This International Student Edition is for use outside of the U.S.

Sixth Edition

Programmable LOGIC CONTROLLERS

Frank Petruzella

Programmable Logic Controllers

Sixth Edition

Frank D. Petruzella

Mc Graw Hill

PROGRAMMABLE LOGIC CONTROLLERS

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 LWI 27 26 25 24 23 22

ISBN 978-1-265-15049-5
MHID 1-265-15049-4

Cover Image: *Xmentoys/Shutterstock*

mheducation.com/highered

About the Author

Frank D. Petruzella has extensive practical experience in the electrical control field, as well as many years of experience teaching and authoring textbooks. Before becoming a full-time educator, he was employed as an apprentice and electrician in areas of electrical installation and maintenance. He holds a Master of Science degree from Niagara University, a Bachelor of Science degree from the State University of New York College–Buffalo, as well as diplomas in Electrical Power and Electronics from the Erie County Technical Institute.

Contents

Preface

Programmable logic controllers (PLCs) continue to evolve as new technologies are added to their capabilities. As PLC technology has advanced, so have programming languages and communications capabilities. Today's PLCs offer faster scan times, space efficient high-density input/output systems, and special interfaces to allow non-traditional devices to be attached directly to the PLC.

The primary source of information for a particular PLC is always the accompanying user manuals provided by the manufacturer. This textbook is not intended to replace the vendor's reference material, but rather to complement, clarify, and expand on this information. The text covers the basics of programmable logic controllers in a manner that complements instruction with an RSLogix 500 or RSLogix 5000 platform. The underlying PLC principles and concepts covered in the text are common to most manufacturers. They serve to maximize the knowledge gained through on-the-job training and programs offered by different vendors.

The text is written in an easy-to-read style that is designed for students with no prior PLC experience. For example, when the operation of a program is called for, a bulleted list is used to summarize its execution. The bulleted list replaces a lengthy paragraph and is especially helpful when covering the different steps related to the execution of a program.

Each chapter begins with a brief introduction outlining chapter coverage and learning objectives. When applicable, the relay equivalent of the virtual programmed instruction is explained first, followed by the appropriate PLC instruction. Chapters conclude with a set of review questions and problems. The review questions are closely related to the chapter objectives and require students to recall and apply information covered in the chapter. The problems range from easy to difficult, thus challenging students at various levels of competence.

Chapter changes in this edition include:

Chapter 1

- Eliminated pictures of outdated/obsolete equipment.
- Updated programming methods to reflect that laptop computers are now most commonly used.
- Clarified I/O connections for the example in Figures 1-15 and 1-16.
- Illustrated the relative ease of modifying processes in a PLC system compared to conventional methods.
- Updated comparison of PLCs and personal computers.
- Removed obsolete references to memory sizes for specific AB controllers.

Chapter 2

- Removed I/O addressing as a chapter objective.
- Added term: Distributed I/O and definition.
- Added some clarification to tag-based addressing used with RSL5K.
- Added Image of Control Logix controller with analog I/O along with a better description of what analog I/O is.
- Removed references to SCP instruction, since that function is now usually performed when setting up analog channels in module properties.
- Removed references to outdated/obsolete I/O modules.
- Expanded on PLC CPU communication functions.
- Removed obsolete references to memory sizes for specific AB controllers.
- Updated data type sizes used for RSL5K.
- Removed thumbwheel switch problem and replaced with RSL5K tag creation problem.

Chapter 3

- Added more detail to binary math examples.

Chapter 4

- Minimal changes were made in this chapter.

Chapter 5

- Expanded on CPU scan cycle and logic evaluation.
- Removed reference to hand-held programming device.

Chapter 6

- Revised motor control circuit drawing.
- Removed instances of outdated/obsolete equipment.
- Updated sequential process programming example with one from RSL5K.

Chapter 7

- Clarified time base for timers.
- Removed references to coil-based instructions.

Chapter 8

- Removed references to coil-based instructions.

Chapter 9

- Removed obsolete instruction SUS.
- Added Label instruction.
- Removed SLC file references.
- Updated "force" definition.
- Removed section on STI.
- Updated fault routine definition.

Chapter 10

- Removed SLC data file map.
- Updated MOV example.
- Updated A to D converter graphic.

Chapter 11

- Minimal updates were made in this chapter.

Chapter 12

- Updated Sequencer problem.
- Updated question to reflect tag and array type variables.

Chapter 13

- Revised PLC power distribution drawing.
- Revised PLC grounding drawing.

Chapter 14

- Added section on variable frequency drives.

Chapter 15

- Corrected SINT definition.
- Revised timing diagram for OSF.
- Corrected tag name for Part 5 problem 1.

Acknowledgments

I would like to thank the following reviewers for their comments and suggestions:

Mahmoud A. Abdallah
Central State University

Michael Beavers
Lake Land College

Stephen Crosskno
Arkansas Northeastern College

Darrel Cunningham
Oklahoma State University Institute of Technology

Keith Dinwiddie
Ozarks Technical Community College

Charles J. Eckard
North Seattle College

Nathan Esbeck
East Central College

Montie Fleshman
New River Community College

Michael Garner
Lakeland Community College

Steve Gellings
Highland Community College

Jim Heffernan
Quinsigamond Community College

Landon Johnson
Bates Technical College

Gregory Jourdan
Wenatchee Valley College

Maged Mikhail
Purdue University

Jeffery Phelps
Southcentral Kentucky Community and Technical College

Sai Ramaswamy
Wenatchee Valley College

Richard Ramey
Wake Technical Community College

Darrell D. Smith
North Central Texas College

Scott Walker
Utah Valley University

Wade Wittmus
Lakeshore Technical College

Wesley Womack
Central Carolina Community College

Donna Zimmerman
Ivy Tech Community College

A special thanks to Brett Blasiman, Instructor, Electrical Engineering Technology, Gateway Technical College, for his outstanding work on performing a technical edit of the text and providing us with detailed feedback, suggestions and recommendations.

Frank D. Petruzella

P*rogrammable Logic Controllers* **makes it easy to learn PLCs from the ground up! Up-to-the-minute revisions include all the newest developments in programming, installing, and maintaining processes. Clearly developed chapters deliver the organizing objectives, explanatory content with helpful diagrams and illustrations, and closing review problems that evaluate retention of the chapter objectives.**

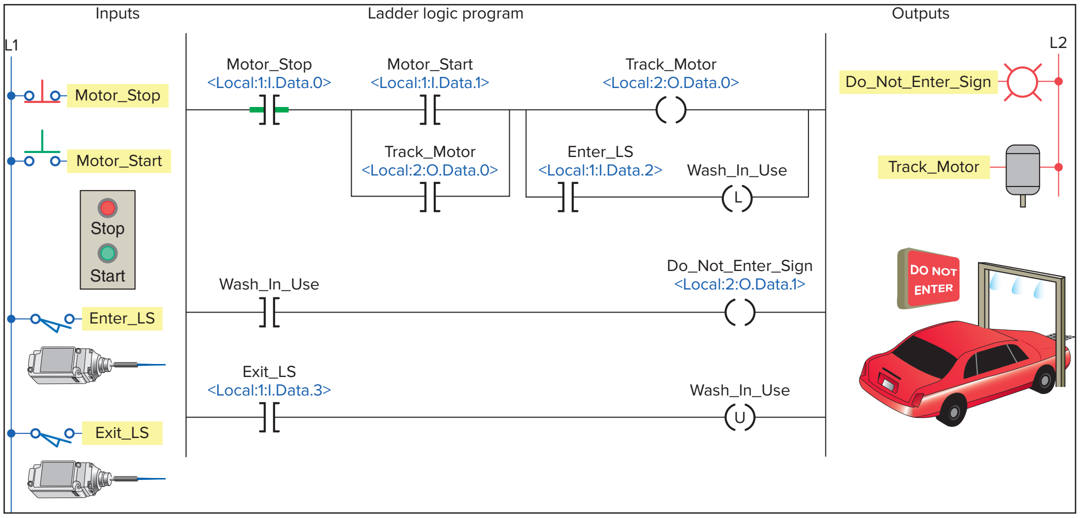
CHAPTER OBJECTIVES overview the chapter, letting students and instructors focus on the main points to better grasp concepts and retain information.

Chapter Objectives

After completing this chapter, you will be able to:

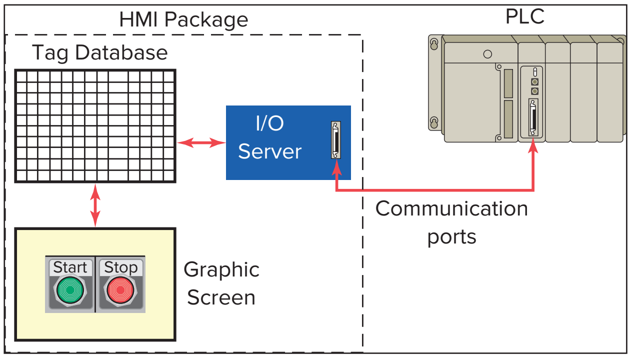
- Describe the operation of pneumatic on-delay and off-delay timers
- Describe PLC timer instruction and differentiate between a nonretentive and retentive timer
- Convert fundamental timer relay schematic diagrams to PLC programs
- Analyze and interpret typical PLC timer programs
- Program the control of outputs using the timer instruction control bits

Chapter content includes rich illustrative detail and extensive visual aids, allowing students to grasp concepts more quickly and understand practical applications

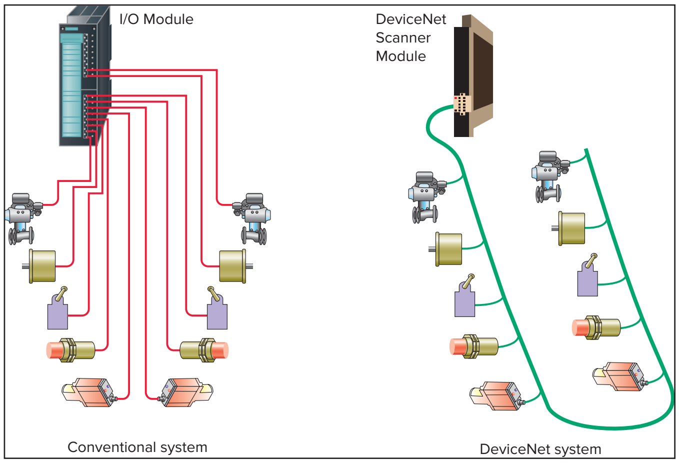


Here, drawings of real-world input and output devices have been included

In chapter 2, students not only read about but can also see how HMIs fit into an overall PLC system, giving them a practical introduction to the topics



Coverage of communications and control networks utilizes clear graphics to demonstrate how things work



A hardwired pilot light motor control circuit is shown in Figure 15-39. The operation of the circuit can be summarized as follows:

- The Stop/Start pushbutton station controls relay coil CR.
- When CR is de-energized, the green standby pilot light is ON, the red run pilot is OFF, and the motor is not operating.

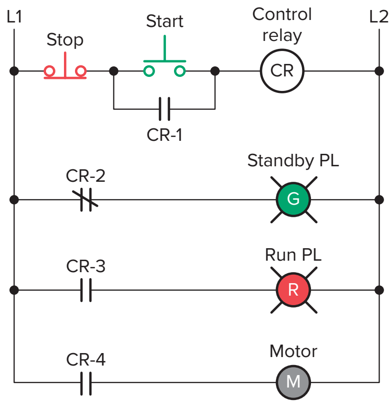


Figure 15-39 Hardwired pilot light motor control circuit

BULLETED LISTS break down processes to helpfully summarize execution of tasks

Diagrams, such as this one illustrating an overview of the function block programming language, help students put the pieces together

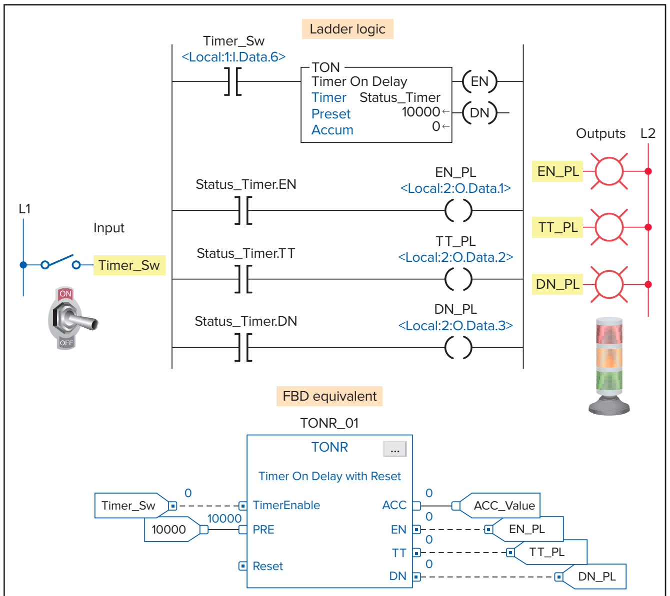
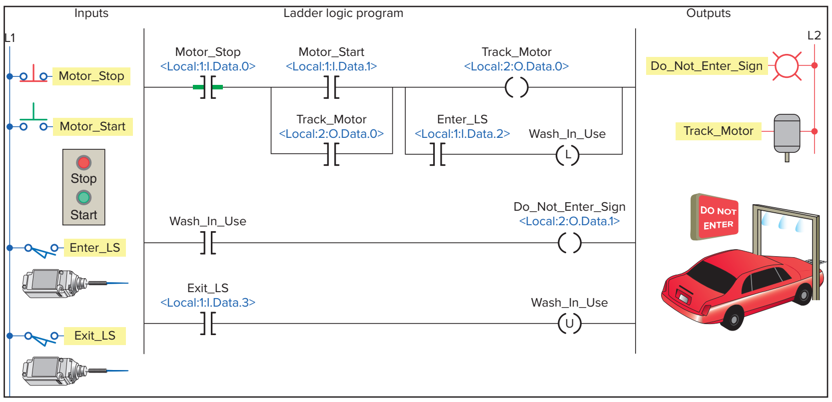


Figure 15-110 Comparison between ladder logic and the FBD equivalent for a 10 second TON and TONR timer.



More than **175** SLC-500 and ControlLogix program simulation **videos tied directly** to the programs studied in the text

END-OF-CHAPTER REVIEWS are structured to reinforce chapter objectives

CHAPTER 6 REVIEW QUESTIONS

1. Explain the basic operating principle of an electromagnetic control relay.
2. What is the operating difference between a normally open and a normally closed relay contact?
3. In what ways are control relay coils and contacts rated?
4. How do contactors differ from relays?
5. What is the main difference between a contactor and a magnetic motor starter?
6. **a.** Draw the schematic for an across-the-line AC magnetic motor starter.
   b. With reference to this schematic, explain the function of each of the following parts:
   - **i.** Main contact M
   - **ii.** Control contact M
   - **iii.** Starter coil M
   - **iv.** OL relay coils
   - **v.** OL relay contact
7. The current requirement for the control circuit of a magnetic starter is normally much smaller than that required by the power circuit. Why?
8. Compare the method of operation of each of the following types of switches:
   a. Manually operated switch
   b. Mechanically operated switch
   c. Proximity switch
15. Compare the operation of the reflective-type and through-beam photoelectric sensors.
16. Give an explanation of how a scanner and a decoder act in conjunction with each other to read a bar code.
17. How does an ultrasonic sensor operate?
18. Explain the principle of operation of a strain gauge.
19. Explain the principle of operation of a thermocouple.
20. What is the most common approach taken with regard to the measurement of fluid flow?
21. Explain how a tachometer is used to measure rotational speed.
22. How does an optical encoder work?
23. Draw an electrical symbol used to represent each of the following PLC control devices:
    a. Pilot light
    b. Relay
    c. Motor starter coil
    d. OL relay contact
    e. Alarm
    f. Heater
    g. Solenoid
    h. Solenoid valve
    i. Motor
    j. Horn
24. Explain the function of each of the following actuators:
    a. Solenoid
    b. Solenoid valve

CHAPTER 6 PROBLEMS

1. Design and draw the schematic for a conventional hardwired relay circuit that will perform each of the following circuit functions when a normally closed pushbutton is pressed:
   - Switch a pilot light on
   - De-energize a solenoid
   - Start a motor running
   - Sound a horn
2. Design and draw the schematic for a conventional hardwired circuit that will perform the following circuit functions using two break-make pushbuttons:
   - Turn on light L1 when pushbutton PB1 is pressed.
   - Turn on light L2 when pushbutton PB2 is pressed.
   - Electrically interlock the pushbuttons so that L1 and L2 cannot both be turned on at the same time.
3. Study the ladder logic program in Figure 6-73, and answer the questions that follow:
   a. Under what condition will the latch rung 1 be true?
   b. Under what conditions will the unlatch rung 2 be true?
   c. Under what condition will rung 3 be true?
   d. When PL1 is on, the relay is in what state (latched or unlatched)?
   e. When PL2 is on, the relay is in what state (latched or unlatched)?
   f. If AC power is removed and then restored to the circuit, what pilot light will automatically come on when the power is restored?
   g. Assume the relay is in its latched state and all three inputs are false. What input change(s) must occur

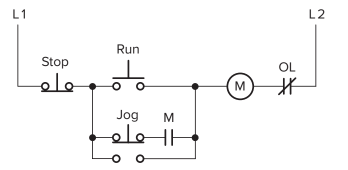
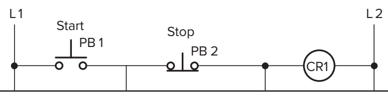


Figure 6-74 Hardwired control circuit for Problem 4.

will correctly execute the hardwired control circuit in Figure 6-74.
Assume: Stop pushbutton used is an NO type.
Run pushbutton used is an NO type.
Jog pushbutton used has one set of NO contacts.
OL contact is hardwired.

5. Design a PLC program and prepare a typical I/O connection diagram and ladder logic program that will correctly execute the hardwired control circuit in Figure 6-75.
   Assume: PB1 pushbutton used is an NO type.
   PB2 pushbutton used is an NC type.
   PS1 pressure switch used is an NO type.
   LS1 limit switch used has only one set of NC contacts.



EXAMPLE PROBLEMS help bring home the applicability of chapter concepts

ANCILLARIES THAT WORK

Expanded on and updated from the previous edition, this new edition includes an outstanding instructor support package available through Connect:

- EZ Test question test banks for each chapter.
- Lab Manual question banks for several chapters.
- Lecture PowerPoints.
- Answers to the questions and problems in the textbook, LogixPro Manual, and RSLogix 5000 Manual.

Proctorio

Remote Proctoring & Browser-Locking Capabilities

Mc Graw Hill connect + proctorio

Remote proctoring and browser-locking capabilities, hosted by Proctorio within Connect, provide control of the assessment environment by enabling security options and verifying the identity of the student.

Seamlessly integrated within Connect, these services allow instructors to control students' assessment experience by restricting browser activity, recording students' activity, and verifying students are doing their own work.

Instant and detailed reporting gives instructors an at-a-glance view of potential academic integrity concerns, thereby avoiding personal bias and supporting evidence-based claims.

ReadAnywhere

Read or study when it's convenient for you with McGraw Hill's free ReadAnywhere app. Available for iOS or Android smartphones or tablets, ReadAnywhere gives users access to McGraw Hill tools including the eBook and SmartBook 2.0 or Adaptive Learning Assignments in Connect. Take notes, highlight, and complete assignments offline – all of your work will sync when you open the app with WiFi access. Log in with your McGraw Hill Connect username and password to start learning – anytime, anywhere!

Connect Course Builds and Edition to Edition Course Rebuilds

Adopters of McGraw Hill content and technology are invited to take advantage of our Connect course build and rebuild services. Instructors who are using Connect for the first time can share their syllabus with McGraw Hill, and McGraw Hill will assist with the initial build of SmartBook readings, assignments, and assessments that align with the instructor's course objectives. If an instructor is already using Connect, when a new edition is published, McGraw Hill is prepared to assist with the transition of the instructor's Connect course to the new edition. McGraw Hill will rebuild the instructor's current Connect course by recreating assignments, mimicking all questions, point values and policy settings, and adding new quiz and test questions to reflect updated content. When the new course is complete, McGraw Hill's Implementation Team is available to review the course with the instructor and highlight new features in Connect.

OLC-Aligned Courses

Implementing High-Quality Online Instruction and Assessment through Preconfigured Courseware

For select titles, in consultation with the Online Learning Consortium (OLC) and our certified Faculty Consultants, McGraw Hill has created pre-configured courseware using OLC's quality scorecard to align with best practices in online course delivery. This turnkey courseware contains a combination of formative assessments, summative assessments, homework, and application activities, and can easily be customized to meet an individual's needs and course outcomes. For more information, visit https://www.mheducation.com/highered/olc.

Tegrity: Lectures 24/7

Tegrity in Connect is a tool that makes class time available 24/7 by automatically capturing every lecture. With a simple one-click start-and-stop process, you capture all computer screens and corresponding audio in a format that is easy to search, frame by frame. Students can replay any part of any class with easy-to-use, browser-based viewing on a PC, Mac, or any mobile device.

Educators know that the more students can see, hear, and experience class resources, the better they learn. In fact, studies prove it. Tegrity's unique search feature helps students efficiently find what they need, when they need it, across an entire semester of class recordings. Help turn your students' study time into learning moments immediately supported by your lecture. With Tegrity, you also increase intent listening and class participation by easing students' concerns about note-taking. Using Tegrity in Connect will make it more likely you will see students' faces, not the tops of their heads.

Test Builder in Connect

Available within Connect, Test Builder is a cloud-based tool that enables instructors to format tests that can be printed, administered within a Learning Management System, or exported as a Word document of the test bank. Test Builder offers a modern, streamlined interface for easy content configuration that matches course needs, without requiring a download.

Test Builder allows you to:

- access all test bank content from a particular title.
- easily pinpoint the most relevant content through robust filtering options.
- manipulate the order of questions or scramble questions and/or answers.
- pin questions to a specific location within a test.
- determine your preferred treatment of algorithmic questions.
- choose the layout and spacing.
- add instructions and configure default settings.

Test Builder provides a secure interface for better protection of content and allows for just-in-time updates to flow directly into assessments.

Writing Assignment

Available within Connect and Connect Master, the Writing Assignment tool delivers a learning experience to help students improve their written communication skills and conceptual understanding. As an instructor you can assign, monitor, grade, and provide feedback on writing more efficiently and effectively.

Application-Based Activities in Connect

Application-Based Activities in Connect are highly interactive, assignable exercises that provide students with a safe space to apply the concepts they have learned to real-world, course-specific problems. Each Application-Based Activity involves the application of multiple concepts, allowing students to synthesize information and use critical thinking skills to solve realistic scenarios.

Create

Your Book, Your Way

McGraw Hill's Content Collections Powered by Create® is a self-service website that enables instructors to create custom course materials—print and eBooks—by drawing upon McGraw Hill's comprehensive, cross-disciplinary content. Choose what you want from our high-quality textbooks, articles, and cases. Combine it with your own content quickly and easily, and tap into other rights-secured, third-party content such as readings, cases, and articles. Content can be arranged in a way that makes the most sense for your course and you can include the course name and information as well. Choose the best format for your course: color print, black-and-white print, or eBook. The eBook can be included in your Connect course and is available on the free ReadAnywhere app for smartphone or tablet access as well. When you are finished customizing, you will receive a free digital copy to review in just minutes! Visit McGraw Hill Create®—www.mcgrawhillcreate.com—today and begin building!

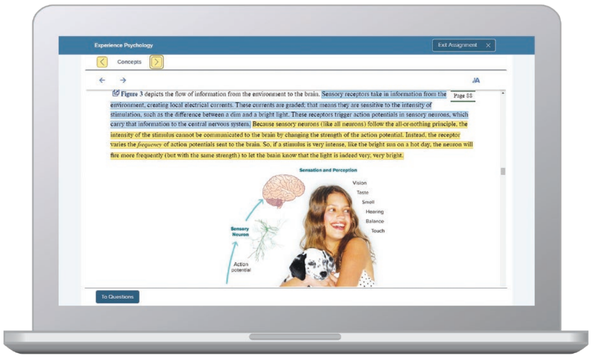
Mc Graw Hill connect®

Instructors: Student Success Starts with You

Tools to enhance your unique voice

Want to build your own course? No problem. Prefer to use an OLC-aligned, prebuilt course? Easy. Want to make changes throughout the semester? Sure. And you'll save time with Connect's auto-grading too.

65%
Less Time Grading



Laptop: McGraw Hill; Woman/dog: George Doyle/Getty Images

Study made personal

Incorporate adaptive study resources like SmartBook® 2.0 into your course and help your students be better prepared in less time. Learn more about the powerful personalized learning experience available in SmartBook 2.0 at **www.mheducation.com/highered/connect/smartbook**

Affordable solutions, added value

Make technology work for you with LMS integration for single sign-on access, mobile access to the digital textbook, and reports to quickly show you how each of your students is doing. And with our Inclusive Access program you can provide all these tools at a discount to your students. Ask your McGraw Hill representative for more information.

Padlock: Jobalou/Getty Images

Solutions for your challenges

A product isn't a solution. Real solutions are affordable, reliable, and come with training and ongoing support when you need it and how you want it. Visit **www.supportateverystep.com** for videos and resources both you and your students can use throughout the semester.

Checkmark: Jobalou/Getty Images

Students: Get Learning that Fits You

Effective tools for efficient studying

Connect is designed to help you be more productive with simple, flexible, intuitive tools that maximize your study time and meet your individual learning needs. Get learning that works for you with Connect.

Study anytime, anywhere

Download the free ReadAnywhere app and access your online eBook, SmartBook 2.0, or Adaptive Learning Assignments when it's convenient, even if you're offline. And since the app automatically syncs with your Connect account, all of your work is available every time you open it. Find out more at **www.mheducation.com/readanywhere**

> ***"I really liked this app—it made it easy to study when you don't have your text-book in front of you."***
>
> - Jordan Cunningham,
> Eastern Washington University

Calendar: owattaphotos/Getty Images

Everything you need in one place

Your Connect course has everything you need—whether reading on your digital eBook or completing assignments for class, Connect makes it easy to get your work done.

Learning for everyone

McGraw Hill works directly with Accessibility Services Departments and faculty to meet the learning needs of all students. Please contact your Accessibility Services Office and ask them to email accessibility@mheducation.com, or visit **www.mheducation.com/about/accessibility** for more information.

Top: Jenner Images/Getty Images, Left: Hero Images/Getty Images, Right: Hero Images/Getty Images

1

Programmable Logic Controllers (PLCs): An Overview

Image Courtesy of Rockwell Automation, Inc.

Chapter Objectives

After completing this chapter, you will be able to:

- Define what a programmable logic controller (PLC) is and list its advantages over relay systems
- Identify the main parts of a PLC and describe their functions
- Outline the basic sequence of operation for a PLC
- Identify the general classifications of PLCs

This chapter gives a brief history of the evolution of the programmable logic controller, or PLC. The reasons for changing from relay control systems to PLCs are discussed. You will learn the basic parts of a PLC, how a PLC is used to control a process, and the different kinds of PLCs and their applications. The ladder logic language, which was developed to simplify the task of programming PLCs, is introduced.

1.1 Programmable Logic Controllers

Programmable logic controllers are now the most widely used industrial process control technology. A **programmable logic controller (PLC)** is an industrial grade computer that is capable of being programmed to perform control functions. The programmable controller has eliminated much of the hardwiring associated with conventional relay control circuits. Other benefits include fast response, easy programming and installation, high control speed, network compatibility, troubleshooting and testing convenience, and high reliability.

The PLC is designed for multiple input and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs for the control and operation of manufacturing process equipment and machinery are typically stored in battery-backed or nonvolatile memory. A PLC is an example of a **real-time system** since the output of the system controlled by the PLC depends on the input conditions.

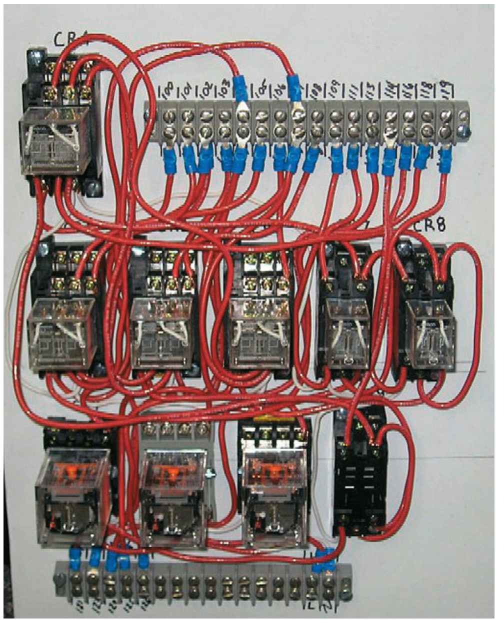
The PLC is, then, basically a digital computer designed for use in machine control. Unlike a personal computer, it has been designed to operate in the industrial environment and is equipped with special input/output interfaces and a control programming language.

Initially the PLC was used to replace **relay logic**, but its ever-increasing range of functions means that it is found in many and more complex applications. Because the structure of a PLC is based on the same principles as those employed in computer architecture, it is capable not only of performing relay switching tasks but also of performing other applications such as timing, counting, calculating, comparing, and the processing of analog signals.

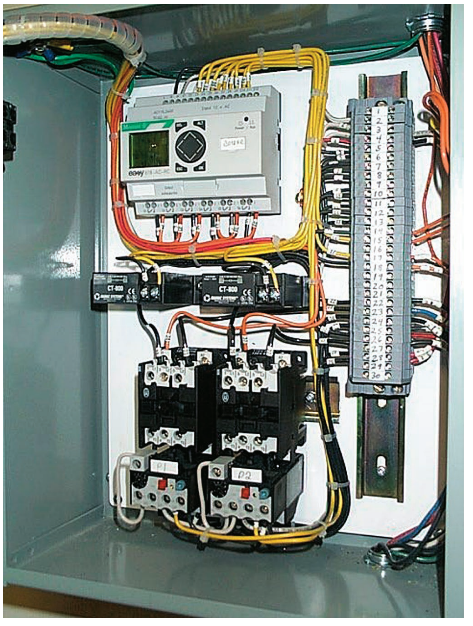
Programmable controllers offer several advantages over a conventional relay type of control. Relays have to be hardwired to perform a specific function. When the system requirements change, the relay wiring has to be changed or modified. In extreme cases, such as in the auto industry, complete control panels had to be replaced since it was not economically feasible to rewire the old panels with each model changeover. The programmable controller has eliminated much of the hardwiring associated with conventional relay control circuits (Figure 1-1). It is small and inexpensive compared to equivalent relay-based process control systems. Modern control systems still include relays, but these are rarely used for logic.

PLCs provide many other benefits including:

- ***Increased Reliability.*** Once a program has been written and tested, it can be easily downloaded to other PLCs. Since all the logic is contained in the PLC's memory, there is no chance of making a logic wiring error (Figure 1-2). The program takes the



(*a*)



(*b*)

Figure 1-1 Relay- and PLC-based control panels. (*a*) Relay-based control panel. (*b*) PLC-based control panel.
Source: (*a*) Courtesy Mid-Illini Technical Group, Inc.; (*b*) Photo courtesy of Ramco Electric Ltd., Toronto

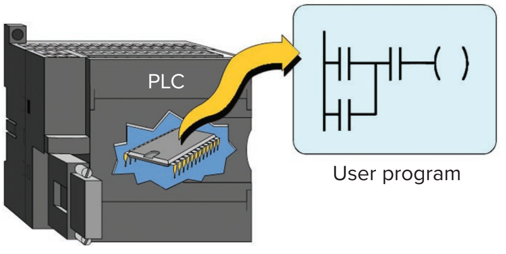


Figure 1-2 All the logic is contained in the PLC's memory.

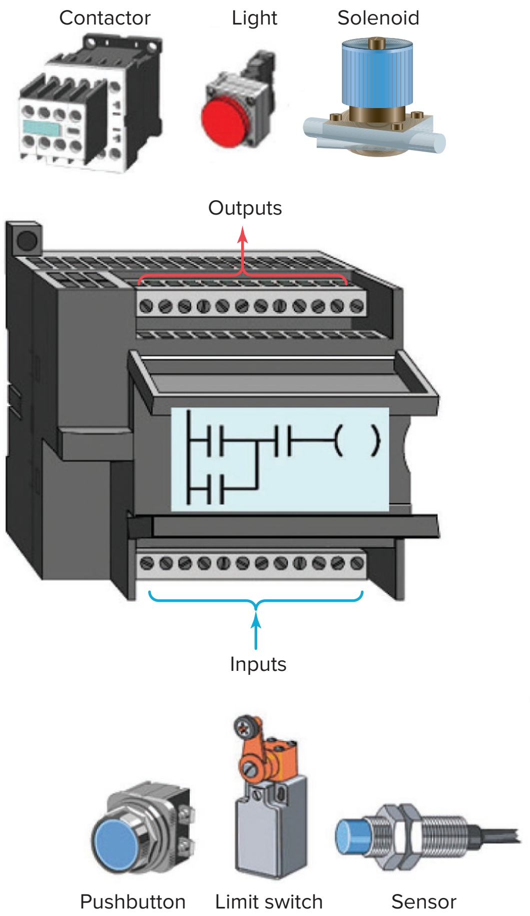


Figure 1-3 Relationships between the inputs and outputs are determined by the user program.

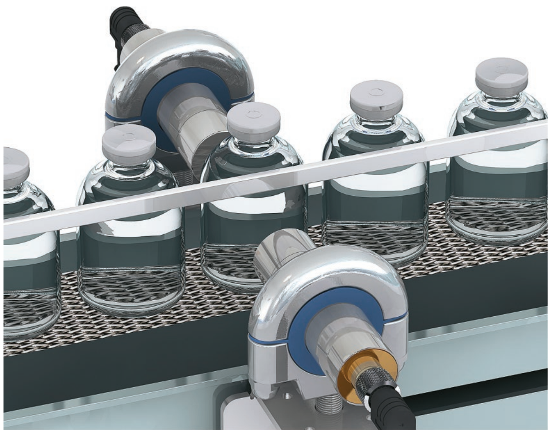


Figure 1-4 High-speed counting.
Source: Courtesy of Banner Engineering Corp.

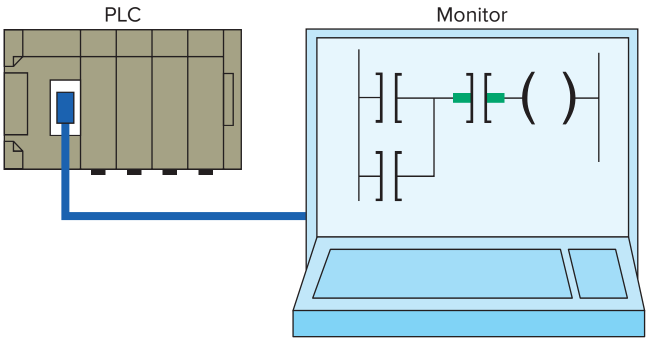


Figure 1-5 Control program can be displayed on a monitor in real time.

place of much of the external wiring that would normally be required for control of a process. Hardwiring, though still required to connect field devices, is less intensive. PLCs also offer the reliability associated with solid-state components.

- ***More Flexibility.*** It is easier to create and change a program in a PLC than to wire and rewire a circuit. With a PLC the relationships between the inputs and outputs are determined by the user program instead of the manner in which they are interconnected (Figure 1-3). Original equipment manufacturers can provide system updates by simply sending out a new program. End users can modify the program in the field, or if desired, security can be provided by hardware features such as key locks and by software passwords.
- ***Lower Cost.*** PLCs were originally designed to replace relay control logic, and the cost savings have been so significant that relay control is becoming obsolete except for power applications. Generally, if an application has more than about a half-dozen control relays, it will probably be less expensive to install a PLC.
- ***Communications Capability.*** A PLC can communicate with other controllers or computer equipment to perform such functions as supervisory control, data gathering, monitoring devices and process parameters, and download and upload of programs.
- ***Faster Response Time.*** PLCs are designed for high-speed and real-time applications (Figure 1-4). The programmable controller operates in real time, which means that an event taking place in the field will result in the execution of an operation or output. Machines that process thousands of items per second and objects that spend only a fraction of a second in front of a sensor require the PLC's quick-response capability.
- ***Easier to Troubleshoot.*** PLCs have resident diagnostics and override functions that allow users to easily trace and correct software and hardware problems. To find and fix problems, users can display the control program on a monitor and watch it in real time as it executes (Figure 1-5).

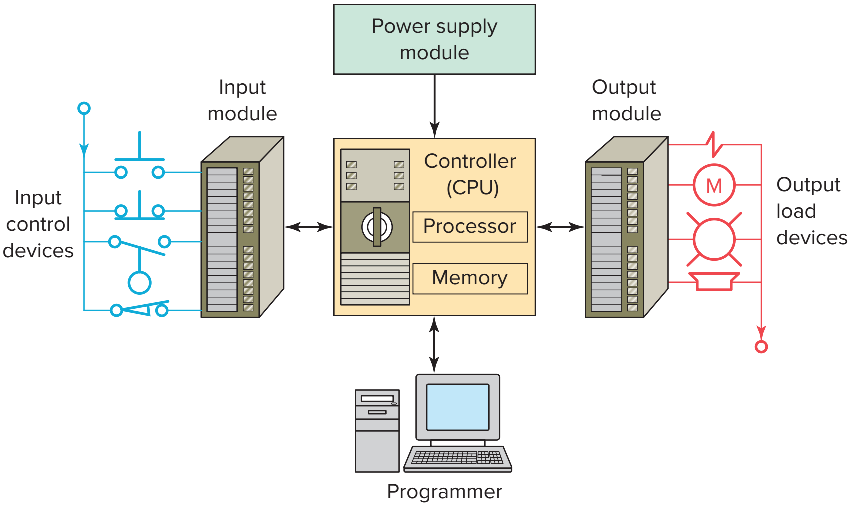


Figure 1-6 Typical parts of a programmable logic controller.

- **Easier to Test Field Devices.** A PLC control panel has the ability to check field devices at a common point. For example, a control system consisting of hundreds of input and output field devices may be contained within a very large manufacturing area. Thus, it would take a considerable amount of time to check each device at its location. By having each device wired back to a common point on a PLC module, each device could be checked for operation fairly quickly.

1.2 Parts of a PLC

A typical PLC can be divided into parts, as illustrated in Figure 1-6. These are the *central processing unit (CPU)*, the *input/output (I/O)* section, the *power supply,* and the *programming device*. The term *architecture* can refer to PLC hardware, to PLC software, or to a combination of both. An *open* architecture design allows the system to be connected easily to devices and programs made by other manufacturers. Open architectures use off-the-shelf components that conform to approved standards. A system with a *closed* architecture is one whose design is proprietary, making it more difficult to connect to other systems. Most PLC systems are in fact **proprietary**, so you must be sure that any generic hardware or software you may use is compatible with your particular PLC. Also, although the principal concepts are the same in all methods of programming, there are differences in addressing, memory allocation, retrieval, and data handling for different models. Consequently, PLC programs cannot be interchanged among different PLC manufacturers.

There are two ways in which I/Os (Inputs/Outputs) are incorporated into the PLC: fixed and modular. ***Fixed I/O*** (Figure 1-7) is typical of small PLCs that come in one package with no separate, removable units. The processor and I/O are packaged together, and the I/O terminals will have a fixed number of connections built in for inputs and outputs. The main advantage of this type of packaging is lower cost. The number of available I/O points varies and usually can be expanded by buying additional units of fixed I/O. One disadvantage of fixed I/O is its lack of flexibility; you are limited in what you can get in the quantities and types dictated by the packaging. Also, for some models, if any part in the unit fails, the whole unit has to be replaced.

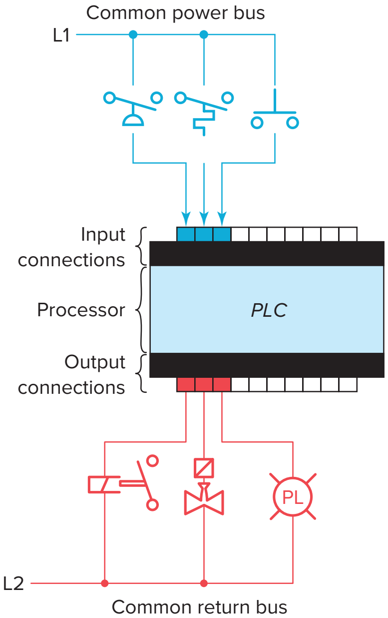


Figure 1-7 Fixed I/O configuration.
Source: Image Courtesy of Rockwell Automation, Inc

Modular I/O (Figure 1-8) is divided by compartments into which separate modules can be plugged. This feature greatly increases your options and the unit's flexibility. You can choose from the modules available from the manufacturer and mix them any way you desire. The basic modular controller consists of a rack, power supply, processor module (CPU), input/output (I/O modules), and an operator interface for programming and monitoring. The modules plug into a rack. When a module is slid into the rack, it makes an electrical connection with a series of contacts called the backplane, located at the rear of the rack. Communication between modules is accomplished by the backplane rail that each module plugs into.

The ***power supply*** supplies DC power through the backplane, to the processor and the other modules that plug into the rack (Figure 1-9). For large PLC systems, this power supply does not normally supply power to the field devices. With most systems, power to field devices is provided by external alternating current (AC) or direct current (DC) supplies. For some small micro PLC systems, the power supply may be used to power field devices.

The ***processor* (CPU)** is the "brain" of the PLC. A typical processor usually consists of a microprocessor for implementing the logic and controlling the communications among the modules. The processor requires memory for storing user program instructions, numerical values, and I/O devices status.

The CPU controls all PLC activity and is responsible for running the program. The PLC program is executed as part of a repetitive process referred to as a scan (Figure 1-10). A typical PLC scan starts with the CPU reading the status of inputs. Then, the program logic is executed. Once the

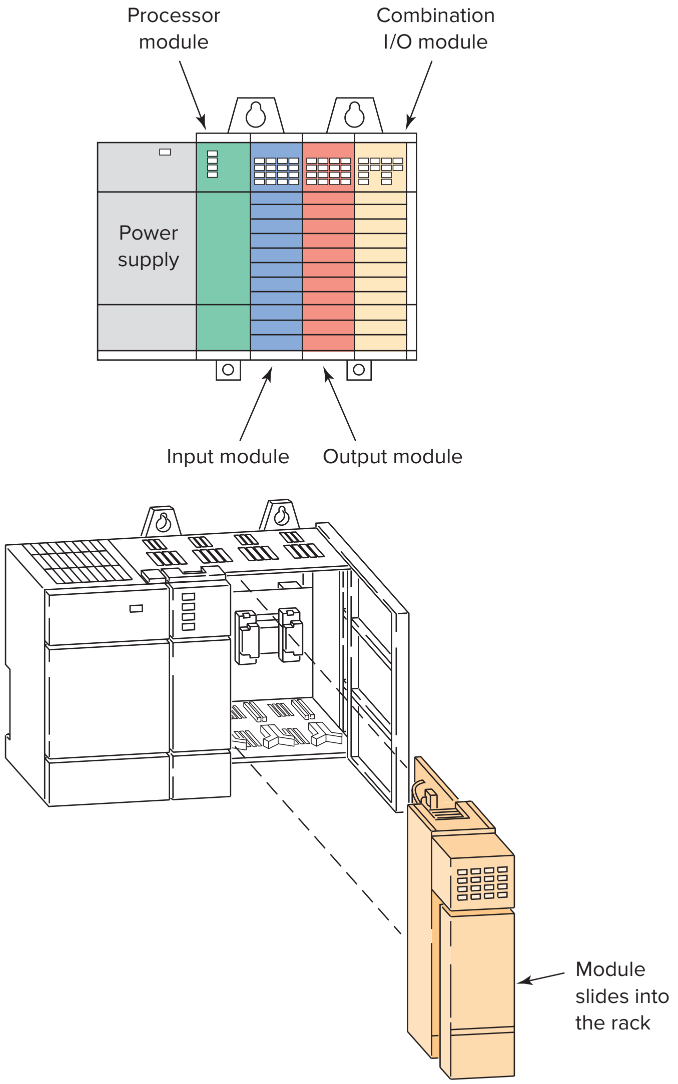


Figure 1-8 Modular I/O configuration.

Figure 1-9 The power supply supplies DC power to other modules that plug into the rack.

Source: Photo of PLC Modicon M340 © Schneider Electric, 2010. **www.schneider-electric.com**.

Figure 1-10 Typical PLC scan cycle.

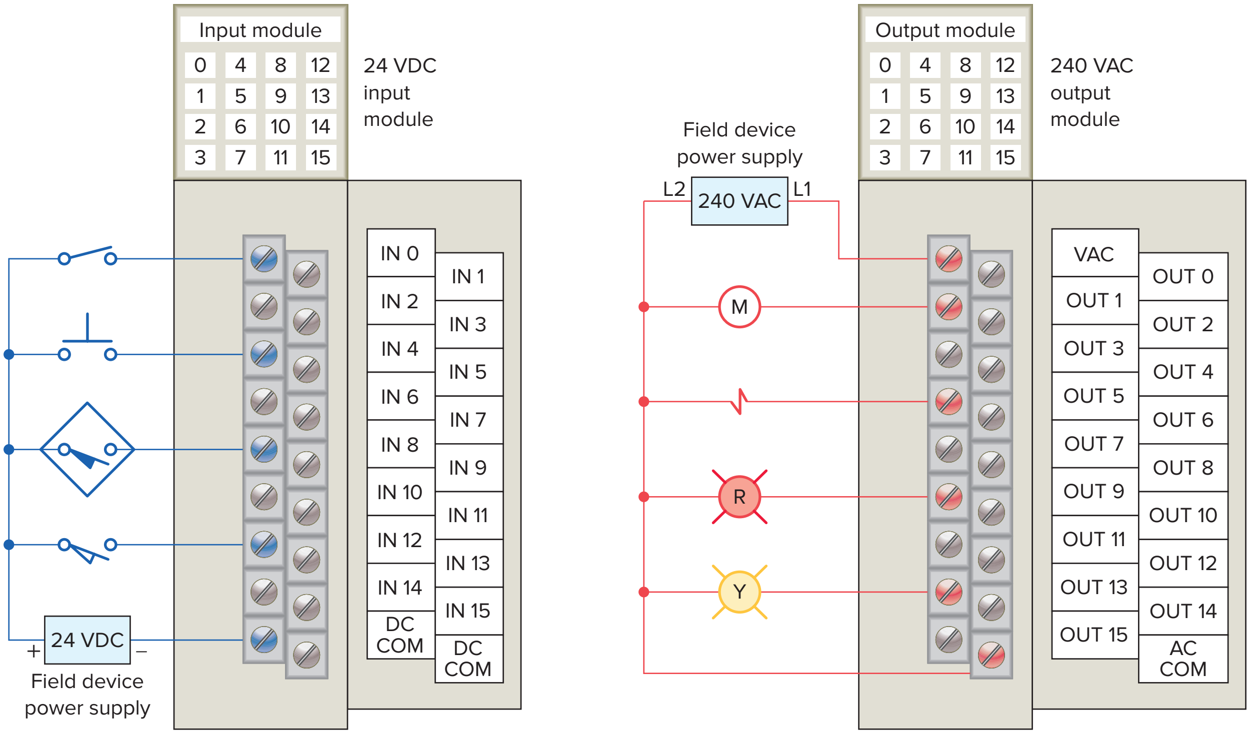


Figure 1-11 Typical PLC input/output (I/O) system connections.

program execution is completed, the status of all outputs is updated. Next, the CPU performs internal diagnostic and communication tasks. This process is repeated continuously as long as the PLC is in the run mode.

The ***I/O system*** forms the interface by which field devices are connected to the controller (Figure 1-11). The purpose of this interface is to condition the various signals received from or sent to external field devices. Input devices such as pushbuttons, limit switches, and sensors are hardwired to the input terminals. Output devices such as small motors, motor starters, solenoid valves, and indicator lights are hardwired to the output terminals. To electrically isolate the internal components from the input and output terminals, PLCs commonly employ an optical isolator, which uses light to couple the circuits together. The external devices are also referred to as "field" or "real-world" inputs and outputs. The terms ***field*** or ***real world*** are used to distinguish actual external devices that exist and must be physically wired from the internal user program that emulates the function of relays, timers, and counters.

A ***programming device*** is used to interface with the PLC in order to develop and transfer logic programs, download or upload data, or supply diagnostic functions to troubleshoot PLC systems. The device may be a dedicated handheld type or a personal computer running special application software. Removing the programming device will not affect the operation of the program.

A laptop computer is the most commonly used programming device. Most brands of PLCs have software available so that a laptop can be used as the programming device. This software allows users to create, edit, document, store, and troubleshoot ladder logic programs. The computer monitor is able to display more logic on the screen than can hand-held types, thus simplifying the interpretation of the program. The programming device communicates with the PLC processor via a serial or parallel data communications link, or Ethernet.

The logic ***program*** is a user-developed series of instructions that directs the PLC to execute actions. A *programming language* provides rules for combining the instructions so that they produce the desired actions. ***Relay ladder logic (RLL)*** is the standard programming language used with PLCs. Its origin is based on electromechanical relay control. The relay ladder logic program graphically represents rungs of contacts, coils, and special instruction blocks. RLL was originally designed for easy use and understanding for its users and has been modified to keep up with the increasing demands of industry's control needs.

1.3 Principles of Operation

To get an idea of how a PLC operates, consider the simple process control problem illustrated in Figure 1-12. Here a mixer motor is to be used to automatically stir the liquid in a vat when the temperature and pressure reach

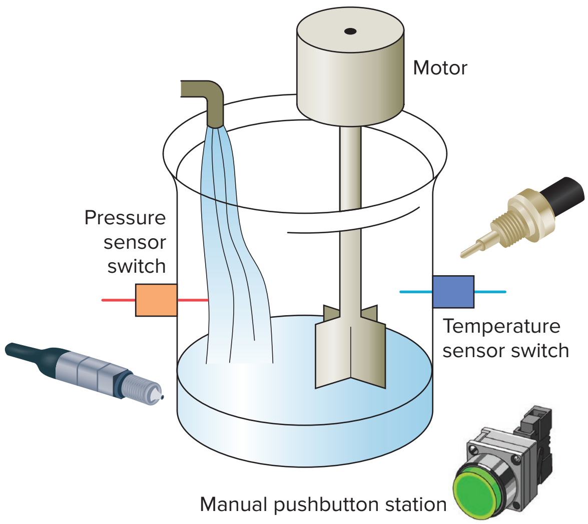


Figure 1-12 Mixer process control problem.

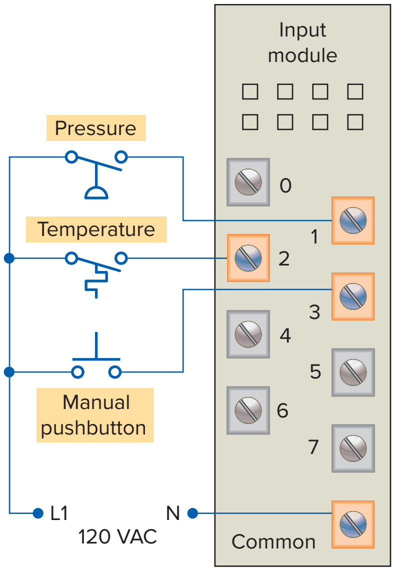


Figure 1-14 Typical wiring connections for a 120 VAC modular configured input module.

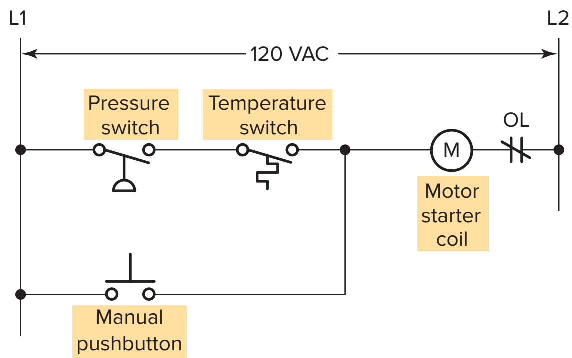


Figure 1-13 Process control relay ladder diagram.

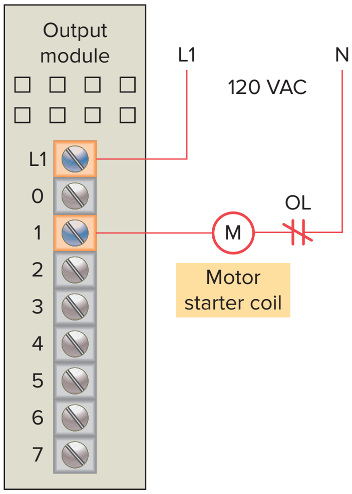


Figure 1-15 Typical wiring connections for a 120 VAC modular configured output module.

preset values. In addition, direct manual operation of the motor is provided by means of a separate pushbutton station. The process is monitored with temperature and pressure sensor switches that close their respective contacts when conditions reach their preset values.

This control problem can be solved using the relay method for motor control shown in the relay ladder diagram of Figure 1-13. The motor starter coil (M) is energized when both the pressure and temperature switches are closed or when the manual pushbutton is pressed.

Now let's look at how a programmable logic controller might be used for this application. The same input field devices (pressure switch, temperature switch, and pushbutton) are used. Each of these devices is wired to a terminal on the 120 VAC input module as shown in Figure 1-14. The processor memory location addresses used are:

I/1 for the Pressure switch
I/2 for the Temperature switch
I/3 for the Manual pushbutton

The same output field device (motor starter coil) would also be used. This device is wired to a terminal on the 120 VAC output module as shown in Figure 1-15. The processor memory location address used for the Motor starter coil is O/1.

Next, the PLC ladder logic program would be constructed and entered into the memory of the CPU. A typical ladder logic program for this process is shown in Figure 1-16. The format used is similar to the layout of the hardwired relay ladder circuit. The individual symbols represent instructions, whereas the numbers represent the instruction memory location addresses. To program the controller, you enter these instructions one by one into the processor memory from the programming

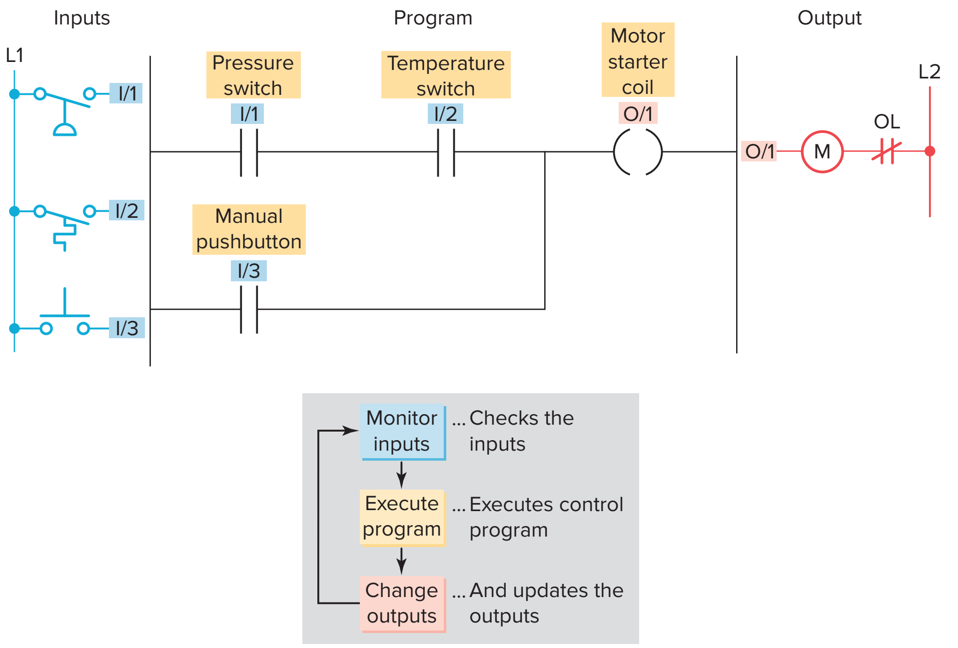


Figure 1-16 Process control PLC ladder logic program with typical addressing scheme.

device. Each input and output device is given an address, which lets the PLC know where it is physically connected. Note that the I/O address format will differ, depending on the PLC model and manufacturer. Instructions are stored in the user program portion of the processor memory. During the program scan the controller monitors the inputs, executes the control program, and changes the output accordingly.

For the program to operate, the controller is placed in the RUN mode, or operating cycle. During the program scan, the controller monitors the inputs, executes the control program, and changes the output accordingly. Each -| |- symbol (looks like a normally open contact) is an instruction. The ( ) symbol is considered to represent a coil that, when energized, will energize the device that is wired to the respective output. In the ladder logic program of Figure 1-16, the coil O/1 is energized when contacts I/1 and I/2 are closed or when contact I/3 is closed. Either of these conditions provides a continuous logic path from left to right across the rung that includes the coil.

A programmable logic controller operates in real time in that an event taking place in the field will result in an operation or output taking place. The RUN operation for the process control logic can be described by the following sequence of events:

- First, the pressure switch, temperature switch, and pushbutton inputs are examined and their status is recorded in the controller's memory.
- A closed contact is recorded in memory as logic 1 and an open contact as logic 0.
- Next the ladder diagram is evaluated, with each internal contact given an OPEN or CLOSED status according to its recorded 1 or 0 state.
- When the states of the input contacts provide logic continuity from left to right across the rung, the output coil memory location is given a logic 1 value and the output module interface contacts will close.
- When there is no logic continuity of the program rung, the output coil memory location is set to logic 0 and the output module interface contacts will be open.
- The completion of one cycle of this sequence by the controller is called a *scan.* The scan time, the time required for one full cycle, provides a measure of the speed of response of the PLC.
- Generally, the output memory location is updated during the scan but the actual output is not updated until the end of the program scan during the I/O scan.

Figure 1-17 shows the typical wiring required to implement the process control scheme using a fixed PLC controller. In this example, the Allen-Bradley Pico controller equipped with 8 inputs and 4 outputs is used to control and monitor the process. Installation can be summarized as follows:

- Fused power lines, of the specified voltage type and level, are connected to the controller's L1 and L2 terminals.
- The pressure switch, temperature switch, and pushbutton field input devices are hardwired between L1

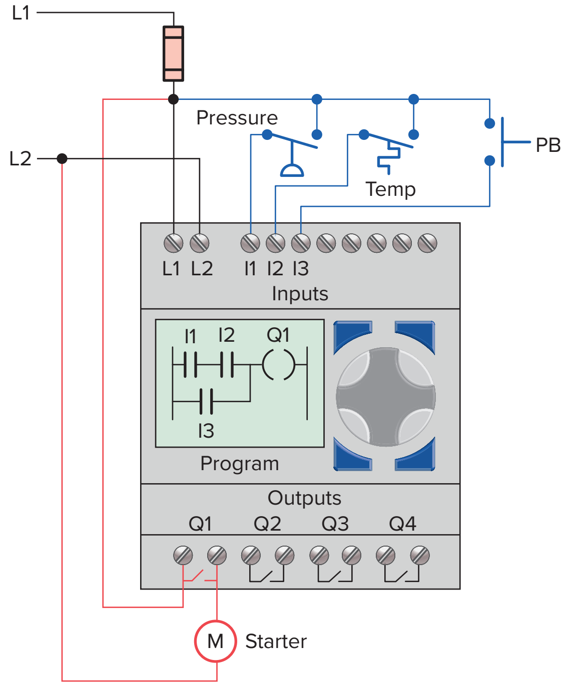


Figure 1-17 Typical wiring required to implement the process control scheme using a fixed PLC controller.

and controller input terminals I1, I2, and I3, respectively.

- The motor starter coil connects directly to L2 and in series with Q1 relay output contacts to L1.
- The ladder logic program is entered using the front keypad and LCD display.
- Pico programming software is also available that allows you to create as well as test your program using a personal computer.

1.4 Modifying the Operation

One of the important features of a PLC is the ease with which the program can be changed. For example, assume that the original process control circuit for the mixing operation must be modified as shown in the relay ladder diagram of Figure 1-18. The change requires that the manual pushbutton control be permitted to operate at any pressure, but not unless the specified temperature setting has been reached.

If a relay system were used, it would require some rewiring of the circuit shown in Figure 1-18 to achieve the desired change. However, if a PLC system were used, no rewiring would be necessary. The inputs and outputs are still the same. All that is required is to change the PLC ladder logic program as shown in Figure 1-19.

At times, a process may call for additional real input or output field devices to be added to the circuit. The

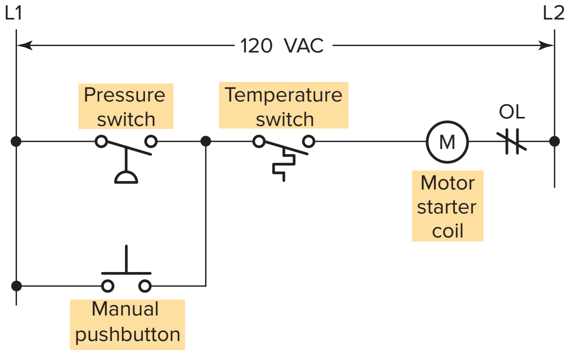


Figure 1-18 Relay ladder diagram for the modified process.

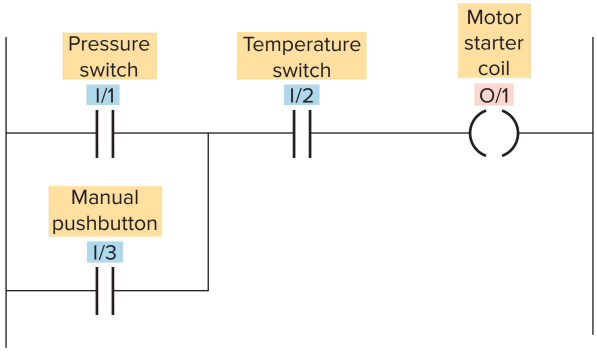


Figure 1-19 PLC ladder logic program for the modified process.

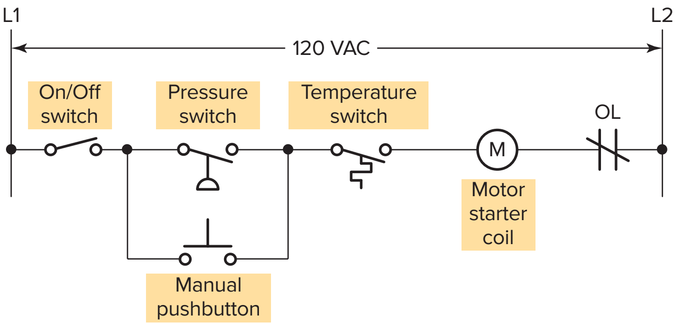


Figure 1-20 Modified relay ladder for the addition of an ON/OFF switch.

relay ladder diagram of Figure 1-20 shows the circuit further modified to include a process ON/OFF switch. To accomplish this, using hard-wired circuit control requires accessing both the pressure switch and manual pushbutton and rewiring the circuit so that the two are in parallel with each other and in series with the ON/OFF switch. The modification implemented using a PLC ladder logic program is shown in Figure 1-21. Note that the original wiring of the existing PLC input circuit remains the same. All that is required is the connection of the ON/OFF switch to I/4 input and the related change in the PLC program.