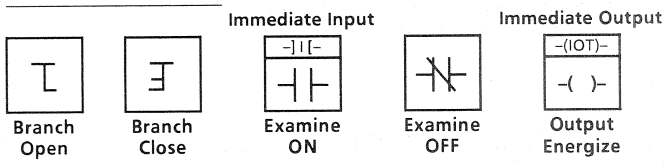


Figure 3.9(a) Operator terminal keyboard. It has 32 keys, color-coded according to function, as shown in Figures 3.9(b), (c), and (d).

WHITE KEYS



Explanation of Symbols

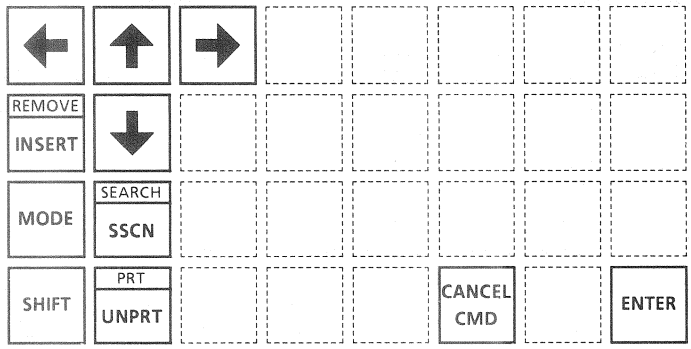


Abbreviations

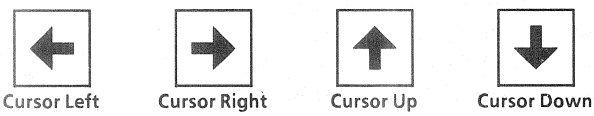
CTD – Down Counter	**SQI** – Sequencer Input
CTU – Up Counter	**SQO** – Sequencer Output
ZCL – Zone Control Last State	**RST** – Reset
MCR – Master Control Reset	**U** – Unlatch
RTF – Retentive Timer Off-Delay	**L** – Latch
RTO – Retentive Timer On-Delay	

Figure 3.9(b) White keys. Used for relay, timer, counter, and sequencer instructions. The Shift key and upper case functions (where applicable) are grey, indicating that the Shift key must be pressed to select these functions.

RED KEYS



Explanation of Symbols

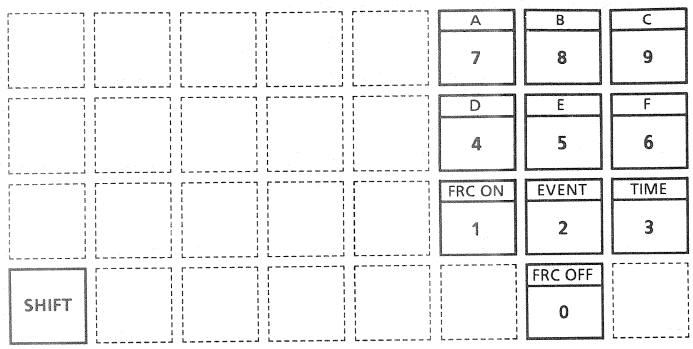


Abbreviations

SSCN – Single Scan	**UNPRT** – Not Protect
PRT – Protect	**CANCEL CMD** – Cancel Command

Figure 3.9(c) Red keys. Used for control, editing, and special functions. The upper case functions (where applicable) are grey, indicating that the Shift key must be pressed to select these functions.

BLUE KEYS



Abbreviations

FRC ON – Force On	**FRC OFF** – Force Off

Figure 3.9(d) Blue keys. Used for entering numerics, forcing, and sequencer data. The upper case functions are grey, indicating that the Shift key must be pressed to select these functions.

3.2.2 Display – The operator terminal display is divided into two functional areas. The upper area displays an entire ladder rung matrix, consisting of up to 3 parallel branches, each having up to 7 series instructions (not including the output).

The lower area is a message/data area, showing: the cursored instruction with its address number; timer, counter, and sequencer values; prompting messages to aid in programming; or error codes to aid in programming and troubleshooting. A display example is shown in Figure 3.10.

3.2.3 Error Codes – The error code numbers which appear on the operator terminal display are defined in Figure 3.11. This information also appears on the back of the operator terminal (Figure 3.12).

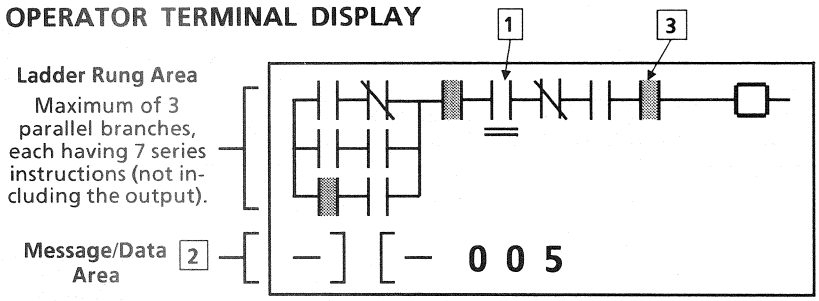
Internal processor errors (codes 01-08) are those occurring in the internal processor circuitry or the memory. When any of these errors occur, the Processor Fault LED on the processor module will light, and all outputs will be disabled. Remedies for these errors as well as error codes 20 and 21 are listed in Section 6 of the User's Manual.

Programming errors (codes 30-40) are user errors, indicating that you have used an invalid procedure, entered incorrect or incomplete data, etc. In most cases, the errors are easily understood and remedied; occasionally, it may be necessary to review the proper keystroke sequence for the particular programming operation you are attempting. Keystroke sequence examples appear later in this section.

The procedure for correcting programming errors is to 1) read the error description, 2) press the CANCEL COMMAND key, which returns the display to the point it was at before the error was made, then 3) take corrective action, as determined by the nature of the error.

3.2.4 Modes of Operation – The operator terminal is used to select the various processor *modes of operation.* A list of modes appears on the back of the operator terminal, as shown in Figure 3.12.

A detailed explanation of most modes is shown in Figure 3.13. You will want to refer to this figure again later as you become more familiar with programming and editing procedures.



1 Cursor – A blinking instruction in the ladder rung, represented here by underlining, indicates the cursor location. This identifies the instruction on which addressing, editing, and control functions can be carried out by the operator terminal. This cursored instruction together with its address (or prompt for an address) will appear in the lower, message/data area of the display, as shown in this example.

The cursor position can be moved to other instructions in the rung, or to instructions in the preceding or following rungs by pressing the appropriate cursor (arrow) keys.

2 Message/Data Area – Address numbers and timer/counter/sequencer data for the blinking, cursored instruction are shown here. This area also displays programming messages prompting for the next keystroke, as well as error codes to aid in programming and troubleshooting.

3 Status Indication – In the Test and Run modes, the display will give a TRUE/FALSE status indication for each instruction in the ladder rung. TRUE instructions in the rung are displayed in reverse video, represented here by shading. The remaining instructions in the rung are FALSE.

Figure 3.10 Typical operator terminal display.

ERROR CODE	DESCRIPTION
	INTERNAL PROCESSOR ERRORS
01 - 04	Processor Hardware Problem
05 - 08	Processor Memory Problem
20	Communication Error
21	EEPROM Module Cannot Be Programmed
	PROGRAMMING ERRORS
30	Invalid Keyboard Entry
31	Ladder Rung Matrix Exceeded
32	Incomplete Rung
33	Branch Error (Short Circuit Or Close Before Open)
34	Invalid Address For Instruction
35	User Memory Exceeded
36	Instruction Cannot Be Forced
37	No Force Present Or Instruction (For Removal)
38	Incomplete Data Entry
39	Program Mode Cannot Be Entered With EEPROM In Place
40	Invalid Mode Number

Figure 3.11 Error code descriptions.

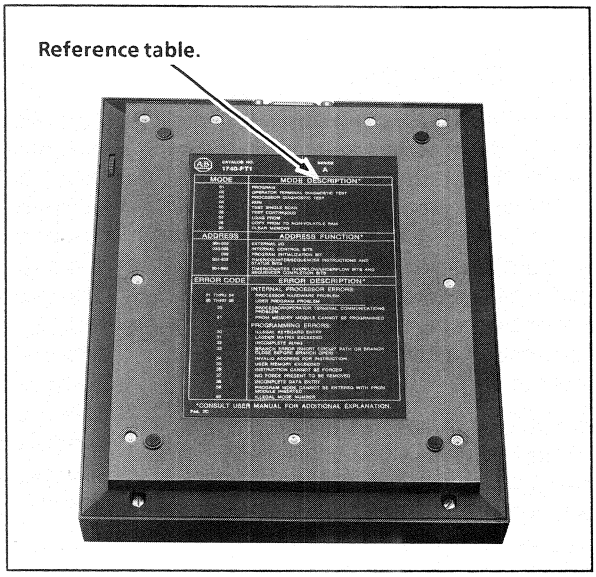


Figure 3.12 Back of operator terminal. Table includes information on modes, addresses, and error codes.

MODES OF OPERATION

Mode 01: PROGRAM. Used to enter a new program or update an existing one in the internal RAM memory.

Mode 02: DIAGNOSTIC TEST – OPERATOR TERMINAL. A sequence of self-checking diagnostic tests. Refer to Section 6 of the User's Manual for details.

Mode 03: DIAGNOSTIC TEST – PROCESSOR. A self-checking diagnostic test. Refer to Section 6 of the User's Manual for details.

Mode 04: RUN. In this mode, the processor scans and executes the user program. Input devices are monitored and output devices are energized accordingly. In this mode, the operator terminal can be used to monitor the user program, force I/O, and change timer and counter Preset and Accumulated values.

Mode 05: TEST – SINGLE SCAN. This mode causes the processor to complete a single scan of the user program each time the SSCN (Single Scan) key is pressed. No outputs will be energized. Timer/counter/sequencer Accumulated values will be incremented by 1 on each scan, if their rung conditions are true. The operator terminal can be used to monitor the user program, force I/O, and change counter/timer values.

Mode 06: TEST – CONTINUOUS SCAN. This mode causes the processor to operate from the user program without energizing any outputs. The operator terminal can be used to monitor the user program, force I/O, and change counter/timer values.

Mode 07: LOAD EEPROM MODULE. This mode allows you to duplicate the user program contained in the on-board RAM memory in an EEPROM memory module.

Mode 08: COPY EEPROM MODULE to PROCESSOR RAM. This mode allows you to duplicate the user program contained in an EEPROM module in the on-board RAM memory.

When completed, you can remove the EEPROM module and modify the program duplicated in the RAM memory as may be required.

Mode 20: CLEAR MEMORY. Selecting this mode erases the contents of the on-board RAM memory. Upon completion, a MEMORY CLEARED message will be displayed.

This mode does not affect EEPROM memory modules.

Figure 3.13 Operating modes.

STOP

Re-read paragraphs 3.2–3.2.4. Then answer the questions in Q/E Unit 4 of the Study Guide.

3.3 A HANDS-ON INTRODUCTION TO PROGRAMMING

The following paragraphs will show you how to use the operator terminal to select the program mode and then program a typical ladder rung. Later paragraphs explain the programming procedures for relay-type instructions, timer and counter instructions, sequencer instructions, and special instructions.

Keystroke examples are included to illustrate the programming steps. These examples show you which keys you should press ("Key Sequence" column), and what the operator terminal display should look like after you press the keys ("Display" column).

In some of the later keystroke examples, we have purposely omitted certain obvious steps to avoid needless repetition.

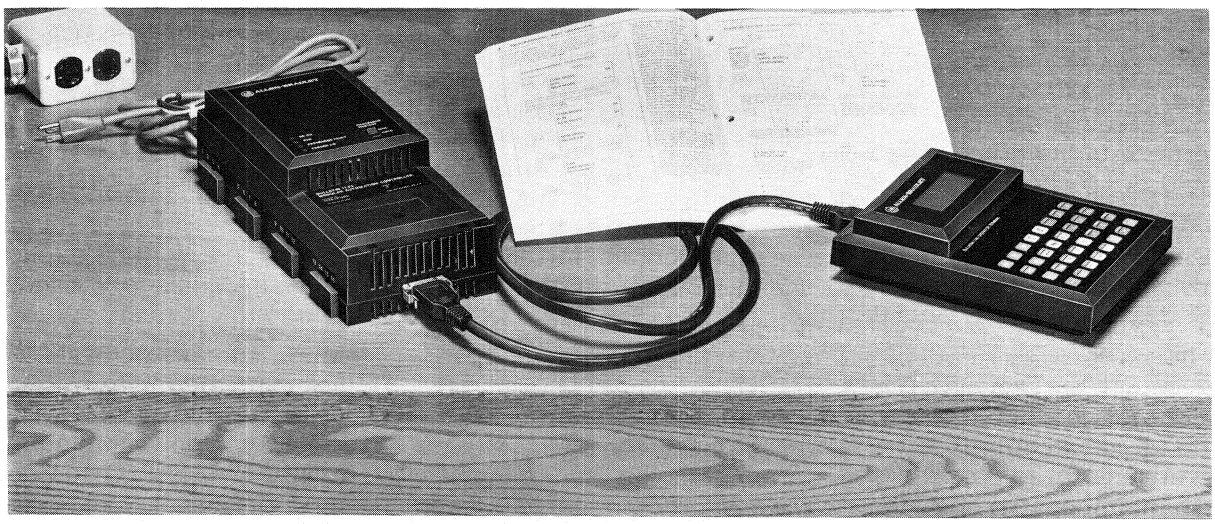
3.3.1 Connecting the Operator Terminal – Learning to program will be easier if you practice the keystroke examples with an energized operator terminal. The most convenient way to do this is to use the setup shown in Figure 3.14. The setup consists of a processor module (I/O modules won't be required) and the operator terminal. Just plug in the operator terminal cable, wire the processor module to an appropriate power source, turn on the power, and you'll be ready to practice.

When power is applied to the processor module, the DC ON indicator should light, indicating that DC voltage is being supplied to the internal logic circuitry. Simultaneously, the operator terminal will be energized and will automatically go thru a series of diagnostic checks; the terminal will then momentarily display the system diagnostic status and the operator terminal catalog number; finally, the display should show the following (assuming that the processor operating mode has not been changed since factory testing):

```
TO START
CLEAR RAM
PRESS ENTER
```

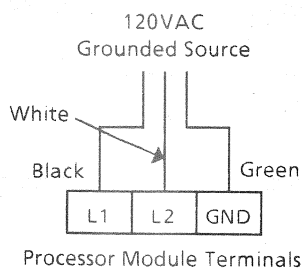
The operator terminal is telling you that the processor is in the Clear Memory mode (20), and you should press the ENTER key to initiate this function.

Before going further, make certain



Connection Procedure

1. Plug the interconnect cable into the processor module and operator terminal as shown. Cable connectors are keyed to guard against improper insertion. Make sure the spring latches are engaged to secure the cable.
2. For 120VAC power connection, we suggest you use a flexible power cord equipped with a 3-prong plug. Connect the wires as follows:



WARNING: Contact with the source voltage can cause serious injury. Do not apply power until the wiring is completed and checked.

3. Plug the cord into a 120VAC grounded outlet. If the outlet is not controlled by a nearby switch, power can be applied and removed by plugging and unplugging the power cord.

Figure 3.14 Setup for practicing keystroke examples.

that the display contrast suits your preference by turning the contrast adjustment knob on the operator terminal. This knob is located at the right edge of the terminal, accessible from the bottom.

Now press the ENTER key (red). The display should show:

```
CLEAR RAM
COMPLETE
PRESS MODE
```

The operator terminal is telling you that the processor RAM memory has been cleared, and you should press the MODE key (red). This will be the first step of the keystroke example of Figure 3.15.

3.3.2 Selecting the Program Mode – A keystroke example for entering the Program mode is shown in Figure 3.15. (Other modes are entered similarly.) The example begins where we left off in the last paragraph.

After the Program mode is entered, the display indicates that you are at the end of the user program; it also tells you how many words remain in the memory. In this particular case, the memory contents has just been cleared, so that you will actually *begin* your program at this point, and you have the full memory, 884 words, available to you.

Keystroke Example – Mode Selection

Key Sequence	Display
The operator terminal is asking you to press the MODE key.	CLEAR RAM COMPLETE PRESS MODE
MODE	CURR MODE 20 CLEAR RAM NEW MODE * *

After the MODE key is pressed, the terminal displays the current mode and prompts for a new mode number with two stars.

In this case, we want the Program mode, number 01:

Key Sequence	Display
1	CURR MODE 20 CLEAR RAM NEW MODE 01
ENTER	NEW MODE 01 PROGRAMMING

The terminal is verifying your entry. The ENTER key must be pressed again to activate the Program mode.

Key Sequence	Display
ENTER	END 0884 WORDS REMAINING

The processor is now in the Program mode. When you enter this mode, the operator terminal places you at the END of the program and tells you the number of words remaining (words available for your program). When you enter the Run and Test modes, the terminal places you at the START of your program.

Figure 3.15.

3.3.3 Programming a Typical Ladder Rung – After entering the Program mode, an instruction (white area of the keyboard) can be entered, beginning your first ladder rung. Remember that rungs begin with one or more condition instructions and end with a **single** output instruction.

Refer to Figure 3.16 for a step-by-step keystroke example for entering a typical ladder rung. The rung consists of an Examine ON, an Examine OFF, and an Output Energize instruction.

Note that after each instruction key is pressed, the instruction will appear in the upper part of the display, and it will be blinking. The instruction will also appear in the lower part of the display, followed by three stars (* * *), which prompt you to enter an address for the blinking instruction.

The address is entered with the numeric keys (blue area of the keyboard). Leading zeros of the address do not have to be entered (just press 1 when you want to enter address 001).

When entering condition instructions, you enter an instruction, its address, then another instruction, its address, and so on.

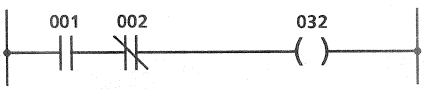
When entering an output instruction, you are completing the ladder rung. After entering the instruction address, you must press the ENTER key in order to enter the rung into the processor memory and go on to the next ladder rung.

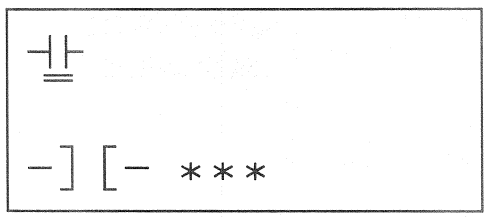
Ladder rungs are numbered 001, 002, etc. As you enter each rung, the display tells you what the next rung number will be.

Keystroke Example – Typical Ladder Rung

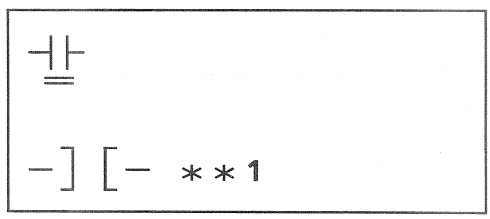
Enter this rung. It will be rung number 001 in your program.

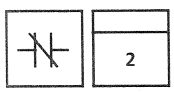
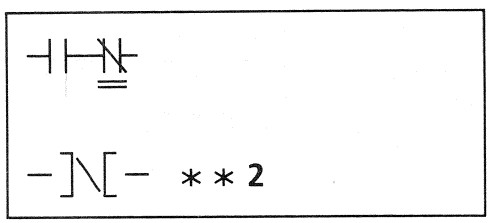
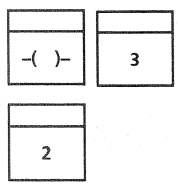
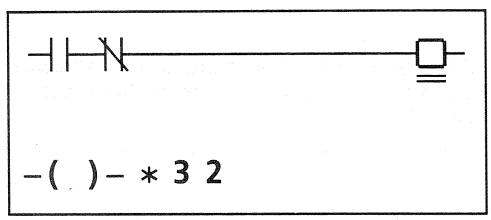
Ladder Diagram



Key Sequence	Display
This is the display which ended the mode selection example.	END 0884 WORDS REMAINING
	

Cursor location is shown in the display by the blinking symbol. It is indicated here by underlining. The three stars (*) are prompting you for an address. Leading zeros do not have to be entered.

Key Sequence	Display
The address has been entered. If you entered an incorrect address, you can change it at this time by repeatedly pressing the "zero" key until "000" appears, then enter the correct address. Editing keys could also be used, as you will learn later. The next instruction can now be entered.	
	
	

The ENTER key must now be pressed to enter the completed rung into RAM memory.

	END 0881 WORDS REMAINING

Note that after pressing ENTER, "RUNG 002" appeared on the display momentarily. This tells you that the present rung is entered, and the following rung will be the second rung of your program.

The "WORDS REMAINING" message appears after each rung entered. This is the unused fraction of memory.

Figure 3.16.

Re-read paragraphs 3.3–3.3.3. Then answer the questions in Q/E Unit 5 of the Study Guide.

3.4 RELAY TYPE INSTRUCTIONS

Relay type instructions are used for external I/O points (addresses 001-032), as well as for internal control (addresses 033-099). These instructions include:

- Examine ON, Examine OFF (condition instructions)
- Output Energize
- Branch Open, Branch Close (used to create parallel paths of condition instructions in a rung)
- Output Latch, Output Unlatch (retentive instructions)

3.4.1 Examine ON, Examine OFF, Output Energize – These instructions were already explained (paragraphs 3.1.5, 3.1.6). A simple program using them is shown in Figure 3.17. This figure shows a hard-wired circuit and a user program (not the only one possible) which provides the same results.

Programmed instructions with addresses 033 and 034 are *internal*. They do not represent external I/O devices. On the other hand, instructions 001, 002, 005, 006, and 007 represent limit switches LS1, LS2, and pilot lights PL1, PL2, PL3 respectively.

You will note similarities between the hard-wired circuit and the user program. Thus, when relay coil CR1 in the hard-wired circuit is energized, its normally open and normally closed contacts in rungs 3, 4, and 5 operate, or change state. Similarly, when Output Energize instruction 033 in the user program goes TRUE, its associated Examine ON and Examine OFF instructions in rungs 3, 4, and 5 change their TRUE/FALSE state.

In comparing the diagrams, you will also discover an apparent contradiction. That is, normally open limit switch LS1 in rung 1 and normally closed limit switch LS2 in rung 2 are *both* represented in the program by an Examine ON instruction. This makes sense when you consider that rungs 1 and 2 must accomplish the same thing: The rung must be TRUE when the external limit switch is closed, and FALSE when the limit switch is open. Using an Examine ON instruction to represent the limit switch satisfies these requirements. The N.O./N.C. mechanical action of the switch is not a consideration.

Unconditional Output Energize: It is possible to enter a rung consisting of an Output Energize instruction only. Since the rung lacks a condition instruction, the output will be continuously TRUE. Altho the unconditional Output Energize has a few specialized applications, it is generally an undesirable programming procedure which you should be careful to avoid.

3.4.2 Branch Instructions – These are used to create parallel paths of condition instructions, allowing more than one set of conditions (OR logic) to establish logical continuity in a rung.

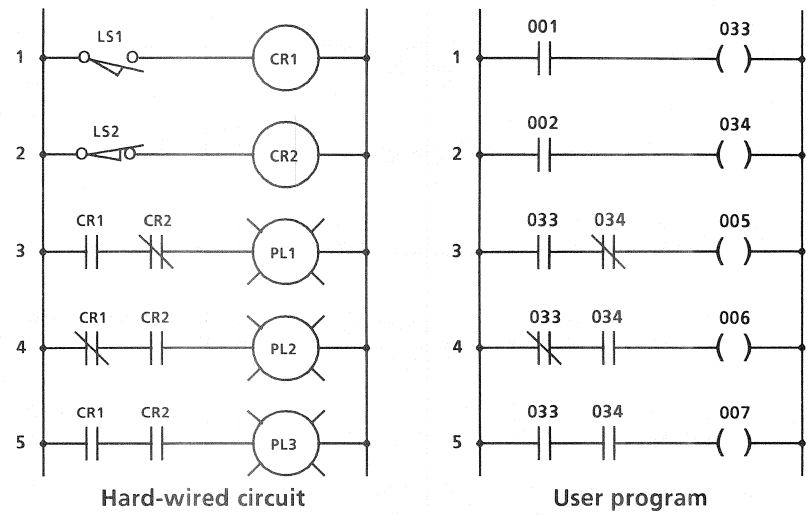
Figure 3.18 illustrates a simple branching condition. The rung will be TRUE if *either* instruction 001 *or* instruction 002 is TRUE.

Use branch instructions as follows:

Press the Branch Open key immediately before entering the first instruction of each parallel path. Press the Branch Close key after the last instruction of the last branch is entered.

Thus, the proper keystroke sequence for the example of Figure 3.18 is:

1. Branch Open.
2. Examine ON, address 001.
3. Branch Open.
4. Examine ON, address 002.
5. Branch Close.
6. Output Energize, address 005.
7. ENTER.



Limit switches LS1, LS2, and pilot lights PL1, PL2, and PL3 are represented in the user program by instructions 001, 002, 005, 006, and 007 respectively. Note that both the N.O. limit switch LS1 and the N.C. limit switch LS2 are represented by Examine ON instructions (001, 002).

Figure 3.17.

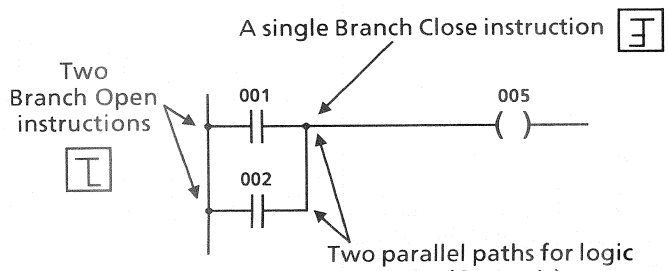


Figure 3.18.

NOTE: A nested branch (a branch within a branch) cannot be programmed directly, but it is possible to program a logically equivalent branching condition. This is illustrated in Figure 3.19.

A typical operator terminal display of a ladder rung with parallel branches is illustrated in Figure 3.20. The continuous path of TRUE condition instructions causes the Output to be TRUE. (The operator terminal display will indicate the TRUE/FALSE status of instructions whenever the processor is in the Run or Test operating mode).

3.4.3 Output Latch, Output Unlatch – These are retentive output instructions, used as a pair at the same address.

When a rung containing the Output Latch instruction goes TRUE, the Output Latch status bit is set to ON. This status bit will *remain* ON when logical continuity in the rung is lost. The Output Unlatch instruction, located in a second rung, is used to return the status bit to OFF. This is summarized in Figure 3.21.

The Latch/Unlatch instruction status is retained when the controller is changed from the Run mode to another mode. Thus, in the example of Figure 3.21, if the controller is changed from the Run mode to any other mode while status bit 015 is ON, the external output device will be de-energized, but the status bit will remain ON; when returning the controller to the Run mode, the output device will again be energized.

The Latch/Unlatch instruction status is also retained on power loss, since the processor memory is provided with battery back-up.

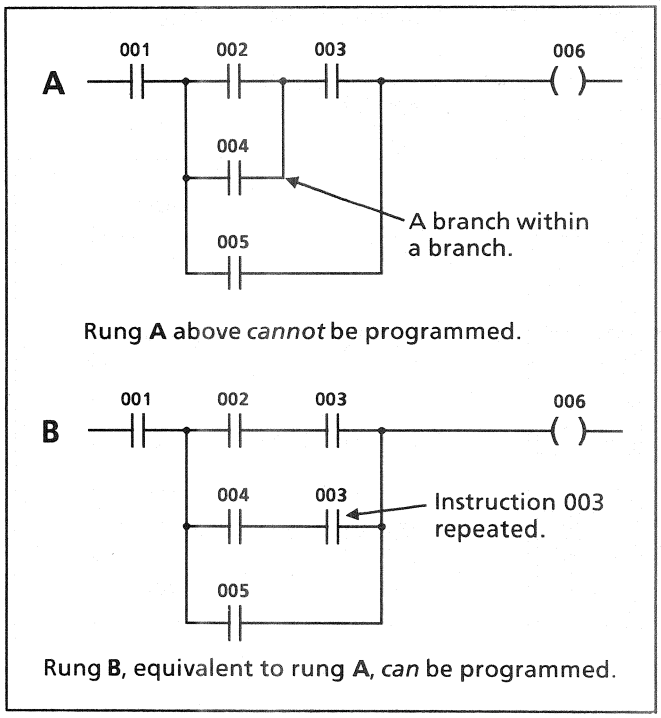


Figure 3.19 Rung with nested branch and equivalent branching.

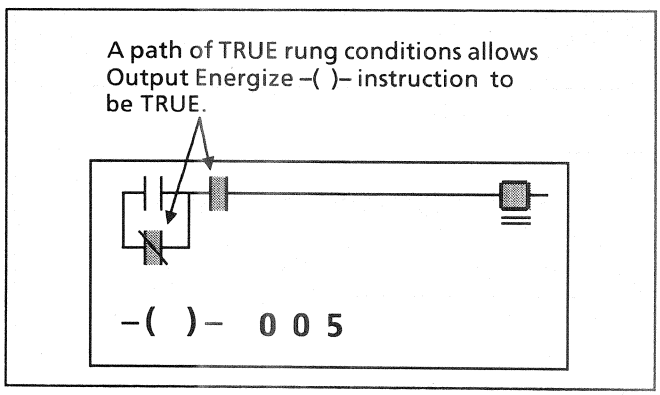
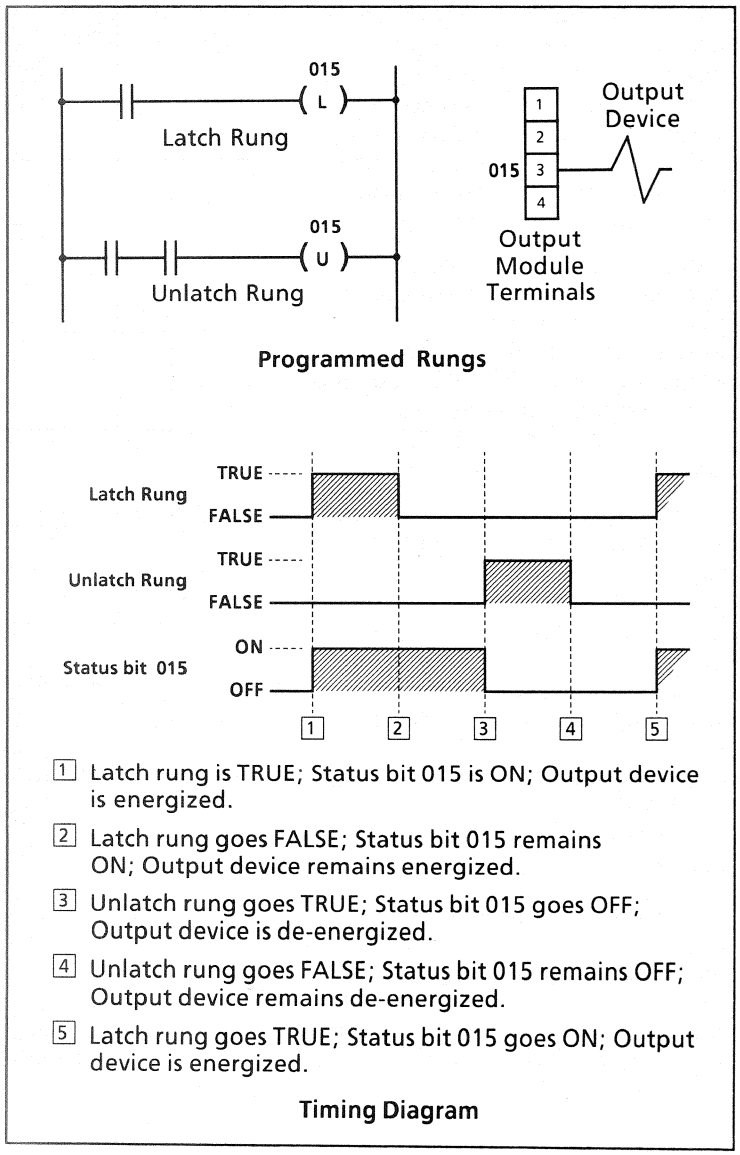


Figure 3.20 Operator terminal display of a rung with parallel branches. TRUE rung conditions are indicated.



1 Latch rung is TRUE; Status bit 015 is ON; Output device is energized.

2 Latch rung goes FALSE; Status bit 015 remains ON; Output device remains energized.

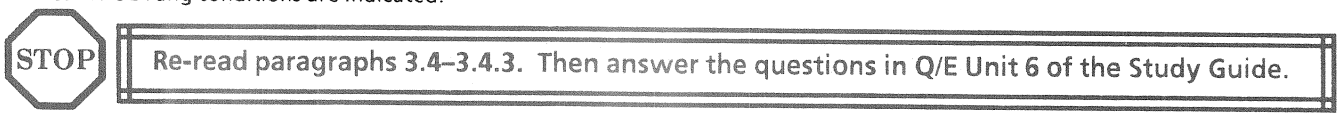
3 Unlatch rung goes TRUE; Status bit 015 goes OFF; Output device is de-energized.

4 Unlatch rung goes FALSE; Status bit 015 remains OFF; Output device remains de-energized.

5 Latch rung goes TRUE; Status bit 015 goes ON; Output device is energized.

Timing Diagram

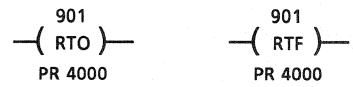
Figure 3.21 The Latch/Unlatch instruction.



3.5 TIMER INSTRUCTIONS

Timer instructions include the Retentive Timer On-Delay –(RTO)– and the Retentive Timer Off-Delay –(RTF)–. Both require the use of the Reset instruction –(RST)–. Figure 3.22 indicates some timer characteristics you should become familiar with.

RTO and RTF Timer Instructions:



Address: 901-932 (internal).

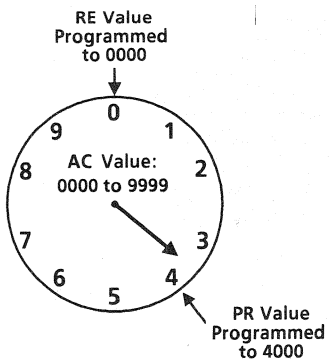
The timer instruction functions as an internal "clock", counting 0.1-second intervals. The number of intervals counted is called the *Accumulated value* (AC).

Counting takes place under these TRUE/FALSE rung conditions:

RTO Timer Rung Conditions		
TRUE	**FALSE**	**TRUE**
Timer is Counting	Counting Stops. AC value retained	Counting Resumes
AC value represents the cumulative time during which rung is TRUE.		

RTF Timer Rung Conditions		
FALSE	**TRUE**	**FALSE**
Timer is Counting	Counting Stops. AC value retained	Counting Resumes
AC value represents the cumulative time during which rung is FALSE.		

The following clock figure represents the AC value. The time delay is set by programming a *Preset value* (PR) and a *Reset value* (RE). In this particular case, the RE value is set at 0000 and the PR value is set at 4000.



The time delay is represented by the PR value minus the RE value.

In this figure the RE value is 0000, so that the PR value of 4000 represents the time delay.

The range of timers is 0.1 to 999.9 seconds (0001 to 9999 counts). A PR value of 4000 then represents: 4000 x 0.1 = 400.0 seconds.

Status and Overflow Bits:

The timer *status bit* has the same address as the timer instruction.

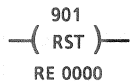
The RTO status bit is set ON when the AC value reaches the PR value. Examine ON instructions at the status bit address go TRUE; Examine OFF instructions go FALSE.

The RTF status bit is set OFF when the AC value reaches the PR value. Examine ON instructions at the status bit address go FALSE; Examine OFF instructions go TRUE.

The *overflow bit* is given the timer address plus 50. (901 + 50 = 951).

The overflow bit is set to ON when the AC value "overflows" from a count of 9999 to 0000. Examine ON instructions at the overflow bit address go TRUE; Examine OFF instructions go FALSE.

Reset Instruction:



The reset (RST) instruction is given the same address as the timer instruction. When the RST instruction goes TRUE, status bits and overflow bits return to their previous states, and the AC value is reset to the RE value. The RST instruction must go FALSE again before the timer can resume counting.

Note: The operator terminal display does not show the RST instruction in reverse video when it is TRUE.

Typical Timer Operation:

Status Bit 901 / Overflow Bit 951 (Examine ON Instructions)	1. Timer Reset AC = RE	2. RTO Rung TRUE or RTF Rung FALSE. AC Value Increments		3. AC Value Overflows Beyond Count of 9999		4. RST Rung TRUE. Timer Reset
	AC<PR	AC<PR	AC ≥ PR	AC<PR	AC ≥ PR	AC<PR
RTO Status Bit	OFF	OFF	ON	ON	ON	OFF
RTF Status Bit	ON	ON	OFF	OFF	OFF	ON
Overflow Bits	OFF	OFF	OFF	ON	ON	OFF

Power Down: The timer status is retentive. When you apply power after a power-down, the AC value and the ON/OFF states of status and overflow bits will be the same as before power was removed. The status is also retained when going from the Run mode to another mode.

Monitoring and Changing Data: PR and AC values can be monitored and changed in the Run and Test modes. The PR value can be protected from changes in the Run and Test modes by using the UNPRT/PRT key. Refer to Figure 3.26.

Figure 3.22 Characteristics of timer instructions.

3.5.1 On-Delay Timer (RTO) – As pointed out in Figure 3.22, the AC value of the RTO timer increments when the RTO rung is TRUE. This is illustrated in Figure 3.23, which shows an RTO timer ladder diagram and the corresponding timing diagram.

For illustrative purposes, both Examine ON and Examine OFF instructions are used at the timer status bit address, 901. These instructions are used to energize output 005 and de-energize output 006 when the AC value reaches the programmed PR value.

Letters **A** to **F** at the bottom of the timing diagram indicate the following events:

A. Rung 1 goes TRUE. The AC value increments, beginning at AC = RE. Examine ON instruction 901 in rung 2 is FALSE. Examine OFF instruction 901 in rung 3 is TRUE.

B. Rung 1 goes FALSE. The AC value stops incrementing, but the existing value is retained.

C. Rung 1 goes TRUE. The AC value continues to increment from the point it left off.

D. The AC value has reached the PR value. Examine ON instruction 901 goes TRUE, making rung 2 TRUE. Examine OFF instruction 901 goes FALSE, making rung 3 FALSE. The AC value continues to increment.

E. Rung 1 goes FALSE. The AC value stops incrementing, but the existing value is retained. Rung 2 remains TRUE. Rung 3 remains FALSE.

F. Rung 4 goes TRUE. The AC value is reset to the programmed RE value. Examine ON instruction 901 goes FALSE, making rung 2 FALSE. Examine OFF instruction 901 goes TRUE, making rung 3 TRUE. While the RST instruction is TRUE, the timer is disabled. Counting can resume when rung 4 goes FALSE again.

These timing diagram events illustrate the retentive nature of the timer. In effect, the timer measures the cumulative periods during which the timer rung (1) is TRUE. The AC value will continue to increment during TRUE rung periods as long as the reset rung (4) remains FALSE. On the count after 9999, an overflow bit (not used here) is set ON and the AC value continues to increment from 0000.

After the timer is reset and rung 1 goes TRUE again, the timing period will begin at AC = RE. Of course, the RE value could have been programmed to zero. The choice depends on the particular application.

3.5.2 Off-Delay Timer (RTF) – The essential difference between the RTO and RTF timers is that with the RTF timer, the AC value increments when the timer rung goes FALSE rather than TRUE.

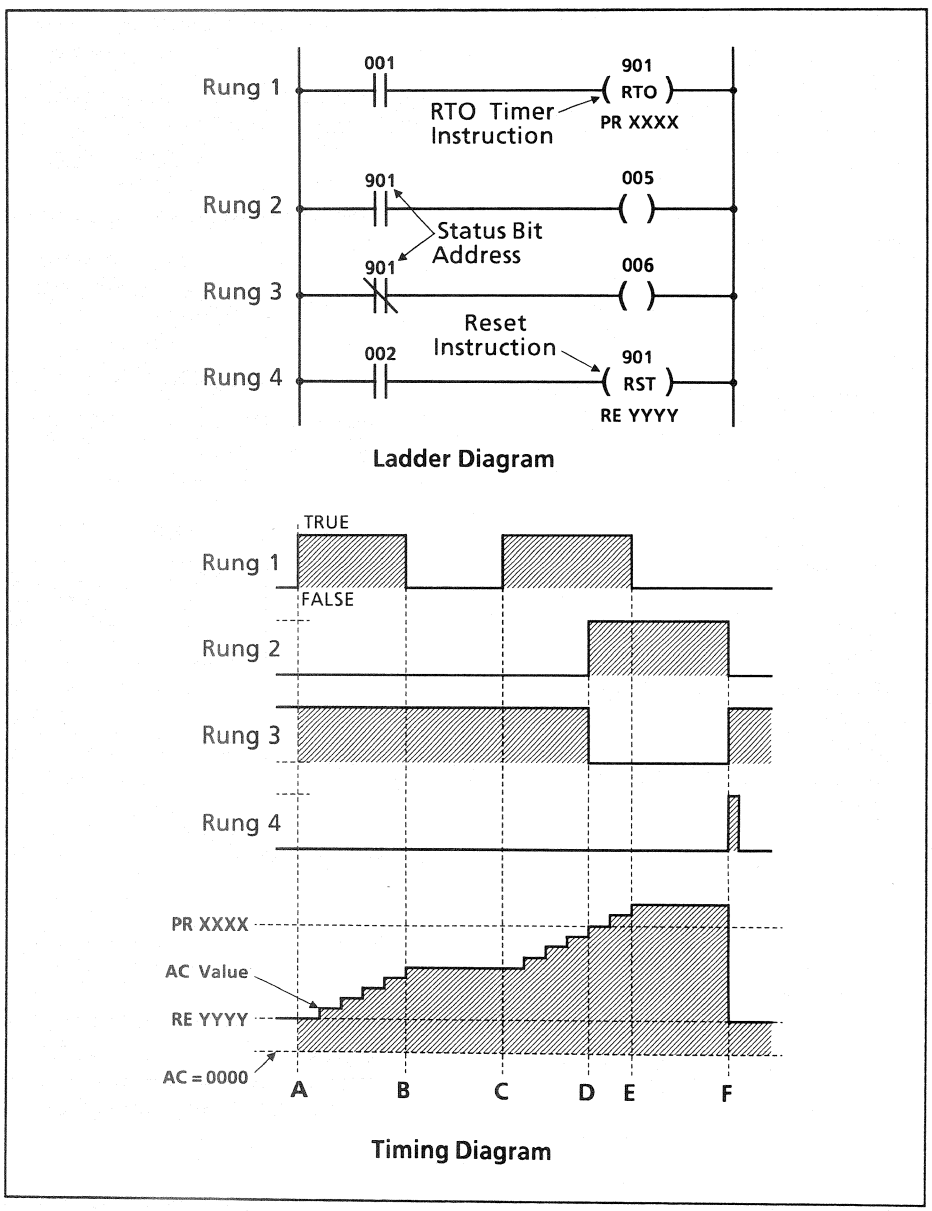


Figure 3.23 The RTO instruction.

Figure 3.24a shows an RTF timer ladder diagram and the corresponding timing diagram. For illustrative purposes, both Examine ON and Examine OFF instructions are used at the timer status bit address, 901. These instructions are used to de-energize output 005 and energize output 006 when the AC value reaches the programmed PR value.

Letters **A** to **F** at the bottom of the timing diagram indicate the following events:

A. Rung 1 is FALSE, and the AC value is incrementing from AC = RE. Examine ON instruction 901 in rung 2 is TRUE. Examine OFF instruction 901 in rung 3 is FALSE. (This condition might occur on power-up.)

B. Rung 1 goes TRUE. The AC value stops incrementing, but the existing value is retained.

C. Rung 1 goes FALSE. The AC value continues to increment from the point it left off.

D. The AC value has reached the PR value. Examine ON instruction 901 goes FALSE, making rung 2 FALSE. Examine OFF instruction 901 goes TRUE, making rung 3 TRUE. The AC value continues to increment.

E. Rung 1 goes TRUE. The AC value stops incrementing, but the existing value is retained. Rung 2 remains FALSE. Rung 3 remains TRUE.

F. Rung 4 goes TRUE. The AC value is reset to the programmed RE value. Examine ON instruction 901 goes TRUE, making rung 2 TRUE. Examine OFF instruction 901 goes FALSE, making rung 3 FALSE.

While the RST instruction is TRUE, the timer is disabled. Counting can resume when rung 4 goes FALSE again.

Here again, the timing diagram events illustrate the retentive nature of the timer. In this case, the timer measures the cumulative periods during which the timer rung (1) is FALSE. The AC value will continue to increment during FALSE rung periods as long as the reset rung (4) remains FALSE. On the count after 9999, an overflow bit (not used here) is set ON and the AC value continues to increment from 0000.

After the timer is reset and rung 1 goes FALSE again, the timing period will begin at AC = RE.

Correct rung order for programs which include the RTF timer: The RTF timer and reset rungs must be located as shown in Figure 3.24b if your program includes RTO timers, CTU counters, or CTD counters in addition to RTF timers.

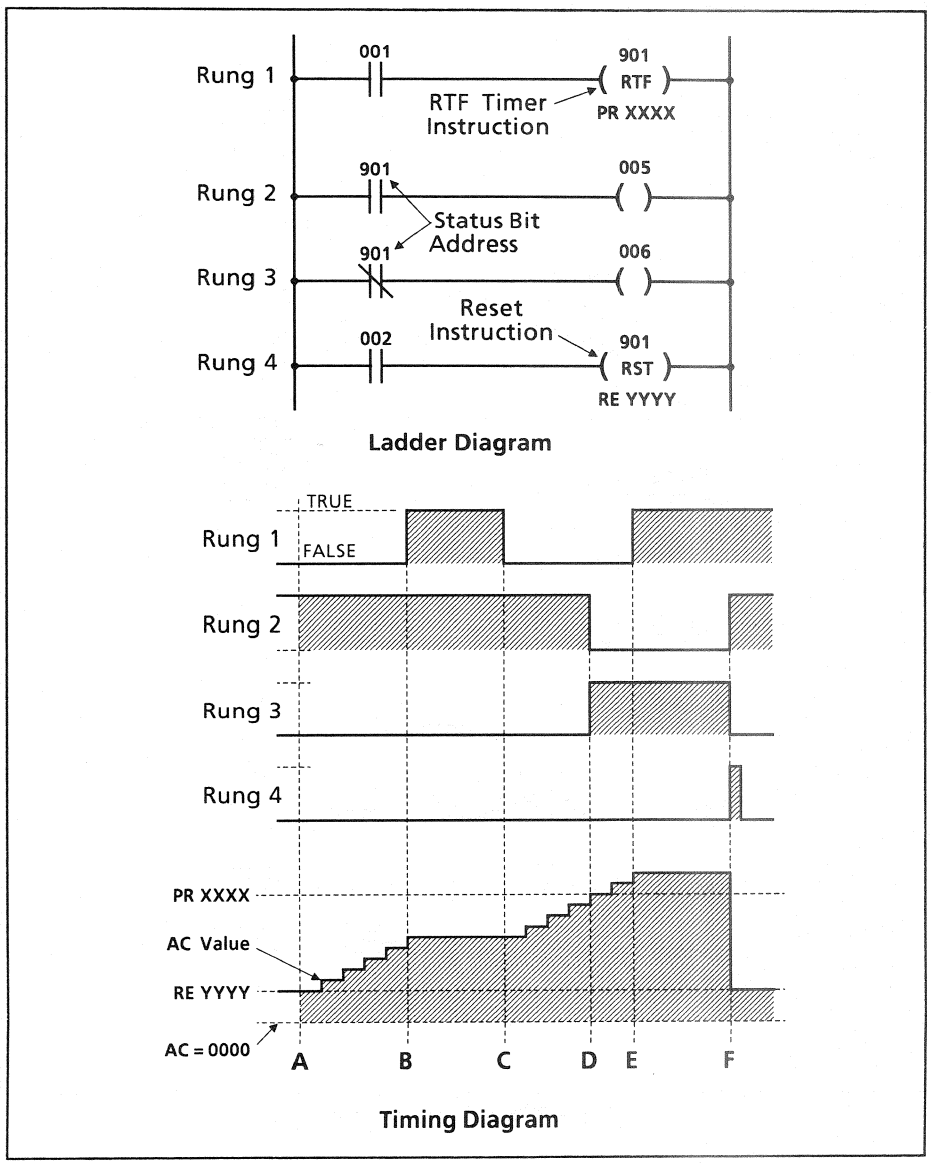


Figure 3.24a The RTF instruction.

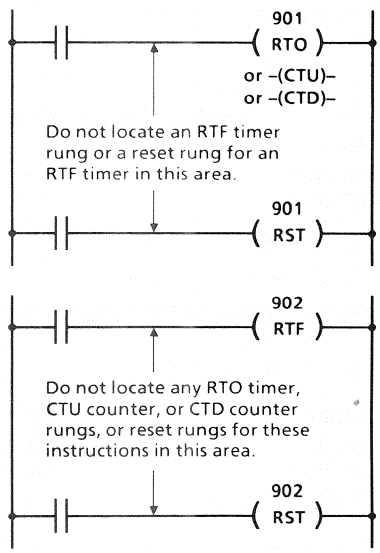


Figure 3.24b.

3.5.3 Timing Beyond 999.9 Seconds – If you require time delay periods beyond 999.9 seconds, one way of accomplishing it is to use the overflow bit of one timer to initiate a second timer. This is done in the example of Figure 3.25, which shows an RTO timer programmed to produce an output at 1400 seconds.

Timer 901 in rung 1 is used for the first part of the timing period and timer 902 in rung 2 is used for the last part. The Examine ON instruction 001 of rung 1 is repeated in rung 2, so that if timing is interrupted during the last part of the timing period, the AC value will stop incrementing, preventing an output in rung 3.

Letters **A** to **E** at the bottom of the timing diagram indicate the following events:

A. Examine ON instructions 001 in rungs 1 and 2 go TRUE. Rung 1 goes TRUE. AC value of timer 901 increments from 0000.

B. AC value of timer 901 overflows beyond 9999, setting the overflow bit (address 951) ON. Examine ON instruction 951 in rung 2 goes TRUE, making rung 2 TRUE. AC value of timer 902 increments from 0000.

1000 seconds have elapsed.

C. AC value of timer 902 reaches programmed PR value of 4000, setting the status bit (address 902) ON. Examine ON instruction 902 in rung 3 goes TRUE, making rung 3 TRUE.

Total time elapsed:
1000 + 400 = 1400 seconds.

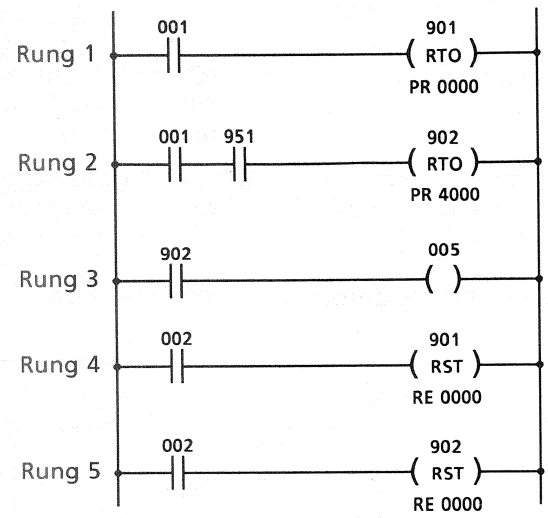
D. Instructions 001 in rungs 1 and 2 go FALSE, making rungs 1 and 2 FALSE. Rung 3 remains TRUE.

E. Reset rungs 4 and 5 go TRUE. Examine ON instructions 902 and 951 go FALSE. AC values are reset to 0000.

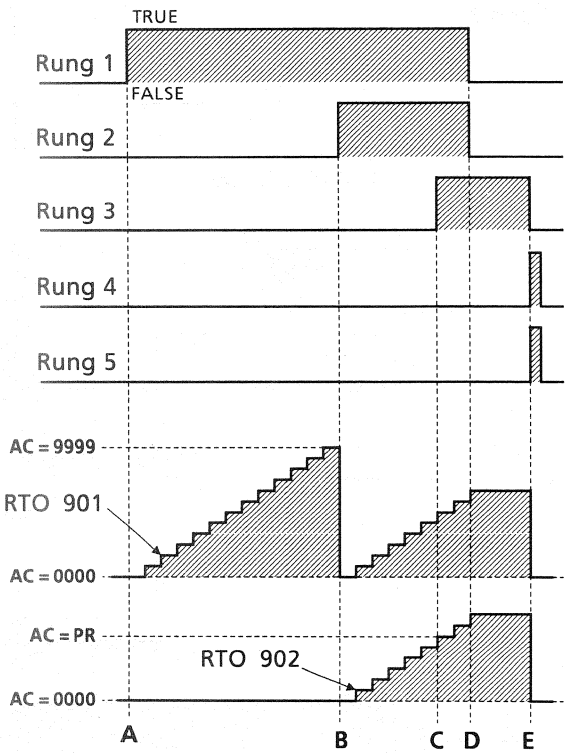
While the RST instructions are TRUE, the timer is disabled. Counting can resume when rungs 4 and 5 go FALSE again.

Another way of obtaining time delays exceeding 999.9 seconds is to use the ladder diagram of Figure 3.30 (Page 3-21), substituting a timer 901 for counter 901. By programming the PR value of timer 901 to 4000, the status bits of timer 901and counter 902 would both be TRUE and produce an output at 1400 seconds.

3.5.4 Timer Keystroke Example – Figure 3.26 shows you how to program the basic RTO timer. The keystokes required for programming the RTF timer are similar.



Ladder Diagram



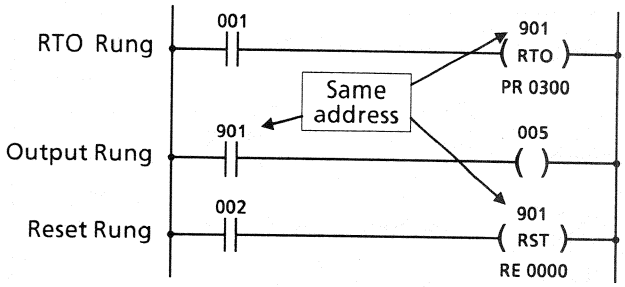
Timing Diagram

Figure 3.25 Cascading timers.

Keystroke Example – RTO Instruction

(RTF, CTU, CTD instructions are entered this same way)

Program these rungs for the basic RTO timer:



Keystrokes are shown for the RTO and RST instructions only, to illustrate how you enter PR and RE values.

Key Sequence	Display
The RTO instruction: –(RTO)–	(RTO) * * *
When the RTO key is pressed, the rung is completed with the output instruction symbol. Stars prompt for address.	
9, 0, 1	(RTO) 9 0 1
ENTER	PRESET * * * *
Stars prompt for a preset (PR) value.	
3, 0, 0, ENTER	PRESET 0 3 0 0 U

The "U" after the value means the preset is unprotected. An unprotected PR value can be changed in the Run or Test mode, whereas a protected PR value *cannot* be changed in the Run or Test mode. If you choose to protect the PR value, press SHIFT + PRT

Key Sequence	Display
ENTER	END XXXX WORDS REMAINING
The RST instruction: –(RST)–	(RST) * * *
When the RST key is pressed, the rung is completed. Stars prompt you for an address. Enter the same address as you entered for the RTO instruction and its status bit:	
9, 0, 1	(RST) 9 0 1
ENTER	RESET * * * *
Stars prompt for a reset (RE) value. The timer AC value will be set to this RE value when the reset rung goes TRUE. We are selecting an RE value of zero in this case.	
Note: The RST instruction is not displayed in reverse video when it is TRUE.	
0	RESET * * * 0
ENTER	END XXXX WORDS REMAINING

Figure 3.26.

STOP **Re-read paragraphs 3.5–3.5.4. Then answer the questions in Q/E Unit 7 of the Study Guide.**

3.6 COUNTER INSTRUCTIONS

Counter instructions include the Up Counter –(CTU)– and the Down Counter –(CTD)– Both are retentive, requiring the use of the Reset instruction.

Figure 3.27 indicates some counter characteristics you should become familiar with.

CTU and CTD Counter Instructions:

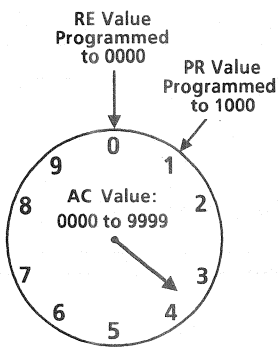
Address: 901-932 (internal).

Counter instructions count successive FALSE-to-TRUE transitions of the rung containing the counter instruction. After each count, the rung must return to FALSE before another count can take place.

The existing count is called the *Accumulated value* (AC). For up counters, the AC value increases by 1 for each FALSE-TRUE transition. For down counters, the AC value decreases by 1 for each FALSE-TRUE transition.

Up-down counters have both an up-counter rung and a down-counter rung. Accordingly, the AC value both increases and decreases, responding to FALSE-TRUE transitions in both rungs.

The clock figure below represents the AC value. An output can be obtained at a particular count by programming a *Preset value* (PR) and a *Reset value* (RE). In this case, the RE value is set at 0000 and the PR value is set at 1000.



Setting the Counter: The range of counters is 1 to 9999 counts. Up counters begin to count up at AC = RE and produce an output at AC = PR. Thus, to obtain an output at 1000 counts, you can set the PR value to 1000 and the RE value to 0000. Or, since RE needn't be set to 0000, you might set the PR value at 1500 and the RE value at 0500, etc.

Up-down counters add and subtract counts, starting at AC = RE, and produce an output at AC = PR. Thus, if you set RE at 500 and PR at 1500, an output will be produced whenever the up-counts exceed the down-counts by 1000.

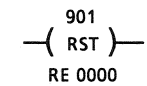
Status and Overflow/Underflow Bits:

The counter *status bit* has the same address as the counter instruction.

The status bit is set ON when the AC value reaches the PR value. Examine ON instructions at the status bit address go TRUE; Examine OFF instructions go FALSE.

Overflow and *underflow bits* are assigned the counter address plus 50. (901 + 50 = 951.) The overflow bit is set ON when the AC value "overflows" from a count of 9999 to 0000. The underflow bit is set ON when the AC value "underflows" from 0000 to 9999. Examine ON instructions at the overflow/underflow bit address go TRUE; Examine OFF instructions go FALSE.

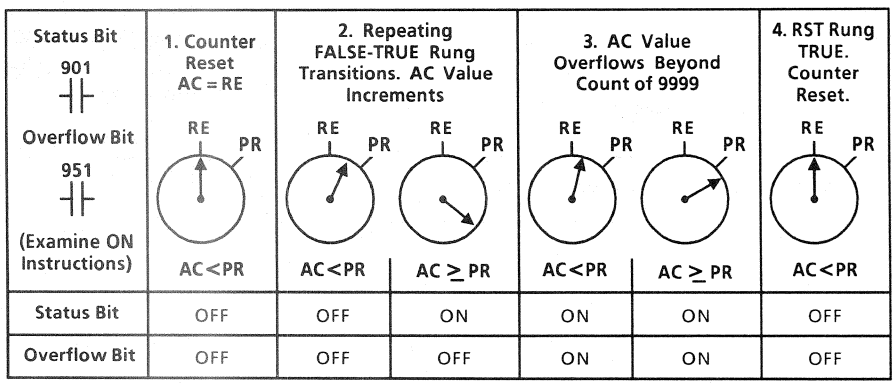
Reset Instruction:



The reset (RST) instruction is given the same address as the counter instruction. When the RST instruction goes TRUE, the status, overflow, and underflow bits are reset to OFF, and the AC value is reset to the RE value. The RST instruction must go FALSE again before the counter can resume counting.

Note: The operator terminal display does not show the RST instruction in reverse video when it is TRUE.

Typical Up Counter Operation:



Power Down: The counter status is retentive. When you apply power after a power-down, the AC value and the ON/OFF states of status and overflow/underflow bits will be the same as before power was removed. The counter status is also retained when going from the Run mode to another mode.

Monitoring and Changing Data: PR and AC values can be monitored and changed in the Run and Test modes. The PR value can be protected from changes in the Run and Test modes by using the UNPRT/PRT key. Refer to Figure 3.26.

Figure 3.27 Characteristics of counter instructions.

3.6.1 Up Counter (CTU) – As pointed out in Figure 3.27, the AC value of the CTU counter increases by one for each FALSE-to-TRUE transition occurring in the CTU counter rung. This is illustrated in Figure 3.28, which shows a CTU counter ladder diagram and the corresponding timing diagram.

For illustrative purposes, both Examine ON and Examine OFF instructions are used at the counter status bit address, 901.

These instructions are used to energize output 005 and de-energize output 006 when the AC value reaches the PR value.

Letters **A** to **E** at the bottom of the timing diagram indicate the following events:

A. The AC value equals the RE value. Rung 1 is FALSE. Examine ON instruction 901 in rung 2 is FALSE. Examine OFF instruction 901 in rung 3 is TRUE.

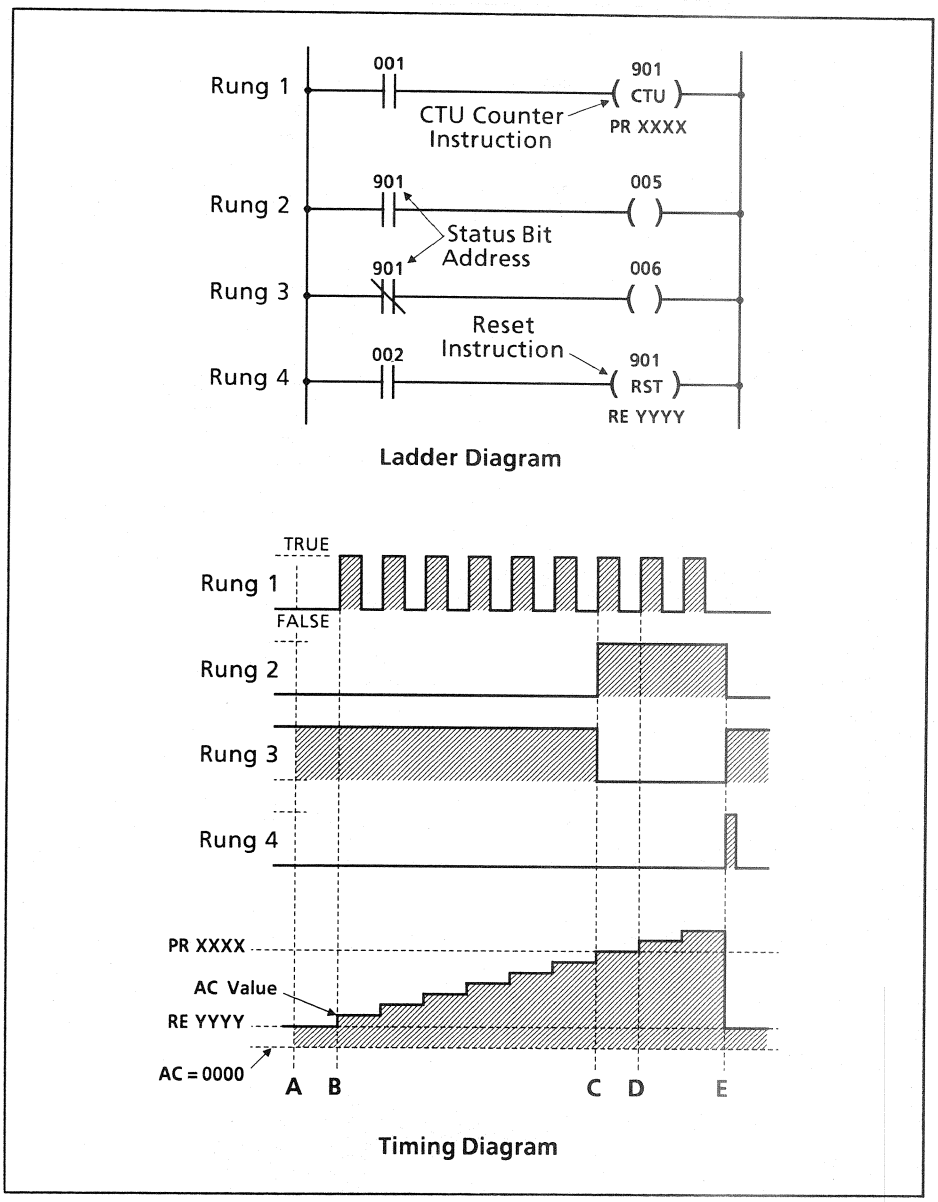


Figure 3.28 The CTU instruction.

B. Rung 1 goes from FALSE to TRUE, causing the AC value to increase by one. The rung then returns to FALSE. This cycle repeats again and again, with the AC value increasing by one count for each FALSE-TRUE transition.

C. The AC value has reached the programmed PR value. Examine ON instruction 901 goes TRUE, making rung 2 TRUE. Examine OFF instruction 901 goes FALSE, making rung 3 FALSE.

D. The AC value continues to increase for each FALSE-TRUE transition of rung 1. Rung 2 remains TRUE. Rung 3 remains FALSE.

E. Rung 4 goes TRUE. The AC value is reset to the programmed RE value. Examine ON instruction 901 goes FALSE, making rung 2 FALSE. Examine OFF instruction 901 goes TRUE, making rung 3 TRUE.

While the RST instruction is TRUE, the counter is disabled. Counting can resume when rung 4 goes FALSE again.

If the counter is not reset as was done in event **E**, the AC value continues to increase for FALSE-TRUE counter rung transitions. On the count of 9999, an overflow bit (not used here) is set ON and the AC value continues to increment from 0000.

3.6.2 Up-Down Counter – This counter includes a CTU instruction and a CTD instruction which share the same address (including status and overflow/underflow addresses). AC, PR, and RE values are also shared. This is illustrated in Figure 3.29, which shows an up-down counter programmed to produce two outputs:

The first output is energized when the AC value reaches 9000, then de-energized when the AC value overflows beyond 9999. In terms of counts, this output is energized only when the net count (up-counts minus down-counts) is within the range of 3000 to 3999.

The second output is energized when the AC value overflows beyond 9999, that is, when the net count is 4000 or more.

In this example, we assume that the AC value does not underflow below 0000.

Letters **A** to **F** at the bottom of the timing diagram indicate the following events:

A. CTU counter in rung 1 begins a series of up-counts, beginning at AC = RE = 6000. Examine ON instruction 901 in rung 3 is FALSE. Examine OFF instruction 951 in rung 3 is TRUE. Examine ON instruction 951 in rung 4 is FALSE.

B. The AC value has reached the programmed PR value of 9000, setting the status bit (address 901) ON. Examine ON instruction 901 in rung 3 goes TRUE, making rung 3 TRUE.

The net count is 3000.

C. CTD counter in rung 2 is down-counting. The AC value is now less than the PR value of 9000. Examine ON instruction 901 in rung 3 goes FALSE, making rung 3 FALSE.

The net count has dropped below 3000.

D. Rung 1 is up-counting, and the AC value has reached the PR value of 9000. Examine ON instruction 901 goes TRUE, making rung 3 TRUE.

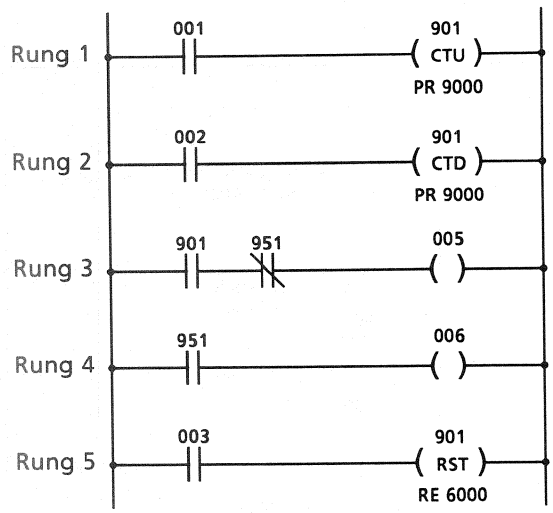
The net count is 3000 again.

E. The AC value overflows beyond 9999, setting the overflow bit (address 951) ON. Examine ON instruction 951 in rung 4 goes TRUE, making rung 4 TRUE. Examine OFF instruction 951 in rung 3 goes FALSE, making rung 3 FALSE.

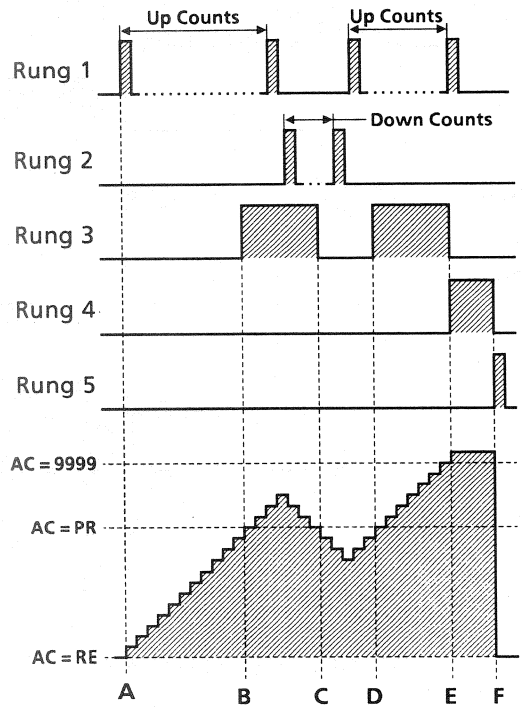
The net count is 4000.

F. Rung 5 goes TRUE. The AC value is reset to the programmed RE value of 6000. Status bit and overflow bit are reset to OFF. Rung 4 goes FALSE.

While the RST instruction is TRUE, the counter is disabled. Counting can resume when rung 5 goes FALSE again.



Ladder Diagram



Timing Diagram

Figure 3.29 Up-down counter.

3.6.3 Counting beyond 9999 – When it is necessary to count beyond 9999, one way of accomplishing it is by programming the counter shown in Figure 3.30.

In this figure, the overflow bit of counter 901 initiates a single count in counter 902 at a count of 10,000. The status bits of both counters are programmed in series to produce an output when an additional 3000 counts are made by counter 901.

Letters **A** to **E** at the bottom of the timing diagram indicate the following events:

A. Counter 901 begins a series of counts, beginning at AC = RE = 0000. Examine ON instructions 951 in rungs 2 and 3 are FALSE. Examine ON instructions 901 and 902 in rung 4 are FALSE.

B. The AC value has reached counter 901 PR value of 3000. Examine ON instruction 901 in rung 4 goes TRUE, but rung 4 remains FALSE.

C. The AC value of counter 901 overflows beyond 9999, setting the overflow bit (address 951) ON. This causes the following sequence of events:

Examine ON instruction 951 in rung 3 goes TRUE, resetting counter 901. Examine ON instruction 901 in rung 4 goes FALSE; Examine ON instruction 951 in rung 3 goes FALSE; Counter 901 continues to count from AC = 0000.

Simultaneously, Examine ON instruction 951 in rung 2 has gone TRUE, then FALSE, causing the AC value of counter 902 to go from 0000 to the PR value of 0001. Examine ON instruction 902 in rung 4 goes TRUE.

10,000 counts have occurred.

D. The AC value of counter 901 has again reached the PR value of 3000. Examine ON instruction 901 in rung 4 goes TRUE, making rung 4 TRUE.

13,000 counts have occurred.

E. Rungs 5 and 6 go TRUE. The AC values of both counters are reset to the RE value. Status bits and overflow bits are reset to OFF. Rung 4 goes FALSE.

While the RST instructions are TRUE, the counter is disabled. Counting can resume when rungs 5 and 6 go FALSE again.

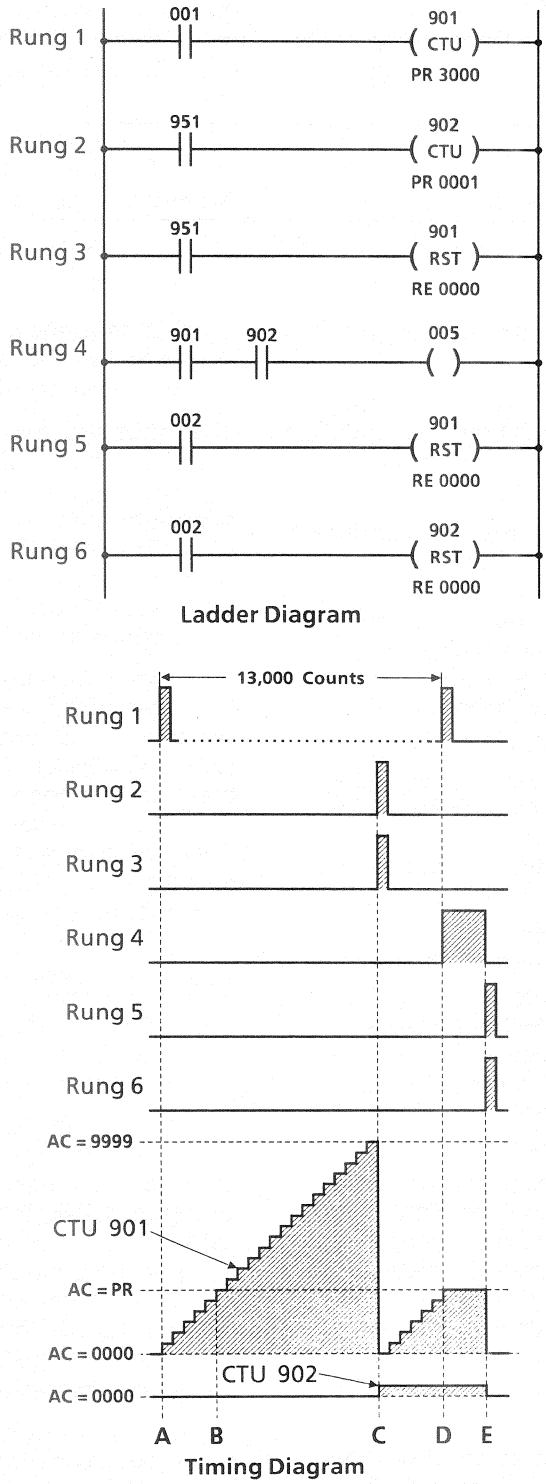


Figure 3.30 Cascading counters.

STOP

Re-read paragraphs 3.6–3.6.3. Then answer the questions in Q/E Unit 8 of the Study Guide.

3.7 SEQUENCER INSTRUCTIONS

Sequencer instructions include the Sequencer Output –(SQO)– and the Sequencer Input –(SQI)–. Both are retentive, requiring the use of the Reset instruction.

Sequencer instructions are typically used with machines or processes involving repeating operating cycles which can be segmented into *steps*.

Figure 3.31 indicates some basic sequencer characteristics you should become familiar with.

The basic operation of sequencer instructions is shown in Figure 3.32. The examples involve simple, 4-step sequencers using 5 external outputs or inputs.

NOTE: Understanding and using sequencer instructions will be easier if you first become familiar with timer and counter instructions (Pages 3-13 to 3-21).

SQO and SQI Sequencer Instructions:

Address: 901-932 (internal).

Sequencers can have up to 255 steps and can be *time-driven* or *event-driven*. Operation involves an Accumulated value (AC) and a Preset value (PR).

Time-driven sequencers count 0.1-second intervals while the sequencer rung is TRUE. When the AC value reaches the PR value, the sequencer advances to the next step and the AC value increments from 0000 again.

Event-driven sequencers count FALSE-TRUE transitions of the sequencer rung. When the AC value reaches the PR value, the sequencer advances to the next step and the AC value increments from 0000 again.

The Sequencer Output (SQO) instruction sets the ON/OFF status of up to 8 external outputs (2 output modules) for each step. After the final step, the sequencer continues with step 1.

The Sequencer Input (SQI) instruction examines the ON/OFF status of up to 8 external inputs (2 input modules) for each step, setting an *input-satisfied status bit* ON when the status of the external inputs matches programmed data. After the final step, the sequencer continues with step 1.

The basic operation of sequencers is shown in Figure 3.32.

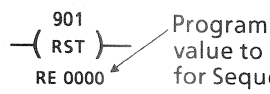
Input-Satisfied Status Bit and Completion Bit:

The SQI sequencer *input-satisfied status bit* has the same address as the SQI sequencer instruction.

The input-satisfied status bit is set ON when the ON/OFF status of the external inputs matches the programmed data for the current step. Examine ON instructions at the input-satisfied status bit address go TRUE; Examine OFF instructions go FALSE.

The *completion bit* of SQO and SQI sequencers is assigned the sequencer address plus 50. (901 + 50 = 951.) The completion bit is set ON when the sequencer completes its final step. Examine ON instructions at the completion bit address go TRUE; Examine OFF instructions go FALSE.

Reset Instruction:



The reset (RST) instruction is given the same address as the sequencer instruction. When the RST instruction goes TRUE, the SQI input-satisfied status bit and the completion bit are reset to OFF, and the sequencer instruction is reset to step 1. The RST instruction must go FALSE again before sequencer operation can resume.

Note: The operator terminal display does not show the RST instruction in reverse video when it is TRUE.

Sequencer Data:

In programming the ON/OFF status of inputs and outputs for sequencer steps, 1's and 0's are used, where 1 = ON and 0 = OFF. To simplify programming, a 2-digit code is used to represent the status of the 8 inputs or outputs for a particular step.

I/O module group numbers define which 8 external addresses are involved in the sequencer data. Masking data (using the 2-digit code) indicates which of these addresses apply to the sequencer and which do not.

Additional details on sequencer data appear in Paragraph 3.7.1.

Power Down:

The sequencer status is retentive. When you apply power after a power-down, the current step number, AC value, and the completion bit status will be the same as before power was removed. The status of the input-satisfied bit will also be the same, provided the status of the external inputs remain the same.

The sequencer status is also retained when going from the Run mode to another mode.

Monitoring Data:

The PR value, AC value, and current step number can be monitored in the Run and Test modes. Unlike timers and counters, the PR and AC values of sequencers cannot be changed in the Run or Test modes.

Figure 3.31 Characteristics of sequencer instructions.

Sequencer Output (SQO)

The SQO instruction can control the ON/OFF status of up to 8 external outputs for up to 255 steps. It can be either time-driven or event-driven. With time-driven sequencers, each step functions similar to timer instructions, in that it involves an AC value and a programmed PR value (*dwell time*). In the same way, the event-driven sequencer functions similar to the counter instruction.

Time-Driven SQO:

Step	External Outputs (Data Entry: ON = 1, OFF = 0)					Dwell Time (PR Value)
	A	B	C	D	E	
1	OFF	OFF	OFF	OFF	OFF	5 seconds
2	OFF	ON	OFF	ON	OFF	20 seconds
3	ON	ON	ON	ON	OFF	60 seconds
4	ON	OFF	OFF	ON	ON	10 seconds

Beginning with the sequencer reset: When the SQO instruction goes TRUE, step 1 is initiated; outputs A thru E are OFF. After a dwell time of 5 seconds (assuming SQO remains TRUE), step 2 begins; outputs B and D go ON. After 20 seconds, step 3 begins; outputs A and C go ON. After 60 seconds, step 4 begins; outputs B and C go OFF, output E goes ON. After 10 seconds, a completion bit is set ON and the cycle repeats with step 1.

Event-Driven SQO:

Step	External Outputs (Data Entry: ON = 1, OFF = 0)					FALSE-TRUE Transitions (PR Value)
	A	B	C	D	E	
1	OFF	OFF	OFF	OFF	OFF	1
2	OFF	ON	OFF	ON	OFF	1
3	ON	ON	ON	ON	OFF	1
4	ON	OFF	OFF	ON	ON	1

The PR value is set at 1 for each step (the typical case). Beginning with step 1, outputs A thru E are OFF. After a FALSE-TRUE transition of the SQO instruction occurs, step 2 is in effect; outputs B and D go ON. After a 2nd transition, step 3 is in effect; outputs A and C go ON. After a 3rd transition, step 4 is in effect; outputs B and C go OFF, output E goes ON. After a 4th transition, a completion bit is set ON and the cycle repeats with step 1.

Sequencer Input (SQI)

The SQI instruction differs from the SQO instruction in that it examines up to 8 external *inputs* and sets an input-satisfied status bit ON when the ON/OFF status of these inputs matches the programmed data for a particular step. The SQI sequencer can also be either time-driven or event-driven.

Time-Driven SQI:

Step	External Inputs (Data Entry: ON = 1, OFF = 0)					Dwell Time (PR Value)	Input-satisfied status bit
	A	B	C	D	E		
1	OFF	OFF	OFF	OFF	OFF	120 seconds	Each step: This bit is ON only if inputs match programmed data.
2	OFF	ON	OFF	ON	OFF	60 seconds	
3	ON	ON	ON	ON	OFF	60 seconds	
4	ON	OFF	OFF	ON	ON	120 seconds	

This sequencer moves from step to step in the same way as the time-driven SQO sequencer.

During the time that a particular step of the SQI instruction is in effect, the input-satisfied status bit will be set ON only when the status of external inputs A thru E matches the programmed input data for that step.

Event-Driven SQI:

Step	External Inputs (Data Entry: ON = 1, OFF = 0)					FALSE-TRUE Transitions (PR Value)	Input-satisfied status bit
	A	B	C	D	E		
1	OFF	OFF	OFF	OFF	OFF	1	Each step: This bit is ON only if inputs match programmed data.
2	OFF	ON	OFF	ON	OFF	1	
3	ON	ON	ON	ON	OFF	1	
4	ON	OFF	OFF	ON	ON	1	

This sequencer moves from step to step in the same way as the event-driven SQO sequencer.

During the time that a particular step of the SQI instruction is in effect, the input-satisfied status bit will be set ON only when the status of external inputs A thru E matches the programmed input data for that step.

Figure 3.32 Basic operation of sequencer instructions.

3.7.1 Sequencer Data – Sequencer data includes:

- Sequencer classification.
- I/O module group numbers and terminal addresses.
- Mask data.
- I/O status data for each step.
- Program entry codes.
- Preset values for each step.

We recommend that you use the Sequencer Instruction Data Form to document this data. This form (included in the Appendix) allows you to document sequencer data in an orderly, systematic way, reducing the chances for programming errors.

A sample data form is shown on the facing page. To illustrate its use, we'll enter data for the time-driven SQO sequencer shown in Figure 3.33 (same sequencer as discussed in Figure 3.32).

The following steps are indicated for entering data.

Step 1: a) Enter the sequencer classification and address. We've indicated SQO, address 901, and time-driven in the heading of the data form.

b) Enter the I/O module group numbers and corresponding terminal addresses. Identification of group numbers is shown in Figure 3.34. We've selected group numbers 7 and 8, which have addresses 025 to 028 and 029 to 032.

You have the option of assigning any of the I/O module group numbers to either the first or second sequencer data group. Under each I/O module group number you must list the corresponding four terminal addresses in *reverse* order, beginning with the highest number at the left.

Step 2: a) Enter mask data. Enter a 1 in the form under those addresses to be controlled by the sequencer. Enter a 0 under those to be excluded.

Our sequencer has 5 outputs; we've chosen to exclude addresses 030, 031, and 032 from sequencer control. These outputs can be used elsewhere in your program.

b) Enter I/O status data for each step. 1 corresponds to an ON condition and 0 corresponds to an OFF condition. Thus, for each step, the ON and OFF states of outputs A, B, C, D, and E (Figure 3.33) correspond to the 1's and 0's entered under addresses 029, 028, 027, 026, and 025 in the data form.

Step 3: a) Enter program entry codes for mask and I/O status data. The program code is shown in Figure 3.35. Applying this code to mask data 0001 under the second group, we've entered 1 under "2nd Grp" in the form. Similarly, mask data 1111 under the first group is identified by code F.

Note that for the I/O status data, the masked addresses are considered as 0's. Thus, for step 4 under the second group, the status data is considered to be 0001. Code 1 is entered.

b) Enter Preset values. The Preset values for time-driven sequencers represent 0.1-second intervals, just as with timers. Thus, 0050 indicates 5 seconds, 0200 indicates 20 seconds, and so on.

For event-driven sequencers, the Preset values directly represent the number of FALSE-TRUE transitions of the sequencer instruction required to move to the next step.

Time-Driven SQO:

Step	External Outputs (Data Entry: ON = 1, OFF = 0)					Dwell Time (PR Value)
	A	B	C	D	E	
1	OFF	OFF	OFF	OFF	OFF	5 seconds
2	OFF	ON	OFF	ON	OFF	20 seconds
3	ON	ON	ON	ON	OFF	60 seconds
4	ON	OFF	OFF	ON	ON	10 seconds

Figure 3.33 Data for this SQO time-driven sequencer is entered in the sequencer data form appearing on the next page.

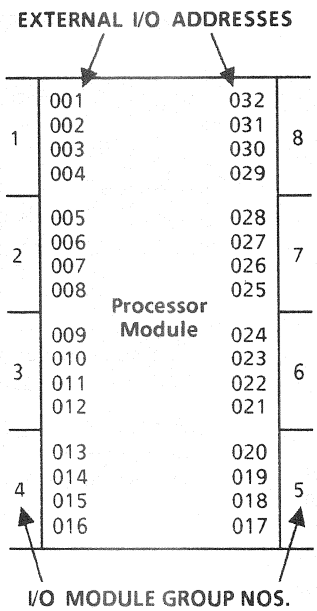


Figure 3.34 I/O module group numbers 1 thru 8 relate to external I/O address numbers as shown above.

Sequencer Data	Program Code
0000 0001 0010 0011	0 1 2 3
0100 0101 0110 0111	4 5 6 7
1000 1001 1010 1011	8 9 A B
1100 1101 1110 1111	C D E F

Example - For sequencer data 0001 1111, enter code F for 1st Sequencer Data Group, and code 1 for 2nd Sequencer Data Group. Thus:

0001 1111
1 F

Figure 3.35 Sequencer data coding.

Sequencer Instruction Data Form

(Copies are in the Appendix)

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☐ ADDRESS: ____ TIME DRIVEN ☐ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST						
MODULE GROUP NUMBER →									2nd Grp	1st Grp	
I/O TERMINAL ADDRESS →											
MASK DATA →											
STEP NO. 1											
2											
3											
4											

Step 1: a) Enter sequencer classification and address. b) Enter I/O module group numbers, terminal addresses.

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☒ ADDRESS: 901 TIME DRIVEN ☒ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST						
MODULE GROUP NUMBER →	8				7				2nd Grp	1st Grp	
I/O TERMINAL ADDRESS →	032	031	030	029	028	027	026	025			
MASK DATA →											
STEP NO. 1											
2											
3											
4											

Step 2: a) Enter mask data. b) Enter I/O status data for each step.

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☒ ADDRESS: 901 TIME DRIVEN ☒ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST						
MODULE GROUP NUMBER →	8				7				2nd Grp	1st Grp	
I/O TERMINAL ADDRESS →	032	031	030	029	028	027	026	025			
MASK DATA →	0	0	0	1	1	1	1	1			
STEP NO. 1				0	0	0	0	0			
2				0	1	0	1	0			
3				1	1	1	1	0			
4				1	0	0	1	1			

Step 3: a) Enter program entry codes for mask and I/O status data. b) Enter Preset values.

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☒ ADDRESS: 901 TIME DRIVEN ☒ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST						
MODULE GROUP NUMBER →	8				7				2nd Grp	1st Grp	
I/O TERMINAL ADDRESS →	032	031	030	029	028	027	026	025			
MASK DATA →	0	0	0	1	1	1	1	1	1	F	
STEP NO. 1	⊗	⊗	⊗	0	0	0	0	0	0	0	0 0 5 0
2	⊗	⊗	⊗	0	1	0	1	0	0	A	0 2 0 0
3	⊗	⊗	⊗	1	1	1	1	0	1	E	0 6 0 0
4	⊗	⊗	⊗	1	0	0	1	1	1	3	0 1 0 0

⊗ Masked addresses are 0 for coding.

3.7.2 Sequencer Output (SQO) – Figure 3.36 shows a ladder diagram and a completed data form for an SQO instruction. This is a 10-step, time-driven sequencer controlling 7 external outputs.

Operation begins when rung 1 goes TRUE. As long as the rung remains TRUE, the sequencer advances from one step to the next each time the Preset value of the current step is reached. At the beginning of each step, external outputs at addresses 014 thru 020 are set ON or OFF as indicated by the 1's and 0's in the data form.

Rungs 2 and 3 are included to demonstrate how the completion bit functions. The first time the sequencer recycles from the final (10th) step to step 1, the completion bit is set ON, causing the Examine ON instruction in rung 2 to go TRUE, and the Examine OFF instruction in rung 3 to go FALSE. The completion bit remains ON until the sequencer is reset.

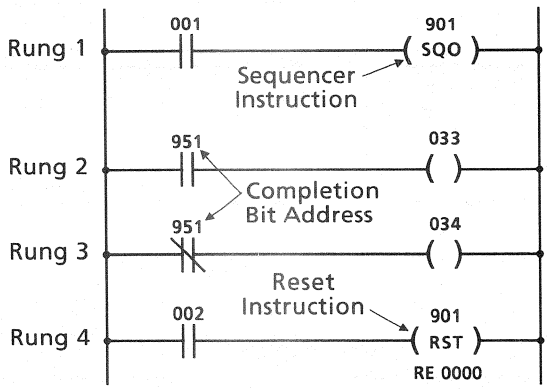
Rung 4 contains the Reset instruction. When this rung goes TRUE, the sequencer is reset to step 1 and the completion bit is reset to OFF.

While the RST instruction is TRUE, the sequencer is disabled. Operation can resume when rung 4 goes FALSE again.

3.7.3 Sequencer Input (SQI) – Figure 3.37 shows a ladder diagram and a completed data form for an SQI instruction. This 10-step, event-driven sequencer examines 7 external inputs.

FALSE-TRUE transitions of rung 1 move the sequencer from one step to the next. In this example, a single FALSE-TRUE transition advances the sequencer, since all Preset values are set at 1.

The sequencer examines external inputs at addresses 002 thru 004 and 009 thru 012. The ON/OFF states of these inputs determine whether the input-satisfied status bit will be ON or OFF. That is, when the ON/OFF states of the external inputs match the data programmed for the current step,



Ladder Diagram

SEQUENCER CLASSIFICATION: –(SQO)– ☒ **ADDRESS: 901** **TIME DRIVEN** ☒

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET			
SEQUENCER DATA GROUP →	SECOND				FIRST									
MODULE GROUP NUMBER →	5				4				2nd Grp	1st Grp				
I/O TERMINAL ADDRESS →	020	019	018	017	016	015	014	013						
MASK DATA →	1	1	1	1	1	1	1	0	F	E				
STEP NO. 1	1	0	0	1	0	1	1	⊗	9	6	0	0	5	0
2	0	0	0	1	1	0	0	⊗	1	8	0	0	5	0
3	1	1	1	0	0	0	0	⊗	E	0	0	0	2	5
4	0	1	0	1	0	1	0	⊗	5	4	0	0	1	0
5	0	0	0	0	1	1	1	⊗	0	E	0	0	1	0
6	0	1	0	1	1	1	0	⊗	5	C	0	0	1	0
7	1	1	1	0	0	0	0	⊗	E	0	0	0	0	5
8	1	0	0	0	1	1	1	⊗	8	E	0	0	0	5
9	1	1	1	1	1	0	0	⊗	F	8	0	0	1	5
10	1	1	1	1	1	1	1	⊗	F	E	0	0	3	0

⊗ Masked address. Used as 0 for coding purposes.

Figure 3.36 A time-driven SQO instruction.

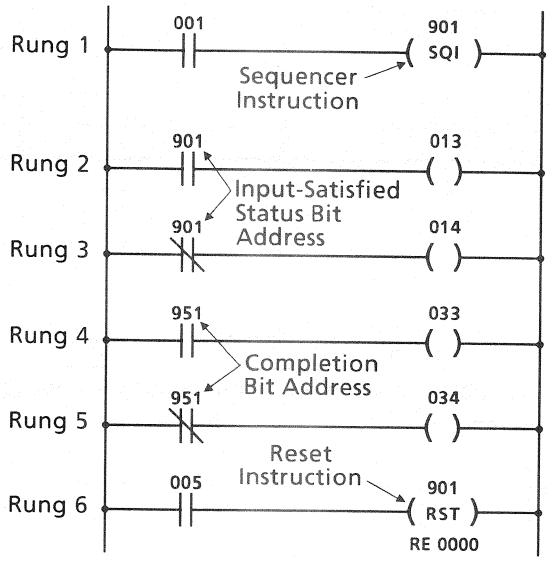
the input-satisfied status bit will be ON; when the external inputs do not match the data programmed for the current step, the status bit will be OFF.

Rungs 2 and 3 are output rungs, containing examine instructions at the input-satisfied status bit address. When the status bit is ON, rung 2 is TRUE and rung 3 is FALSE; when the status bit is OFF, rung 2 is FALSE and rung 3 is TRUE.

Rungs 4 and 5 show how the completion bit functions. The first time the sequencer recycles from the final (10th) step to step 1, the completion bit is set ON, causing the Examine ON instruction in rung 4 to go TRUE and the Examine OFF instruction in rung 5 to go FALSE. The completion bit remains ON until the sequencer is reset.

Rung 6 contains the Reset instruction. When this rung goes TRUE, the sequencer is reset to step 1 and the status and completion bits are reset to OFF.

While the RST instruction is TRUE, the sequencer is disabled. Operation can resume when rung 6 goes FALSE again.



Ladder Diagram

SEQUENCER CLASSIFICATION: –(SQI)– ☒ **ADDRESS: 901** **EVENT DRIVEN** ☒

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET			
SEQUENCER DATA GROUP →	SECOND				FIRST									
MODULE GROUP NUMBER →	3				1				2nd Grp	1st Grp				
I/O TERMINAL ADDRESS →	012	011	010	009	004	003	002	001						
MASK DATA →	1	1	1	1	1	1	1	0	F	E				
STEP NO. 1	1	1	0	0	1	1	1	⊗	C	E	0	0	0	1
2	0	0	0	1	1	0	0	⊗	1	8	0	0	0	1
3	0	0	1	1	1	1	1	⊗	3	E	0	0	0	1
4	1	1	0	0	1	1	1	⊗	C	E	0	0	0	1
5	1	1	0	0	0	0	0	⊗	C	0	0	0	0	1
6	1	0	0	0	0	0	0	⊗	8	0	0	0	0	1
7	0	1	0	0	0	1	0	⊗	4	4	0	0	0	1
8	0	0	1	0	0	0	0	⊗	2	0	0	0	0	1
9	1	0	1	0	1	0	1	⊗	A	A	0	0	0	1
10	0	1	0	0	1	1	1	⊗	4	E	0	0	0	1

⊗ Masked address. Used as 0 for coding purposes.

Figure 3.37 An event-driven SQI instruction.

3.7.4 Programming Sequencer Instructions – The keystroke example of Figure 3.38 shows you how to program sequencer instructions. The data form is a reproduction of the the first two steps of the SQO sequencer form shown on Page 3-26. The keystroke example indicates how to enter:

1. Sequencer instruction, address.
2. Time- or event-driven operation.
3. I/O module group numbers.
4. Mask data.
5. I/O status data (step 1).
6. Preset value (step 1).

Keystrokes for entering other instructions associated with sequencers are not shown.

The RE value of the RST instruction used with sequencers is normally set to zero.

SEQUENCER CLASSIFICATION: -(SQO)- ☒ ADDRESS: 901 TIME DRIVEN ☒

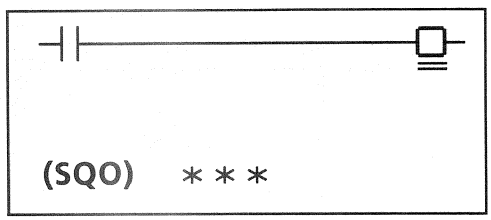
SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET			
SEQUENCER DATA GROUP →	SECOND				FIRST									
MODULE GROUP NUMBER →	5				4				2nd Grp	1st Grp				
I/O TERMINAL ADDRESS →	020	019	018	017	016	015	014	013						
MASK DATA →	1	1	1	1	1	1	1	0	F	E				
STEP NO. 1	1	0	0	1	0	1	1	⊗	9	6	0	0	5	0
2	0	0	0	1	1	0	0	⊗	1	8	0	0	5	0

Keystroke Example – SQO Instruction

(The SQI instruction is entered in this same way)

Key Sequence **Display**

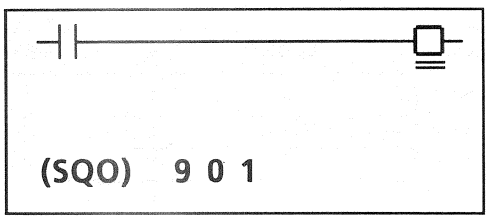
-(SQO)-



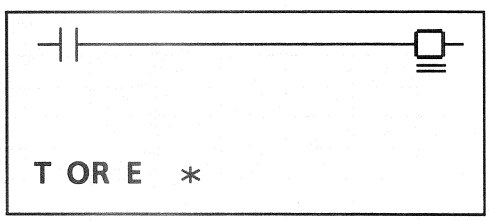
When the SQO key is pressed, the rung is completed with the output instruction symbol. Stars prompt for address.

-(SQO)- 9

0 1



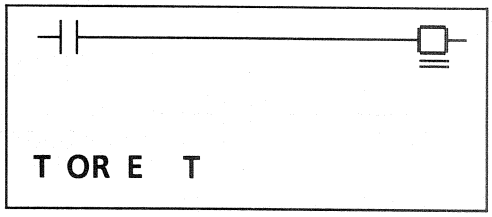
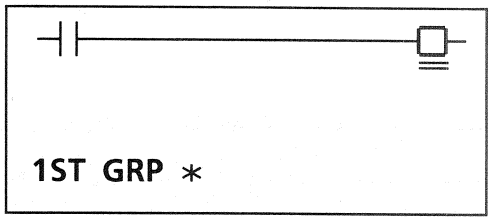
ENTER



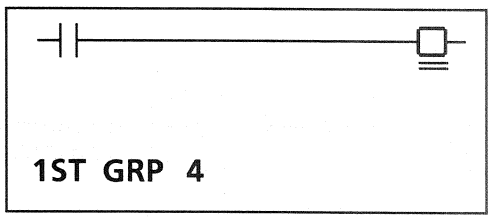
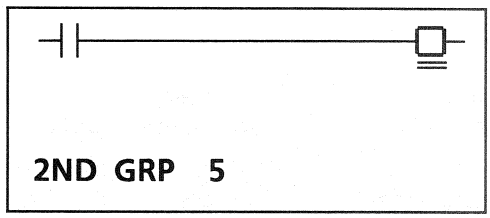
Display is prompting for Time- or Event-driven sequencer.

Continued on next page

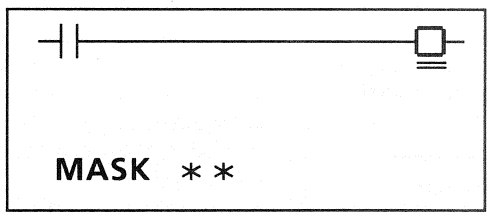
Figure 3.38.

Key Sequence	Display
SHIFT, TIME	
ENTER	

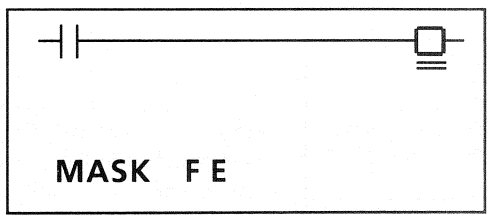
Display is prompting for first I/O module group number.

4	
ENTER, 5	

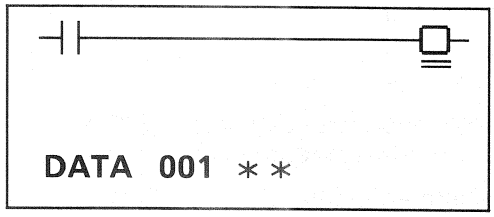
First and second I/O module group numbers are entered.

ENTER	

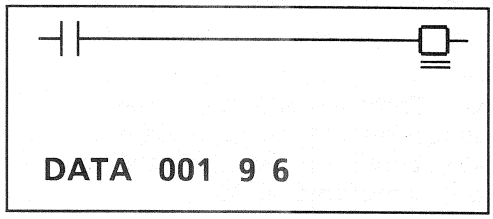
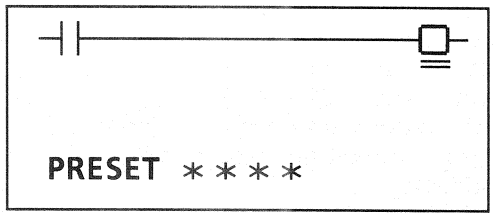
Display is prompting for mask data.

SHIFT, F SHIFT, E	

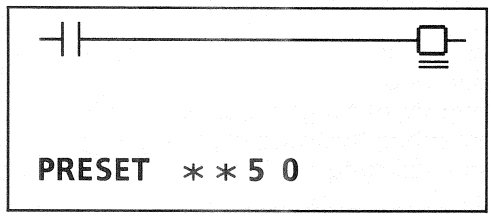
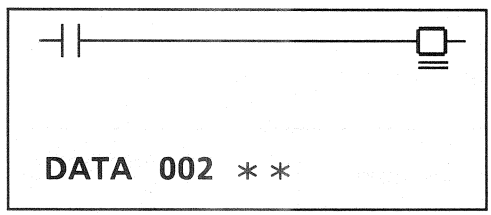
E corresponds to 1st group. F corresponds to 2nd group.

Key Sequence	Display
ENTER	

Display is prompting for step 1 data.

9, 6	
ENTER	

Step 1 data is entered.
Display is prompting for step 1 Preset value.

5, 0	
ENTER	

Display is prompting for step 2 data.

Display will continue to prompt for data and Preset values for the succeeding step. When the final step has been entered (step 10 in this example), press the ENTER key and the End of program will be displayed. Programming of the sequencer rung is then complete.

Figure 3.38.

Re-read paragraphs 3.7–3.7.4. Then answer the questions in Q/E Unit 9 of the Study Guide.

3.8 SPECIAL INSTRUCTIONS

The following are special relay-type instructions.

- Master Control Reset –(MCR)– No address required.
- Zone Control Last State –(ZCL)– No address required.
- Immediate Input –] I [– External I/O Addresses (001-032).
- Immediate Output –(IOT)– External I/O Addresses (001-032).

3.8.1 Master Control Reset and Zone Control Last State – These instructions allow you to use one set of condition instructions to control multiple outputs.

Figure 3.39a illustrates MCR and ZCL *zones*. Note that a zone is defined by a start rung and an end rung. The MCR and ZCL instructions are outputs in these rungs, and do not require an address.

The start rung contains the condition instructions which enable and disable the zone. The end rung contains the MCR or ZCL instruction only – it must not have any condition instructions. Any number of rungs can be programmed between the start rung and end rung.

When the start rung is TRUE, output instructions in the zone function normally. When the start rung is FALSE, outputs within the zone are overridden and controlled by the MCR or ZCL instruction.

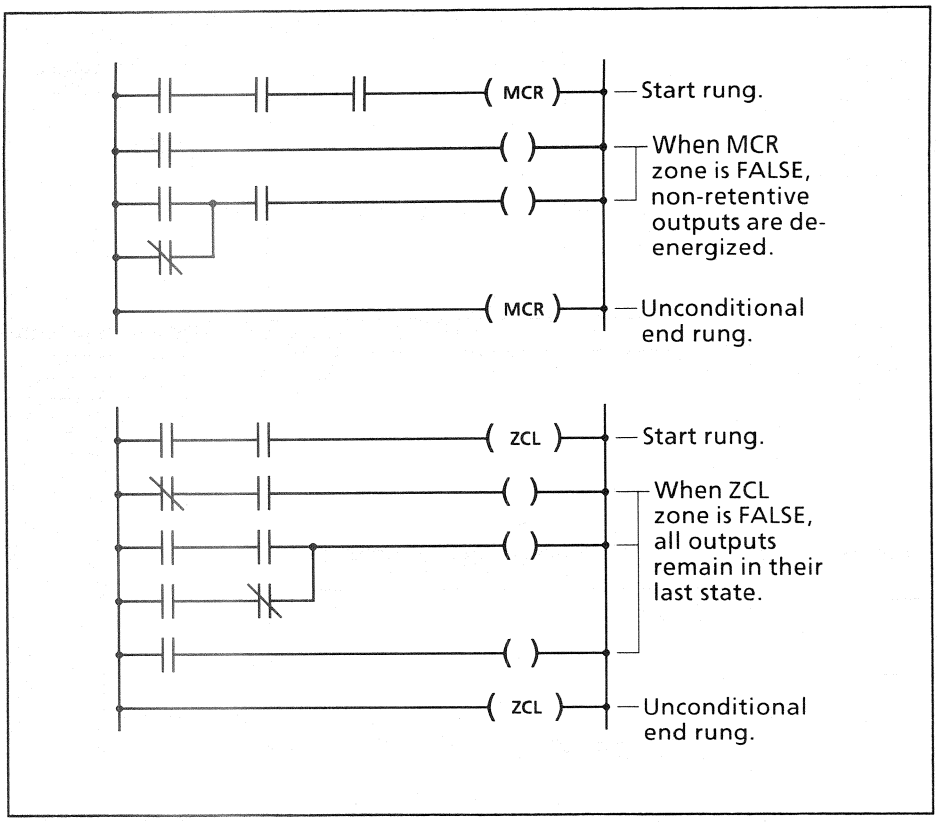


Figure 3.39a MCR and ZCL instructions.

MCR and ZCL instructions affect various instructions as shown in Figure 3.39b.

Each MCR or ZCL zone must be complete and distinct. Do not program one MCR or ZCL zone within another, since it may produce undesired results.

NOTE: The operator terminal does not display MCR or ZCL instructions in reverse video when they are TRUE.

MCR/ZCL instructions vs a master control relay: In some respects, the function of MCR and ZCL instructions is similar to a hard-wired master control relay of a relay system, which can be wired to control the overall function of several operations, based on certain necessary conditions. An important distinction must be made, however:

The master control relay is a safety device, providing Emergency Stop capability to the controller operator; with MCR and ZCL instructions, this is **not** the case.

We strongly recommend that you install a hard-wired master control relay to provide your controller with emergency I/O power shut-down. See Section 4 for wiring details.

Instruction	MCR Zone Enabled (start rung FALSE)	ZCL Zone Enabled (start rung FALSE)
Non-Retentive Outputs	De-energized	Remain in last state
Latch/Unlatch	Remain in last state	Remain in last state
Timers/Counters	AC value stops incrementing, value retained. Status, overflow, and underflow bits reset to OFF.	AC value stops incrementing, value retained. Status, overflow, and underflow bits remain in last state.
Sequencers	AC value stops incrementing, value retained. Step number retained. Completion and input-satisfied status bits remain in last state.	AC value stops incrementing, value retained. Step number retained. Completion and input-satisfied status bits remain in last state.
Reset	Remains in last state. Instructions cannot be reset.	Remains in last state. Instructions cannot be reset.

Figure 3.39b.

3.8.2 Immediate Input –] I [– Instruction – The Immediate Input is a special version of the Examine ON instruction, and is programmed in the same way. This instruction is used with critical input devices to interrupt the program scan and update the status of the input device in advance of the upcoming I/O scan.

Updating of inputs is divided into *adjacent I/O sets* as follows:

External addresses 001-008, I/O module groups 1 and 2
External addresses 009-016, I/O module groups 3 and 4
External addresses 017-024, I/O module groups 5 and 6
External addresses 025-032, I/O module groups 7 and 8

When the immediate input instruction address is updated, all input addresses within the same set are also updated. If one of the modules in the set is an output module, its addresses will not be updated.

Operation is illustrated in Figure 3.40. When the program scan reaches the immediate input instruction, the scan is interrupted and the addressed input device and adjacent inputs are updated. This update occurs whether the addressed input is TRUE or FALSE.

After the update, the program scan is resumed with this updated input status.

You can use the immediate input instruction in your program as many times as needed. However, use it only when necessary. Proper use depends on the response time of the specific input device, and the location of the instruction in the user program.

The immediate input is most useful if the instruction associated with the critical input device is at the middle or toward the end of the program. The immediate input isnot needed near the beginning of the program, since the I/O scan has just occurred at that time.

3.8.3 Immediate Output –(IOT)– Instruction – The Immediate Output is a special version of the Output Energize instruction, and is programmed in the same way. The instruction is used with critical output devices, to interrupt the program scan and update the output device status (energize or de-energize it) in advance of the upcoming I/O scan.

The immediate output instruction is executed regardless of whether rung conditions are TRUE or FALSE.

The immediate input and immediate output instructions function in the same manner, involving *adjacent I/O sets*. For details on the immediate output, read paragraph 3.8.2 again, interchanging the roles of inputs and outputs.

STOP **Re-read paragraphs 3.8 –3.8.3. Then answer the questions in Q/E Unit 10 of the Study Guide.**

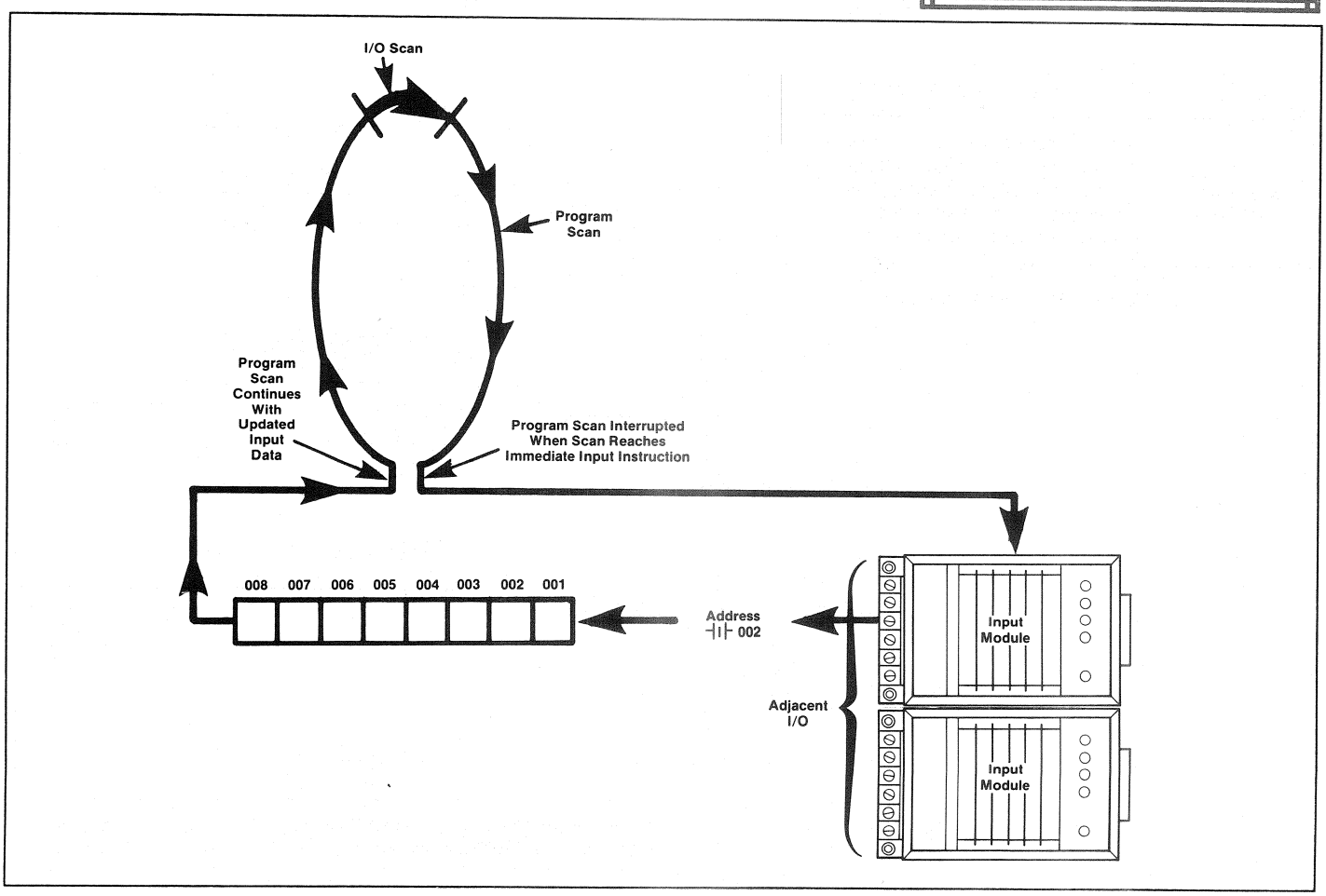


Figure 3.40 Operation of the immediate input instruction.

3.9 PROGRAM EDITING

Your program can be easily edited with the Cursor, Search, Remove/Insert, and Clear Memory functions. Refer to Figure 3.41.

3.9.1 Cursor Control – A blinking instruction in a ladder rung of the operator terminal display indicates the cursor position. The cursored instruction will appear in the lower portion of the display along with its address.

The cursor position can be controlled by the ←, →, ↑, and ↓ keys. These keys function in the Run, Test, and Program modes.

The ← key will move the cursor position from right to left within a rung. The → key will move the cursor position from left to right within a rung.

When the cursor is positioned on a timer, counter or sequencer instruction, pressing the → key will display data associated with that instruction (PRESET, ACCUM, MASK). When accessing data associated with these instructions, pressing the ← key will return the display to the output instruction symbol.

The keystroke example of Figure 3.42 illustrates use of the → key to access data associated with the RTO instruction. Similar keystrokes are used for accessing data

Keys	Function
← ↑ → ↓	**Cursor Control** – Allows you to move thru your program from instruction to instruction or from rung to rung; gives you access to timer, counter, and sequencer data.
SEARCH	**Search** – Use this key to locate instructions and other information in your program.
REMOVE / INSERT	**Editing** – Allows you to insert or remove individual instructions or entire rungs.
MODE 20 KEYSTROKE SEQUENCE	**Clear Memory** – This mode allows you to clear your entire program from the processor memory. The EEPROM memory is not affected.

Figure 3.41 Editing functions.

Keystroke Example – Accessing Timer Instruction Data with Cursor Key

(Counter and sequencer data is accessed this same way)

Key Sequence	Display
Move cursor to the RTO or RTF instruction you would like to monitor.	(RTO) 9 1 0
→	PRESET 0 5 0 0
→	ACCUM 0 0 2 5
→	PRESET 0 5 0 0
←	(RTO) 9 1 0

Note that the cursor-right key is used to access data and the cursor-left key returns you to the timer instruction.

Figure 3.42.

associated with counter and sequencer instructions.

The ↑ key moves the cursor position to the first instruction in the preceding rung. The ↓ key moves the cursor to the first instruction in the following rung.

3.9.2 Search Function – The Search function can be used in the Run, Test, and Program modes. It is used to locate:

1. Any addressed instruction.
2. The START of a program.
3. The END of a program.
4. The number of a cursored rung.

When searching for a specific instruction, you can begin the search at the program START or at any other point in the program. The search will begin at the cursor location and continue to the END of the program.

The keystroke example of Figure 3.43 illustrates the use of the Search key.

Keystroke Example – Search Function

1. To locate a specific instruction and address:

Key Sequence	Display
	−] [− 0 0 6

Search can be initiated from any point in your program.

Key Sequence	Display
SHIFT, SEARCH	−] [− 0 0 1 SRCH

When the Search key is pressed, the display will show the instruction last searched. If you wish to find this instruction, simply press Enter.

If there is no instruction in the search memory, the display will prompt for a new instruction with * * * * *. If you wish to find a new instruction, enter the new instruction and its address.

Key Sequence	Display
⊣⊢, 2, 5	−] [− ∗2 5 SRCH
ENTER	−] [− 0 2 5

The controller will search for the instruction between the cursor location and the End of the program. To continue the search, press SHIFT, SEARCH, ENTER

Key Sequence	Display
ENTER	NOT FOUND

You are at the End of the program. NOT FOUND tells you that the instruction is not located in the part of the program you searched.

Key Sequence	Display
CANCEL CMD	END XXXX WORDS REMAINING

To search the entire program for the instruction, you must first cursor to the Start of the program.

2. To locate the Start of the program:

Press SHIFT, SEARCH, ↑, ENTER

3. To locate the End of the program:

Press SHIFT, SEARCH, ↓, ENTER

4. To locate the number of a cursored rung:

Cursor to any instruction in the rung. Then press SHIFT, SEARCH, ←, ENTER

The rung and the rung number will be displayed. The cursor remains on the selected instruction.

Figure 3.43.

3.9.3 Remove/Insert Function – In the Program mode, you can add and delete Examine instructions and entire rungs in your program with the Remove/Insert key. The keystroke examples of Figures 3.44a and 3.44b illustrate its use.

After you insert or remove an Examine instruction, the change will not be transferred into the processor RAM memory until you: 1) enter the Run, Test, or Prom Load mode, or 2) press the cursor ↑ or ↓ key. If you fail to do one of these, and power is removed, the program change will be lost.

CAUTION: When you add or delete instructions, double-check to make sure that I/O module placement coincides with instructions and addresses of the edited program. Program operation is easily checked in the Test mode.

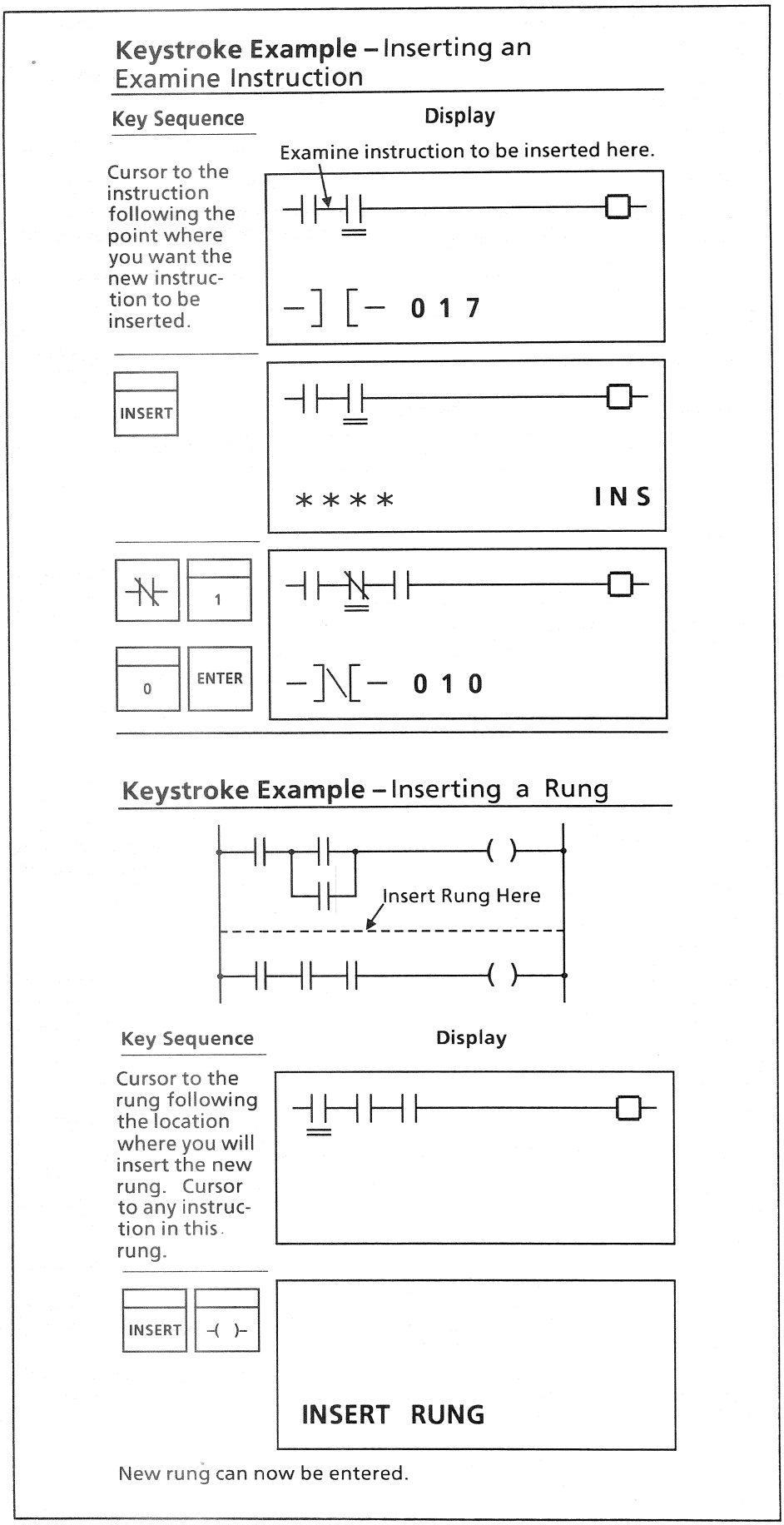
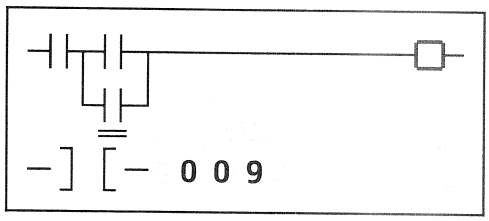
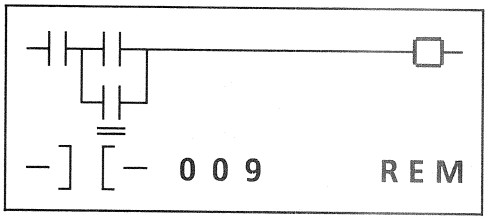
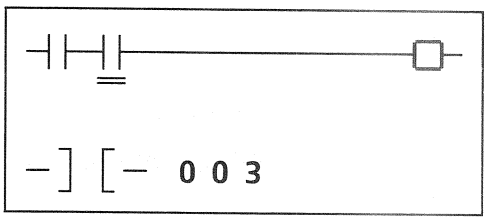


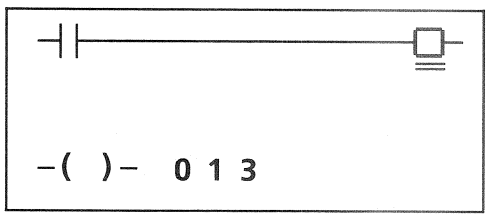
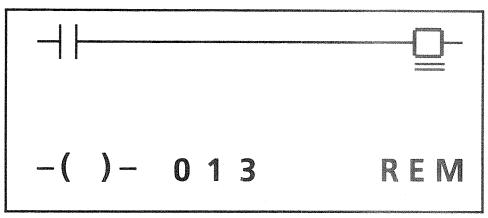
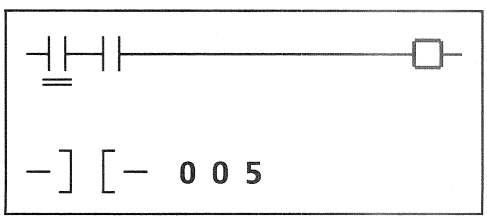
Figure 3.44a.

Keystroke Example – Removing an Examine Instruction

Key Sequence	Display
Cursor to the instruction you want to remove.	
SHIFT REMOVE	
ENTER	

The cursor has shifted to the instruction which preceded the removed instruction. The Branch Open and Branch Close instructions have been automatically removed.

Keystroke Example – Removing a Rung

Key Sequence	Display
Cursor to the output Instruction of the rung you want to remove.	
SHIFT REMOVE	
ENTER	

The rung has been removed, and the cursor has shifted to the rung which followed the removed rung.

Figure 3.44b.

3.9.4 Clear Memory – You can delete your entire program from the processor memory by using Mode 20, Clear Memory. Figure 3.45 illustrates how this is done.

Mode 20 does not affect the contents of an EEPROM memory module.

3.9.5 Editing Timer, Counter, and Sequencer Data – You can change Timer and Counter data in the Run and Test modes. This is discussed under the subject "On-Line Data Control". Refer to Page 3-40. Timer and Counter data can *not* be changed in the Program mode.

Sequencer data can be changed in the Program mode only. Figure 3.46 illustrates the keystrokes required for changing various sequencer instruction data.

The sequencer editing function does not allow steps to be added or deleted.

Keystroke Example – Clear Memory Mode

Key Sequence	Display
MODE, 2, 0	CURR MODE 01 PROGRAM NEW MODE 20
ENTER	NEW MODE 20 CLEAR RAM
ENTER	TO START CLEAR RAM PRESS ENTER
ENTER	CLEAR MEMORY COMPLETE PRESS MODE

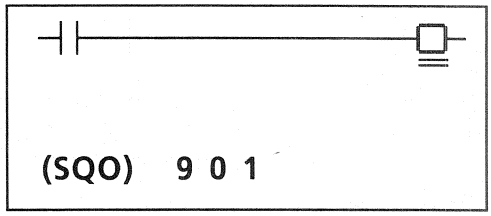
The processor RAM memory has been cleared. You can now enter a new mode.

Figure 3.45.

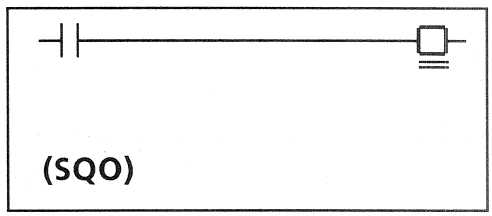
Keystroke Example – Editing Sequencer Instruction Data

Key Sequence | Display

Move cursor to the sequencer instruction you want to change.

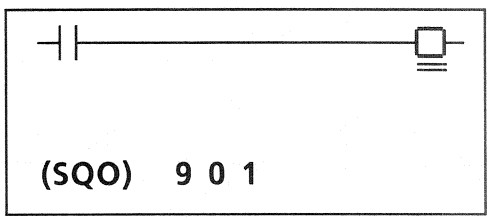


-(SQO)-



We selected an SQO instruction. If you wanted to change an SQI instruction, you would press SHIFT -(SQI)-

ENTER

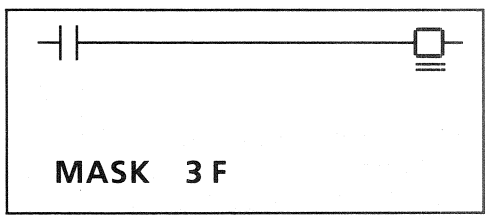


You can now change the instruction, its address, and other parameters. To locate a parameter, press the cursor-right key: →

The cursor-left key returns you to the SQO instruction and terminates the editing function.

In the following example, the mask data is changed from 3F to FF:

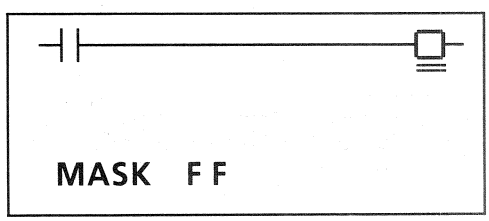
Press cursor-right key until mask data is located: →



To change mask data or step data, you must make two entries, even if you want to change only one of the characters. Changing mask data to FF:

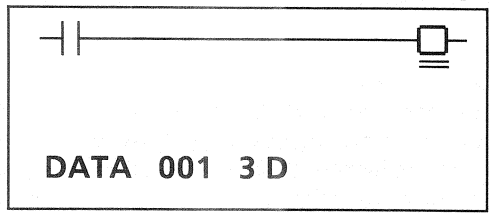
SHIFT F

SHIFT F



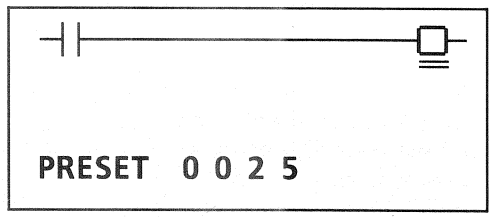
Key Sequence | Display

ENTER



The mask data change has been entered into a temporary storage register, and the display has indexed to the next parameter. If you want to change another parameter, locate it with the cursor-right key. For example, to change the Preset for step 5 from 0025 to 0100:

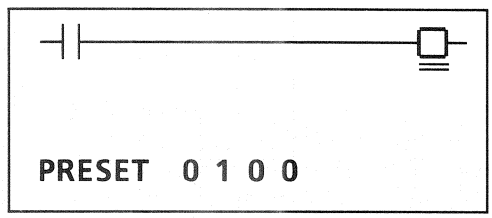
Press cursor-right key until Preset for step 5 is located: →



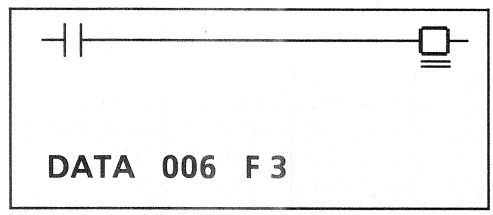
All four characters of the data must be entered to make a parameter change. Changing Preset to 0100:

0 1

0 0



ENTER



The Preset change has been entered into a temporary storage register, and the display has indexed to the next parameter. If no other parameter changes are desired, press Enter a second time:

ENTER

END
XXXX WORDS
REMAINING

All parameter changes held in temporary storage are now entered into the processor RAM memory. This ends the editing function.

Note: You cannot add or delete sequencer steps. Only the existing parameters can be changed.

Figure 3.46.

STOP

Re-read paragraphs 3.9–3.9.5. Then answer the questions in Q/E Unit 11 of the Study Guide.

3.10 ON-LINE DATA CONTROL

With the controller in the Run or Test mode, you can use the operator terminal to monitor, control, and change your program. This is referred to as *On-Line Data Control*. It includes:

- Forcing external I/O addresses.
- Monitoring timer, counter, and sequencer data.
- Changing timer and counter data. (Sequencer data can be changed in the Program mode only.)

> **WARNING:** Before you force external I/O or change data, investigate the effects on machine operation, to avoid possible personal injury and equipment damage.

3.10.1 Force Function – This function (FRC ON and FRC OFF keys) can be used in the Run and Test modes to force an external I/O address to an ON or OFF state regardless of its actual status. You will find this function very helpful in start-up and troubleshooting procedures.

Forcing an input address: When we force an input address, we are forcing the status bit of the instruction at the I/O address to an ON or OFF state. The program scan records this, and the program is executed with this forced status regardless of the actual ON/OFF status of the input device. This is illustrated in Figure 3.47.

Forcing an output address: In this case, we are forcing only the output terminal to an ON or OFF state. The status bit of the output instruction at the address is **not** affected. This is illustrated in Figure 3.48.

NOTE: When you are in the Test mode, the operator terminal display will show the output instruction at a forced-ON address in reverse video (TRUE), but the output module status indicator will remain OFF, since outputs are disabled.

Figure 3.49 shows the keystrokes used for forcing I/O addresses.

When an I/O address is forced, the operator terminal display will show the letter **F** when the instruction at that address is cursored. The FORCED I/O indicator on the processor module will also be lit.

A forced I/O address will be retained in memory when any of the following occurs: 1) A power outage; 2) The mode is changed from Run; 3) The operator terminal is disconnected.

A forced I/O address will not be retained if you enter the Program mode and make a change, or if a processor fault is detected.

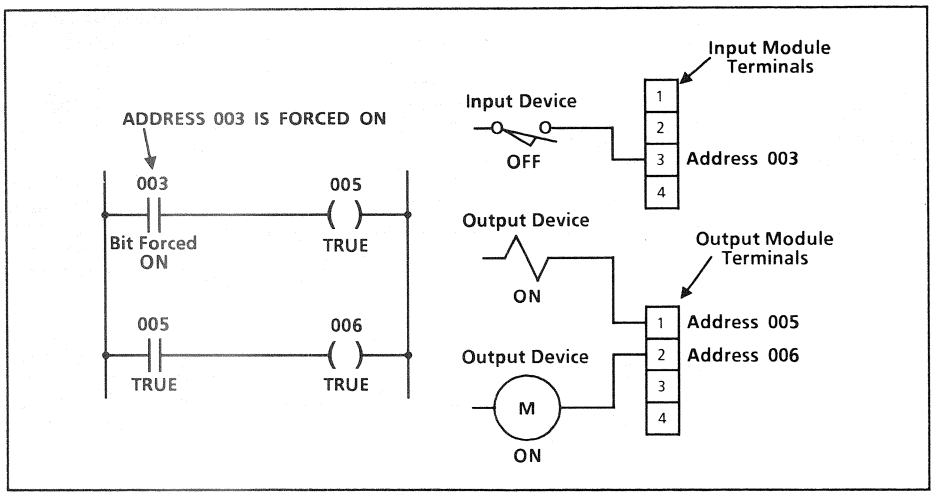


Figure 3.47 Forcing an input address. The program is executed as if the input device were actually ON.

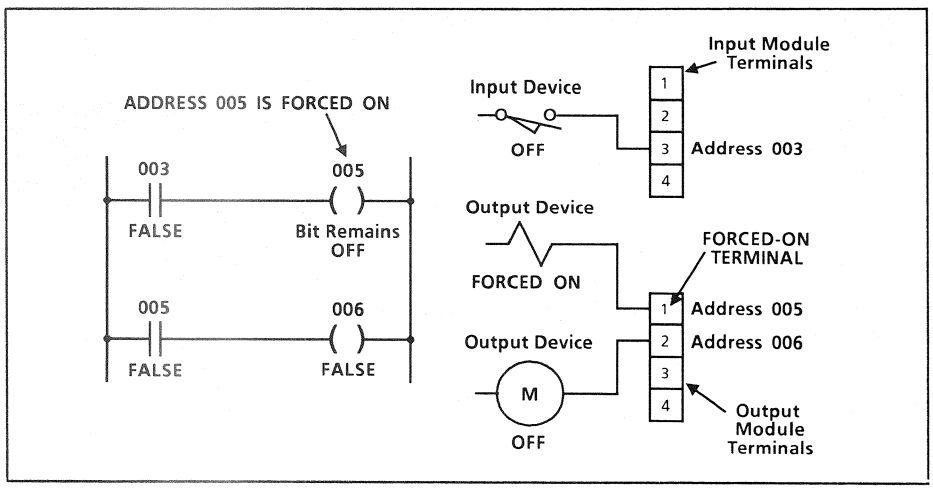
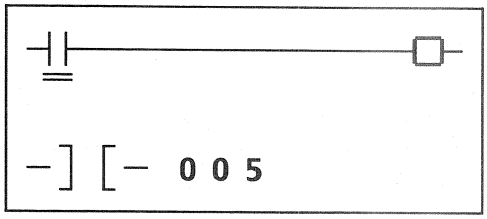
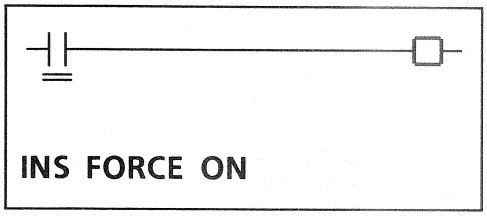
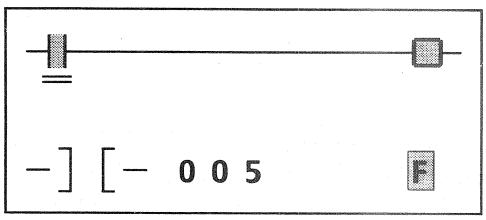


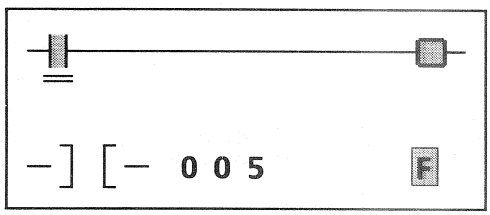
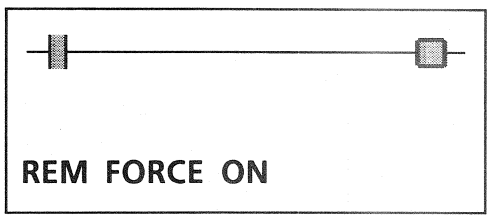
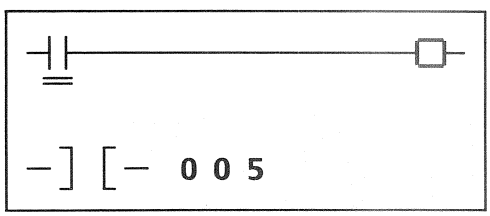
Figure 3.48 Forcing an output address. The output device associated with the forced output address is turned ON. The status bit is not affected.

Keystroke Example – Forcing I/O

Key Sequence	Display
Cursor to the instruction you want to force.	
INSERT, SHIFT FRC ON	
ENTER	

Darkened display symbols (shaded here) indicate TRUE condition. Letter **F** indicates cursored address is forced.

To Remove a Forced I/O:

Cursor to the forced instruction.	
SHIFT, REMOVE	
ENTER	

Note: To remove all forces within a program, cursor to the Start or End of the program and press SHIFT, REMOVE, ENTER

Figure 3.49.

3.10.2 Monitoring Timer, Counter, and Sequencer Data – You can monitor timer, counter and sequencer data in the Run and Test modes. While you are monitoring, the AC value will be incrementing when appropriate rung conditions exist. With sequencers, you can also monitor the current step number. Figure 3.50 shows how PR and AC values are monitored.

3.10.3 Changing Timer and Counter Data – Timer and counter PR and AC values can be changed in the Run and Test modes. This is shown in Figure 3.51.

Exceptions: If a PR value in your program is Protected (discussed in Fig. 3.26, Page 3-17), it cannot be changed. Also, you cannot change AC and PR values when you are operating from an EEPROM memory module.

Sequencer and Reset data: Sequencer data can be changed in the Program mode (Page 3-37), but *not* in the Run or Test mode. The reset instruction RE value can be changed only by deleting the instruction and re-entering it with a new RE value.

Keystroke Example – Monitoring Timer, Counter, and Sequencer Values

Key Sequence	Display
Move the cursor to the instruction you want to monitor.	(RTO) 9 1 0
→	PRESET 0 5 0 0
→	ACCUM 0 0 2 5
→	PRESET 0 5 0 0

The cursor-left key returns you to the timer instruction:

←	(RTO) 9 1 0

Figure 3.50.

Keystroke Example – On-Line Changes to Timer/Counter PR or AC Values

Key Sequence	Display
Move the cursor to the instruction you want to change.	(CTU) 9 0 7
-(CTU)-	PRESET ∗∗∗∗ U

The display is prompting you for a new PR value. If you do not wish to enter a value, press Enter.

6 0	
0	PRESET ∗600
ENTER	ACCUM ∗∗∗∗

If you don't want a new AC value, press Cancel Cmd:

CANCEL CMD	(CTU) 9 0 7

Figure 3.51.

STOP

Re-read paragraphs 3.10–3.10.3. Then answer the questions in Q/E Unit 12 of the Study Guide.

3.11 USING THE EEPROM MEMORY MODULE

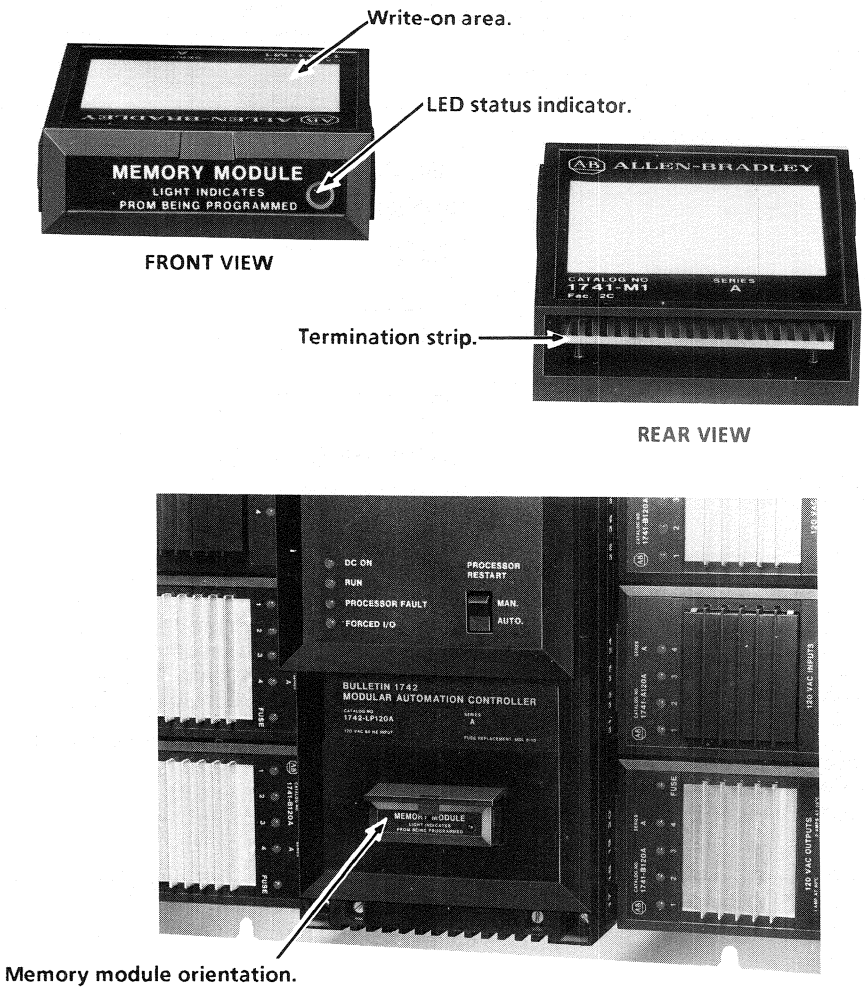
The EEPROM is a non-volatile memory, meaning that it requires no battery back-up to retain its contents. (EEPROM means electrically-eraseable programmable read-only memory.)

The memory can be programmed (loaded) by entering mode 07, which duplicates the contents of the processor RAM memory in the EEPROM. (The EEPROM cannot be programmed by entering individual instructions in sequence, as with the RAM.)

The EEPROM is packaged in a convenient modular form to avoid handling of the integrated circuit. The procedure for installing it in the processor module is shown in Figure 3.52.

IMPORTANT: Always turn off power to the processor module before inserting and before removing the EEPROM. This will guard against possible damage to the EEPROM due to sudden voltage surges at its terminations.

Also, when inserting the EEPROM, be sure the small lip on the access door extends over the front of the EEPROM. This holds the EEPROM securely in place.



Installation procedure:

a) Before inserting the memory module, turn off power to the processor module. This will guard against damage to the memory module due to sudden voltage surges at its terminations.

b) To insert the module, lift the access door on the front of the processor module.

c) Orient the memory module as shown at the left, aligning the termination strip with the receptacle inside the processor module. Push gently to seat the module.

After insertion, be sure that the small lip on the access door extends over the front of the memory module. This holds the module securely in place.

d) Restore power to the processor module.

Figure 3.52.

3.11.1 Loading EEPROM from RAM – To program (load) the EEPROM, follow the procedure shown in the keystroke example of Figure 3.53. This will duplicate the contents of the processor RAM memory in the EEPROM.

The LED indicator on the front of the EEPROM module will be lit during loading. Loading takes 20 milliseconds per word of memory.

A program already in the EEPROM module will be automatically erased and replaced by the new program.

Keystroke Example – Loading a Program into an EEPROM Memory Module

Step 1: Enter your program (or edit an existing program) in the processor RAM memory.

Step 2: Enter mode 07:

Key Sequence	Display
MODE, 7	CURR MODE 01 PROGRAM NEW MODE 07
ENTER	NEW MODE 07 LOAD PROM
ENTER	TO START LOADING PROM PRESS ENTER

Step 3: Turn power off to processor module.

Step 4: Insert EEPROM memory module into receptacle on the processor module. Correct orientation is with the words "MEMORY MODULE" (next to the LED) readable.

Key Sequence	Display

Step 5: Restore power to the processor module.

NOTE: If the EEPROM module has never been been loaded before, error code 06 will appear. In this case, leave the EEPROM in the receptacle and enter mode 07 again (Step 2). Omit Steps 3-4-5, and go on to Step 6.

Step 6: Press Enter:

ENTER	WAIT LOADING PROM IN PROGRESS

LED on memory module will be lit while EEPROM is being loaded.

	LOADING PROM COMPLETE PRESS MODE

NOTE: You cannot enter the Program mode while the EEPROM is inserted. However, we *do* recommend that you enter some appropriate mode (Run or Test) before you remove the EEPROM from the processor module.

IMPORTANT: You must power-down before removing the EEPROM from the processor module.

Figure 3.53.

3.11.2 Copying EEPROM in RAM – To copy the EEPROM contents in the processor RAM, follow the procedure shown in the keystoke example of Figure 3.54. Duplication of the EEPROM contents occurs almost instantaneously.

A program already in the processor RAM memory will be automatically erased and replaced by the new program.

This easy way of duplicating the EEPROM contents in the processor RAM is a time saving convenience in applications where the same program is to be used in a number of processors.

3.11.3 Operating Directly from EEPROM – To operate directly from the EEPROM, turn off power to the processor module, then insert the EEPROM and restore power. The processor RAM contents will be ignored.

The EEPROM program can be monitored by the operator terminal. When operating from the EEPROM, you can force I/O addresses, but you cannot change AC or PR values.

Editing of the EEPROM program cannot be done directly. You must first copy the EEPROM program in the processor RAM, then edit the RAM memory, then duplicate the new program in the EEPROM.

Keystroke Example – Copying a Program from an EEPROM Memory Module to the Processor RAM Memory

Step 1: Enter mode 08.

Key Sequence	Display
MODE, 8	CURR MODE 04 RUN NEW MODE 08
ENTER	NEW MODE 08 COPY PROM
ENTER	TO START COPYING PROM PRESS ENTER

Key Sequence	Display
Step 2: Turn power off to processor module.	
Step 3: Insert EEPROM memory module into receptacle on the processor module. Correct orientation is with the words "MEMORY MODULE" (next to the LED) readable.	
Step 4: Restore power to the processor module.	
Step 5: Press Enter:	
ENTER	COPY PROM COMPLETE PRESS MODE

NOTE: You cannot enter the Program mode while the EEPROM is inserted. However, we *do* recommend that you enter some appropriate mode (Run or Test) before you remove the EEPROM from the processor module.

IMPORTANT: You must power-down before removing the EEPROM from the processor module.

Figure 3.54.

STOP Re-read paragraphs 3.11–3.11.3. Then answer the questions in Q/E Unit 13 of the Study Guide.

Appendix A
BULLETIN 1742 DEMONSTRATOR

DESCRIPTION

If you have access to a Bulletin 1742 Demonstrator, you will find it quite useful as a practice unit for learning to program the Modular Automation Controller.

The demonstrator case houses a completely wired controller, with 8 inputs and 8 outputs. Push buttons, selector switches, and pilot lights are included as I/O devices. Various components are pointed out in Figure A.1.

To set up the demonstrator, remove the top of the case and place the controller in an upright position. Note the hinged support on the back of the case. Fold it out to help guard against overturning the case.

The operator terminal and interconnect cable are inside the compartment in the top of the case. This compartment also contains an EEPROM memory module, a User's Manual, and a demonstrator wiring diagram.

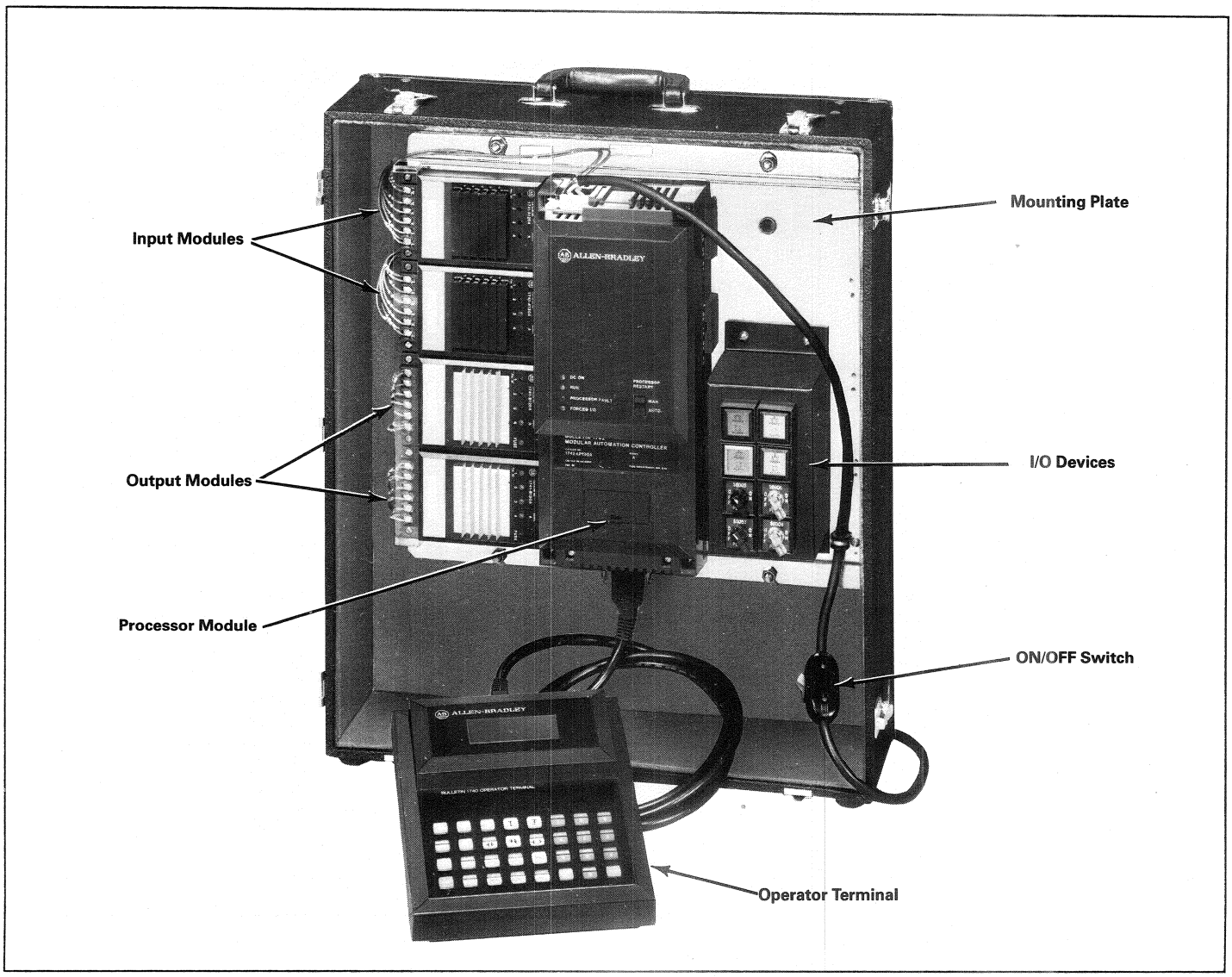


Figure A.1 Bulletin 1742 Demonstrator.

Figure A.2a identifies the I/O devices you will be using. They are located in the control station at the right side of the controller (Figure A.1). This station contains four illuminated push buttons and four illuminated selector switches.

Figure A.2b indicates the external I/O addresses these devices are associated with. The push buttons and selector switches serve as external input devices, while the associated pilot lights serve independently as external output devices.

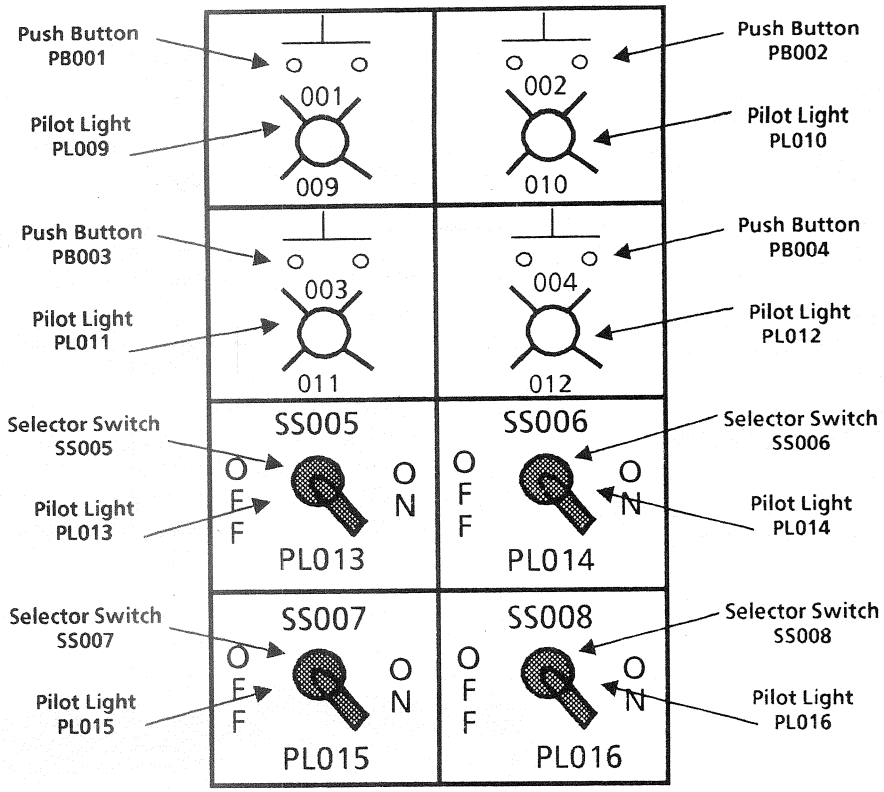


Figure A.2a Control station I/O devices.

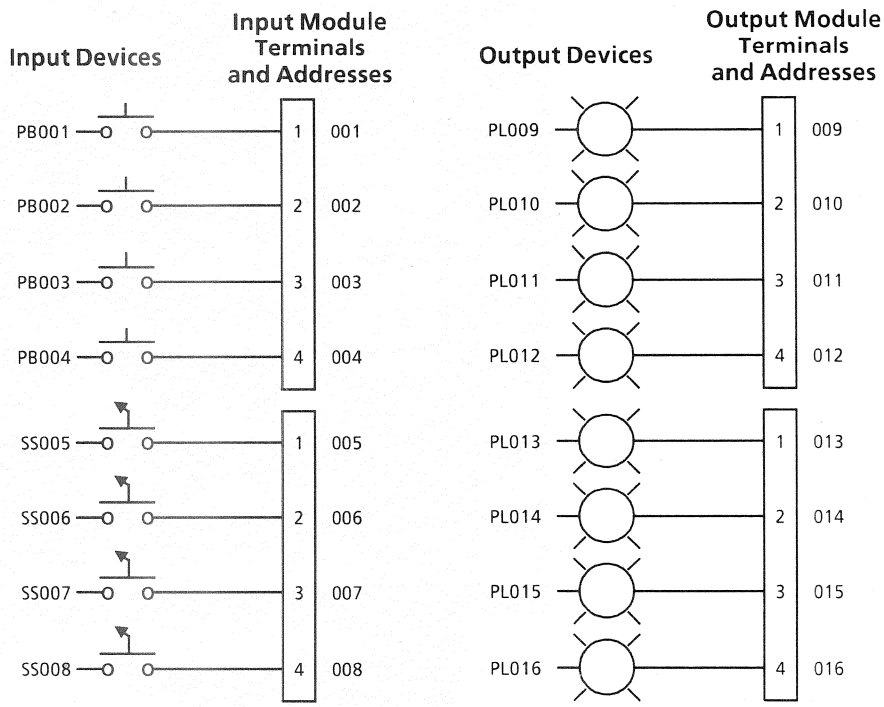


Figure A.2b Addresses of I/O devices.

GETTING STARTED

We assume that you have read the Self-Teach Manual up to paragraph 3.3 and are familiar with the controller components and the operator terminal.

We recommend that you do *not* use the EEPROM module supplied with the demonstrator until you are familiar with the proper procedures outlined in paragraph 3.11.

To begin using the demonstrator, follow these steps:

1. Plug in the operator terminal interconnect cable. Connectors are keyed to guard against improper insertion. When connecting the cable, be sure the spring latches are engaged to secure the cable.
2. Plug the power cord into a grounded 120VAC outlet.

> **CAUTION:** 120VAC power is connected to the processor and the input and output module terminals. Although plastic guards cover these terminals, you should be aware that contact with the 120VAC source voltage can cause serious injury.

3. Turn the line switch on.
4. Press the Mode key on the operator terminal, then enter Mode 20, Clear Memory. This will erase any instructions that may have been programmed previously.
5. Enter the Program Mode (01).

   You can now begin programming your first rung, as described in paragraph 3.3.3.

   Note that in the keystroke sequence of Figure 3.16, you will have to choose an address between 009 and 016 for the output energize instruction, since address 032 is not operational in the demonstrator.

SEQUENCER INSTRUCTION DATA FORM

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☐ ADDRESS 9 : : TIME DRIVEN ☐ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST				2nd Grp	1st Grp	
MODULE GROUP NUMBER →											
I/O TERMINAL ADDRESS →											
EVENT DESCRIPTION MASK DATA →											
STEP NO. 1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

SEQUENCER INSTRUCTION DATA FORM

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☐ ADDRESS 9 : : TIME DRIVEN ☐ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET			
SEQUENCER DATA GROUP →	SECOND				FIRST				2nd Grp	1st Grp				
MODULE GROUP NUMBER →														
I/O TERMINAL ADDRESS →														
EVENT DESCRIPTION MASK DATA →														
STEP NO. 1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														

SEQUENCER INSTRUCTION DATA FORM

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☐ ADDRESS 9 : : TIME DRIVEN ☐ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST				2nd Grp	1st Grp	
MODULE GROUP NUMBER →											
I/O TERMINAL ADDRESS →											
EVENT DESCRIPTION MASK DATA →											
STEP NO. 1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

SEQUENCER INSTRUCTION DATA FORM

SEQUENCER CLASSIFICATION: –(SQI)– ☐ –(SQO)– ☐ ADDRESS 9 : : TIME DRIVEN ☐ EVENT DRIVEN ☐

SEQUENCER DATA									PROGRAM ENTRY CODE		PRESET
SEQUENCER DATA GROUP →	SECOND				FIRST				2nd Grp	1st Grp	
MODULE GROUP NUMBER →											
I/O TERMINAL ADDRESS →											
EVENT DESCRIPTION MASK DATA →											
STEP NO. 1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

ALLEN-BRADLEY

Milwaukee, Wisconsin 53204

Publication 1742-801 — July, 1984