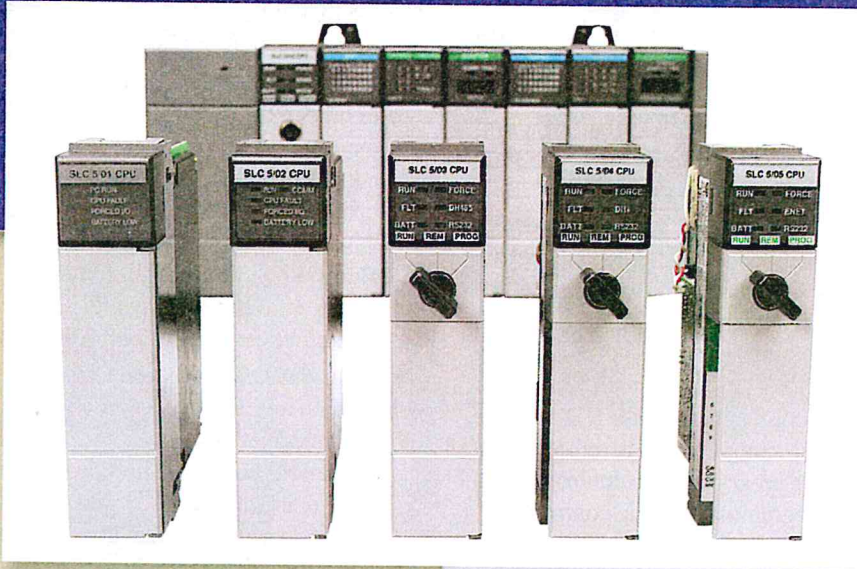


# 1

## Programmable Logic Controllers (PLCs) An Overview



*Image Used with Permission of Rockwell Automation, Inc.*

### Chapter Objectives

After completing this chapter, you will be able to:

- 1.1** Define what a programmable logic controller (PLC) is and list its advantages over relay systems
- 1.2** Identify the main parts of a PLC and describe their functions
- 1.3** Outline the basic sequence of operation for a PLC
- 1.4** 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 (Figure 1-1) 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 easy programming and installation, high control speed, network compatibility, troubleshooting and testing convenience, and high reliability.

The programmable logic controller 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 non-volatile 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.

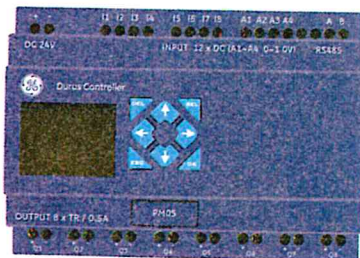
The programmable logic controller is, then, basically a digital computer designed for use in machine control. Unlike a personal computer, it has been designed to operate in the industrial environment and is equipped with special input/output interfaces and a control programming language. The common abbreviation used in industry for these devices, PC, can be confusing because it is also the abbreviation for "personal computer." Therefore, most manufacturers refer to their programmable controller as a PLC, which stands for "programmable logic controller."

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.

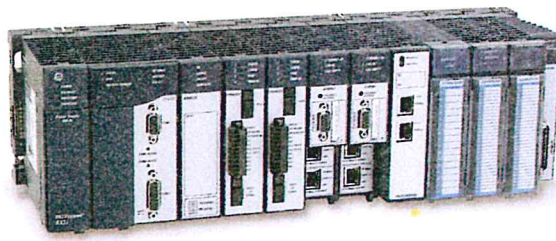
Programmable controllers offer several advantages over a conventional relay type of control. Relays have to be hardwired to perform a specific function. When the system requirements change, the relay wiring has to be changed or modified. In extreme cases, such as in the auto industry, complete control 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 1-2). 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.

In addition to cost savings, 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 1-3). The program takes the 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 1-4). 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



(a)

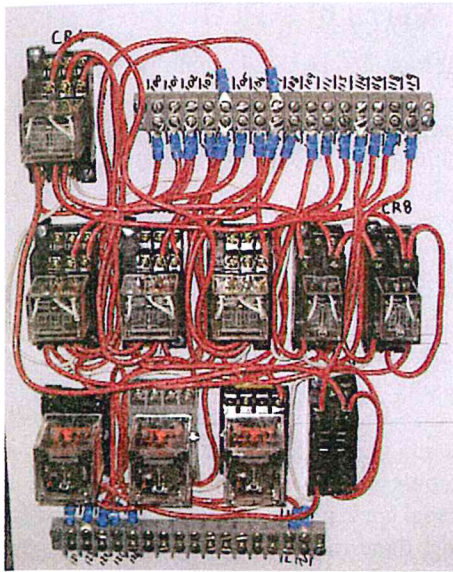


(b)

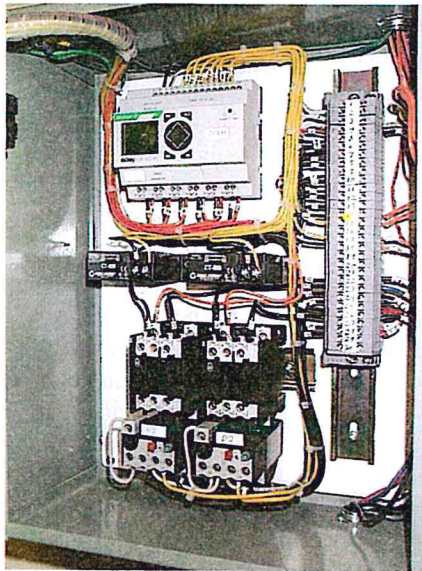
**Figure 1-1** Programmable logic controller.

Source: (a-b) Courtesy GE Intelligent Platforms.





(a)



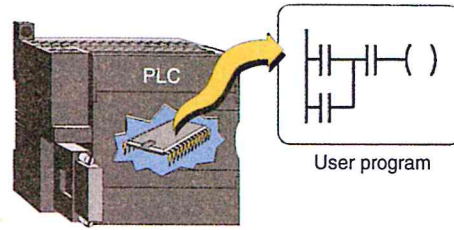
(b)

**Figure 1-2** Relay- and PLC-based control panels. (a) Relay-based control panel. (b) PLC-based control panel.

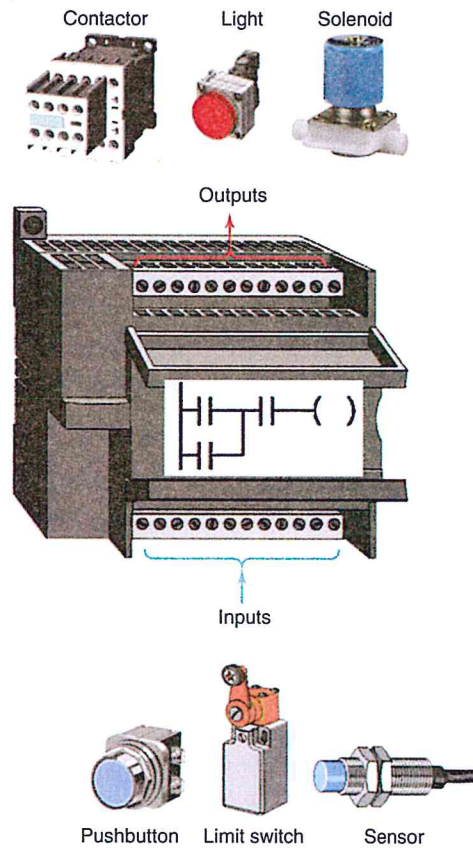
Source: (a) Courtesy Mid-Illini Technical Group, Inc.; (b) Photo courtesy Ramco Electric, Ltd.

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 (Figure 1-5).



**Figure 1-3** All the logic is contained in the PLC's memory.



**Figure 1-4** Relationships between the inputs and outputs are determined by the user program.

- **Faster Response Time.** PLCs are designed for high-speed and real-time applications (Figure 1-6). 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

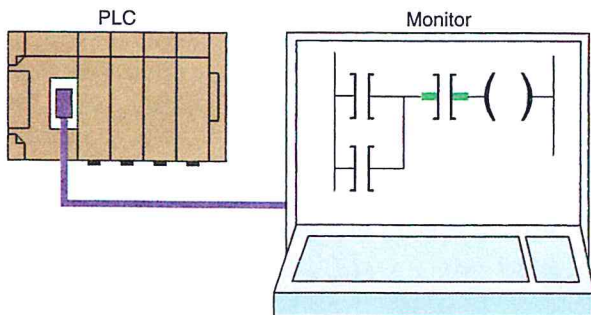




**Figure 1-5** PLC communication module.  
Source: Photo courtesy Automation Direct, [www.automationdirect.com](http://www.automationdirect.com).



**Figure 1-6** High-speed counting.  
Source: Courtesy Banner Engineering Corp.



**Figure 1-7** Control program can be displayed on a monitor in real time.

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 1-7).

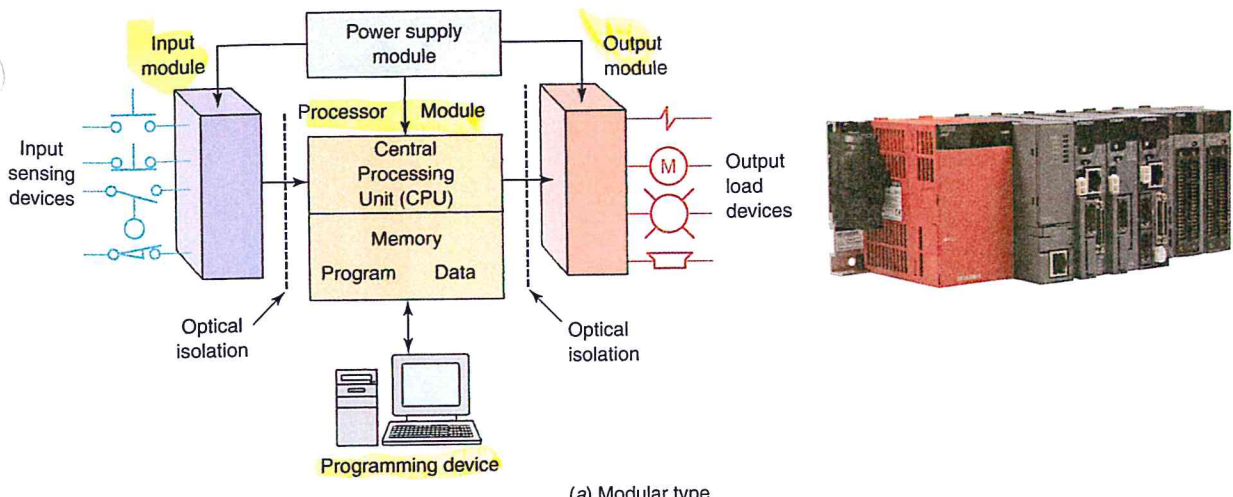
## 1.2 Parts of a PLC

A typical PLC can be divided into parts, as illustrated in Figure 1-8. 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 might be slight differences in addressing, memory allocation, retrieval, and data handling for different models. Consequently, PLC programs cannot be interchanged among different PLC manufacturers.

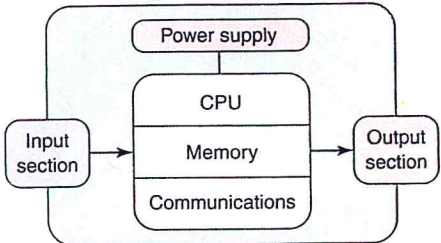
There are two ways in which I/Os (Inputs/Outputs) are incorporated into the PLC: fixed and modular. *Fixed I/O* (Figure 1-9) is typical of small PLCs that come in one package with no separate, removable units. The processor and I/O are packaged together, and the I/O terminals will have a fixed number 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.

*Modular I/O* (Figure 1-10) 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. The PLC processor is also connected to the backplane and can communicate with all the modules in the rack.

The *power supply* supplies DC power to other modules that plug into the rack (Figure 1-11). For large PLC systems, this power supply does not normally supply power to the *field devices*. With larger systems, power to field devices is



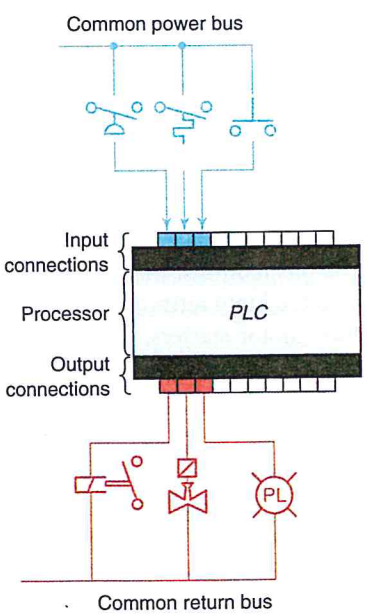
(a) Modular type



(b) Fixed type



**Figure 1-8** Typical parts of a programmable logic controller.  
Source: (a) Courtesy Mitsubishi Automation; (b) Image Used with Permission of Rockwell Automation, Inc.



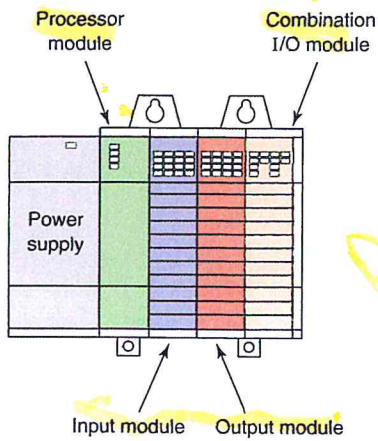
**Figure 1-9** Fixed I/O configuration.

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.

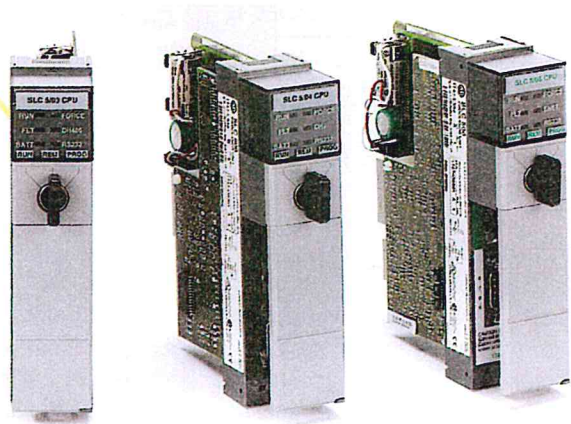
The *processor* (CPU) is the “brain” of the PLC. A typical processor (Figure 1-12) usually consists of a microprocessor for implementing the logic and controlling the communications among the modules. The processor requires memory for storing the results of the logical operations performed by the microprocessor. Memory is also required for the program EPROM or EEPROM plus RAM.

The CPU controls all PLC activity and is designed so that the user can enter the desired program in relay ladder logic. The PLC program is executed as part of a repetitive process referred to as a scan (Figure 1-13). A typical PLC scan starts with the CPU reading the status of inputs. Then, the application program is executed. Once the program execution is completed, the CPU performs internal diagnostic and communication tasks. Next, the status of all outputs is updated. This process is repeated continuously as long as the PLC is in the run mode.

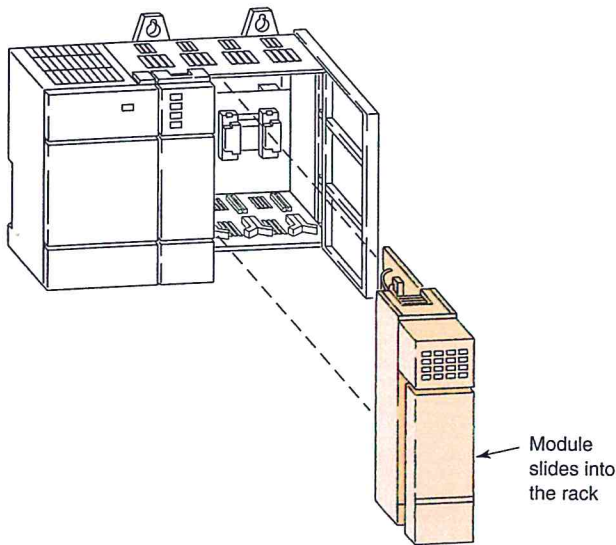




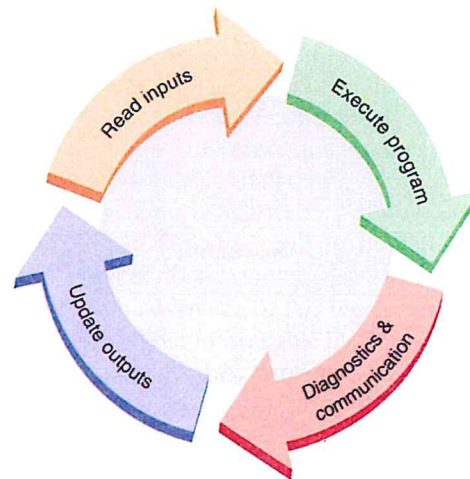
ALAN BRADLEY



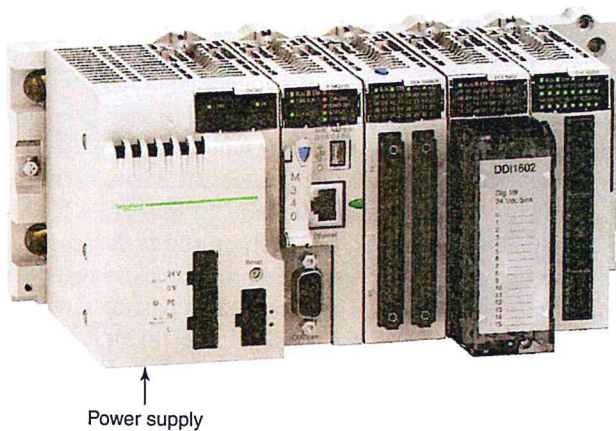
**Figure 1-12** Typical PLC processor modules.  
Source: Image Used with Permission of Rockwell Automation, Inc.



**Figure 1-10** Modular I/O configuration.



**Figure 1-13** Typical PLC scan cycle.



**Figure 1-11** The power supply supplies DC power to other modules that plug into the rack.

Source: This material and associated copyrights are proprietary to, and used with the permission of Schneider Electric.

The I/O system forms the interface by which field devices are connected to the controller (Figure 1-14). 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 duplicates the function of relays, timers, and counters.



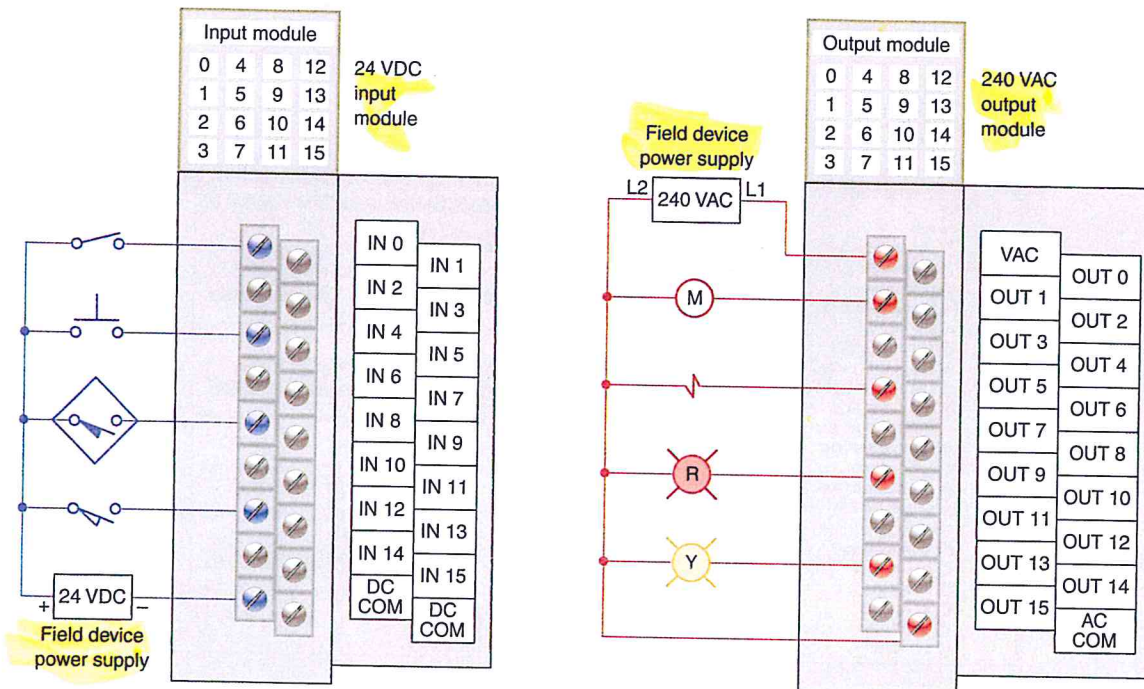


Figure 1-14 Typical PLC input/output (I/O) system connections.

A *programming device* is used to enter the desired program into the memory of the processor. The program can be entered using relay ladder logic, which is one of the most popular programming languages. Instead of words, ladder logic programming language uses graphic symbols that show their intended outcome. A program in ladder logic is similar to a schematic for a relay control circuit. It is a special language written to make it easy for people familiar with relay logic control to program the PLC. Hand-held programming devices (Figure 1-15) are sometimes used to program small PLCs because they are inexpensive and easy to use. Once plugged into the PLC, they can be used to enter and monitor programs. Both compact hand-held units and laptop computers are frequently used on the factory floor for troubleshooting equipment, modifying programs, and transferring programs to multiple machines.

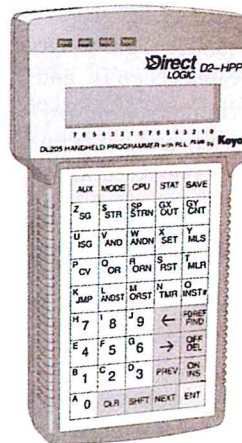


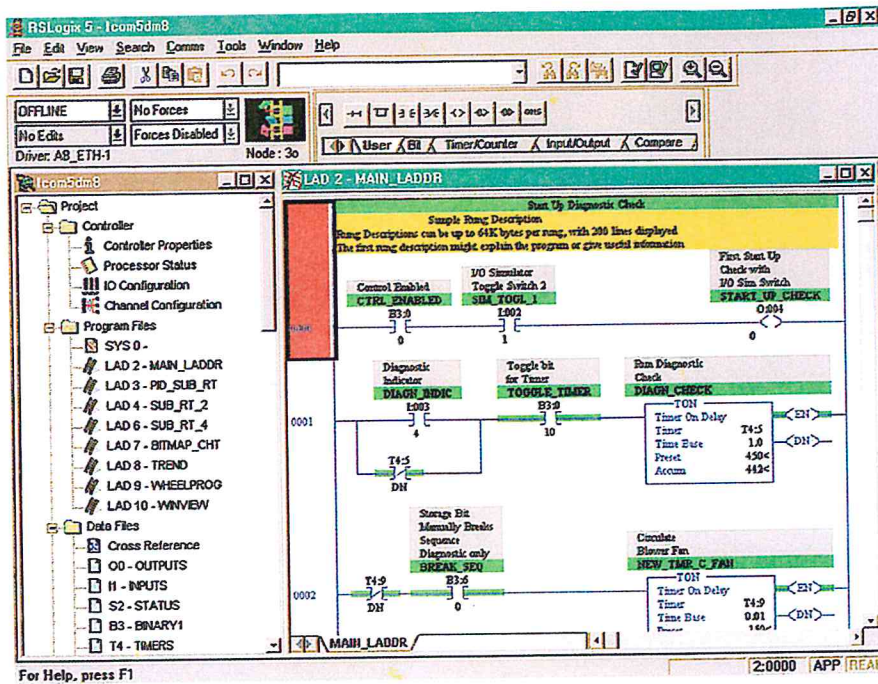
Figure 1-15 Typical hand-held programming device. Source: Photo courtesy Automation Direct, [www.automationdirect.com](http://www.automationdirect.com).

A personal computer (PC) is the most commonly used programming device. Most brands of PLCs have software available so that a PC can be used as the programming device. This software allows users to create, edit, document, store, and troubleshoot ladder logic programs (Figure 1-16). 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 personal computer communicates with the PLC processor via a serial or parallel data communications link, or Ethernet. If

the programming unit is not in use, it may be unplugged and removed. Removing the programming unit will not affect the operation of the user program.

A *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





**Figure 1-16** Typical PC software used to create a ladder logic program.  
Source: Image Used with Permission of Rockwell Automation, Inc.

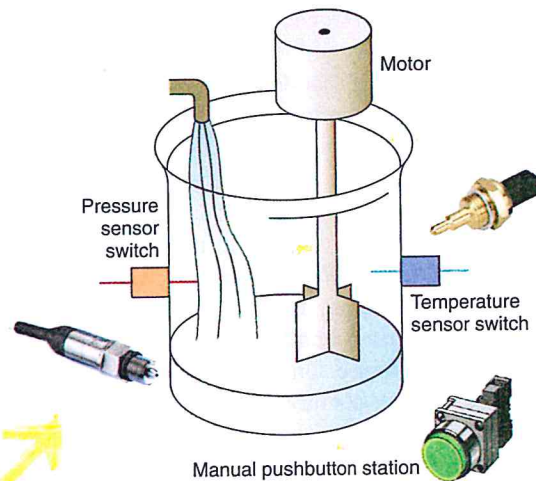
with PLCs. Its origin is based on electromechanical relay control. The relay ladder logic program graphically represents rungs of contacts, coils, and special instruction blocks. RLL was originally designed for easy use and understanding for its users and has been modified to keep up with the increasing 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 1-17. Here a mixer motor is to be used to automatically stir the liquid in a vat when the temperature and pressure reach 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 1-18. 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. These devices would be hardwired to an



**Figure 1-17** Mixer process control problem.

appropriate input module according to the manufacturer's addressing location scheme. Typical wiring connections for a 120 VAC modular configured input module is shown in Figure 1-19.

The same output field device (motor starter coil) would also be used. This device would be hardwired to an appropriate output module according to the manufacturer's addressing location scheme. Typical wiring connections for a 120 VAC modular configured output module is shown in Figure 1-20.



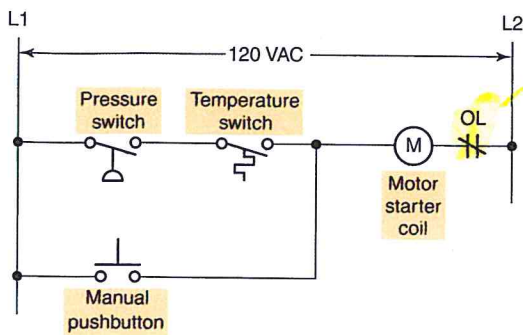


Figure 1-18 Process control relay ladder diagram.

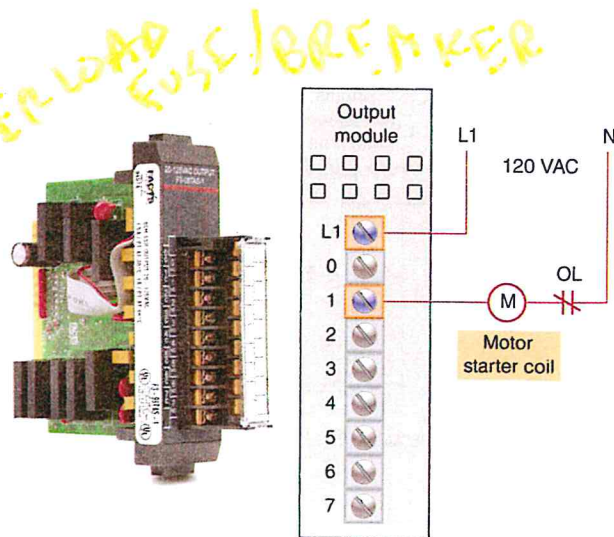


Figure 1-20 Typical wiring connections for a 120 VAC modular configured output module.

Source: Photo courtesy Automation Direct, [www.automationdirect.com](http://www.automationdirect.com).

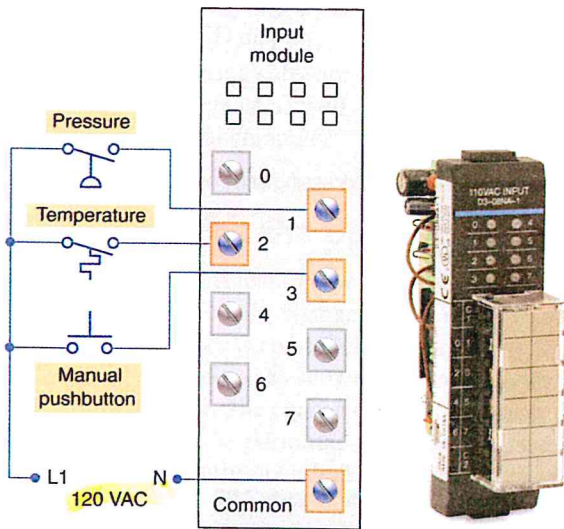


Figure 1-19 Typical wiring connections for a 120 VAC modular configured input module.

Source: Photo courtesy Automation Direct, [www.automationdirect.com](http://www.automationdirect.com).

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 1-21. 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 location addresses. To program the controller, you enter these instructions one by one into the processor memory from the programming 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 each operating cycle, the controller examines the status of input devices, executes the user program, and changes outputs accordingly. Each  $\parallel$  symbol can be thought of as a set of normally open contacts. The  $()$  symbol is considered to represent a coil that, when energized, will close a set of contacts. In the ladder logic program of Figure 1-21, 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 scheme 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



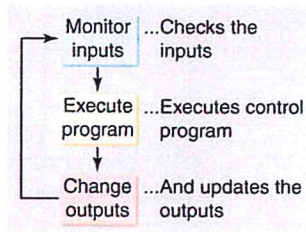
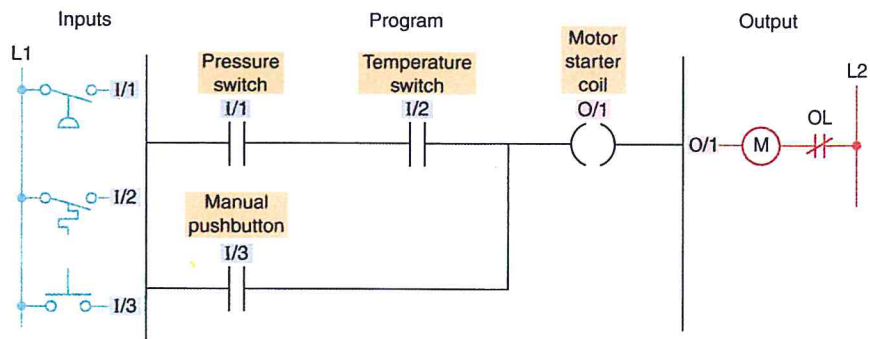


Figure 1-21 Process control PLC ladder logic program with typical addressing scheme.

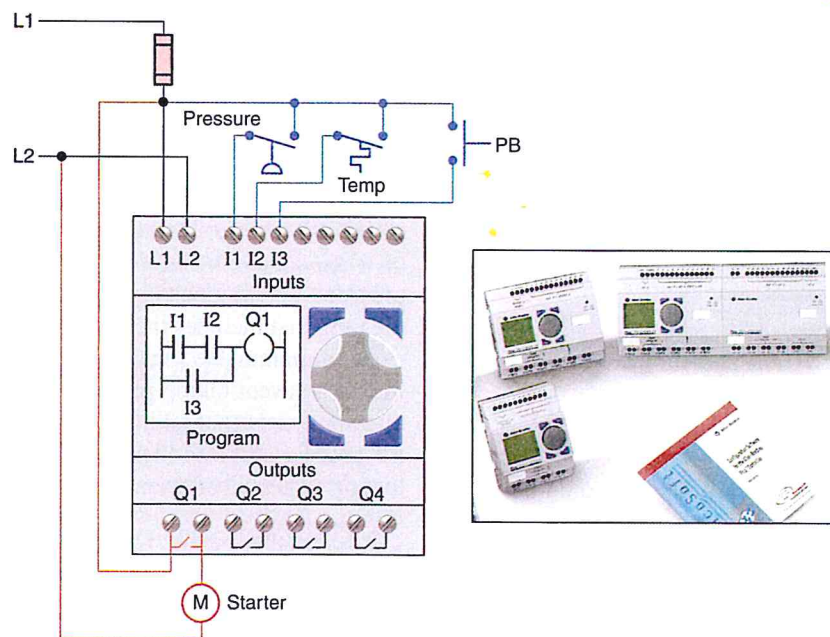


Figure 1-22 Typical wiring required to implement the process control scheme using a fixed PLC controller.

Source: Image Used with Permission of Rockwell Automation, Inc.

- 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 1-22 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 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 1-23. 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 1-23 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 1-24.

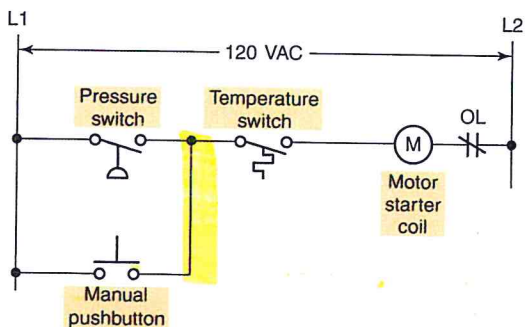


Figure 1-23 Relay ladder diagram for the modified process.

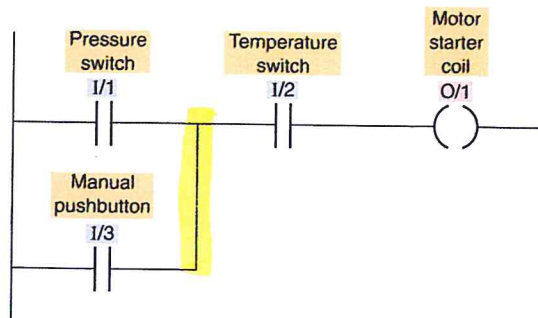


Figure 1-24 PLC ladder logic program for the modified process.

### 1.5 PLCs versus Computers

The architecture of a PLC is basically the same as that of a personal computer. A personal computer (PC) can be made to operate as a programmable logic controller if you provide some way for the computer to receive information from devices such as pushbuttons or switches. You also need a program to process the inputs and decide the means of turning load devices off and on.

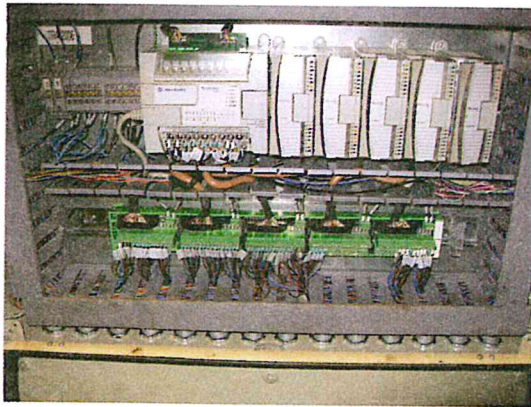
However, some important characteristics distinguish PLCs from personal computers. First, unlike PCs, the PLC is designed to operate in the industrial environment with wide ranges of ambient temperature and humidity. A well-designed industrial PLC installation, such as that shown in Figure 1-25, is not usually affected by the electrical noise inherent in most industrial locations.

Unlike the personal computer, the PLC is programmed in relay ladder logic or other easily learned languages. The PLC comes with its program language built into its memory and has no permanently attached keyboard, CD drive, or monitor. Instead, PLCs come equipped with terminals for input and output field devices as well as communication ports.

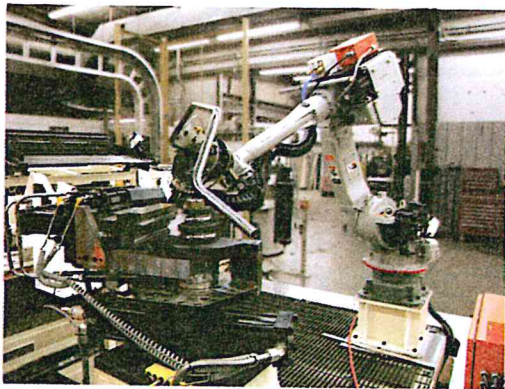
Computers are complex computing machines capable of executing several programs or tasks simultaneously and in any order. Most PLCs, on the other hand, execute a single program in an orderly and sequential fashion from first to last instruction.

PLC control systems have been designed to be easily installed and maintained. Troubleshooting is simplified by the use of fault indicators and messaging displayed on the programmer screen. Input/output modules for connecting the field devices are easily connected and replaced.





(a)



(b)

**Figure 1-25** PLC installed in an industrial environment.  
Source: (a–b) Courtesy Automation IG.

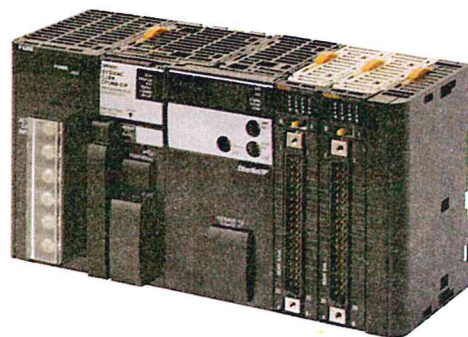
Software associated with a PLC but written and run on a personal computer falls into the following two broad categories:

- PLC software that allows the user to program and document gives the user the tools to write a PLC program—using ladder logic or another programming language—and document or explain the program in as much detail as is necessary.
- PLC software that allows the user to monitor and control the process is also called a *human machine interface (HMI)*. It enables the user to view a process—or a graphical representation of a process—on a monitor, determine how the system is running, trend values, and receive alarm conditions (Figure 1-26). Many operator interfaces do not use PLC software. PLCs can be integrated with HMIs but the same software does not program both devices.

Most recently automation manufacturers have responded to the increased requirements of industrial control systems



**Figure 1-26** PLC operator interface and monitor.  
Source: Courtesy Rogers Machinery Company, Inc.



**Figure 1-27** Programmable automation controller (PAC).  
Source: Photo courtesy Omron Industrial Automation, [www.ia.omron.com](http://www.ia.omron.com).

by blending the advantages of PLC-style control with that of PC-based systems. Such a device has been termed a programmable automation controller, or PAC (Figure 1-27). Programmable automation controllers combine PLC ruggedness with PC functionality. Using PACs, you can build advanced systems incorporating software capabilities such as advanced control, communication, data logging, and signal processing with rugged hardware performing logic, motion, process control, and vision.

## 1.6 PLC Size and Application

The criteria used in categorizing PLCs include functionality, number of inputs and outputs, cost, and physical size (Figure 1-28). Of these, the *I/O count* is the most important factor. In general, the nano is the smallest size with less than 15 I/O points. This is followed by micro types (15 to 128 I/O points), medium types (128 to 512 I/O points), and large types (over 512 I/O points).

Matching the PLC with the application is a key factor in the selection process. In general it is not advisable to





**Figure 1-28** Typical range of sizes of programmable controllers.

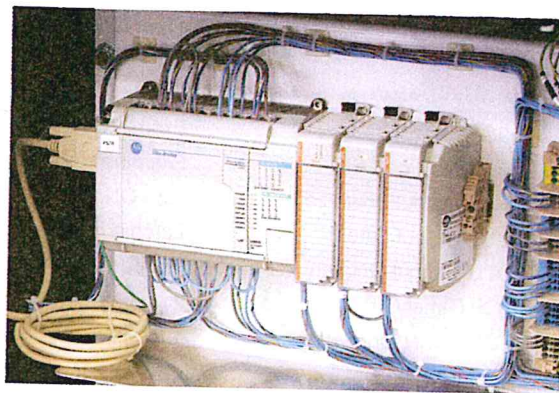
Source: Courtesy Siemens.

buy a PLC system that is larger than current needs dictate. However, future conditions should be anticipated to ensure that the system is the proper size to fill the current and possibly future requirements of an application.

There are three major types of PLC application: single-ended, multitask, and control management. A **single-ended** or **stand-alone** PLC application involves one PLC controlling one process (Figure 1-29). This would be a stand-alone unit and would not be used for communicating with other computers or PLCs. The size and sophistication of the process being controlled are obvious factors in determining which PLC to select. The applications could dictate a large processor, but usually this category requires a small PLC.

A **multitask** PLC application involves one PLC controlling several processes. Adequate I/O capacity is a significant factor in this type of installation. In addition, if the PLC would be a subsystem of a larger process and would have to communicate with a central PLC or computer, provisions for data communications network are also required.

A **control management** PLC application involves one PLC controlling several others (Figure 1-30). This kind



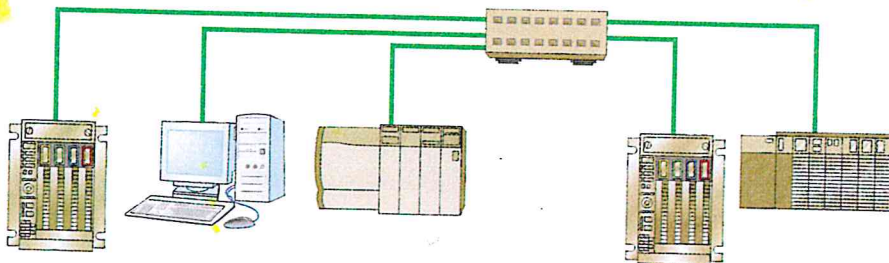
**Figure 1-29** Single-ended PLC application.

Source: Courtesy Rogers Machinery Company, Inc.

of application requires a large PLC processor designed to communicate with other PLCs and possibly with a computer. The control management PLC supervises several PLCs by downloading programs that tell the other PLCs what has to be done. It must be capable of connection to all the PLCs so that by proper addressing it can communicate with any one it wishes to.

**Memory** is the part of a PLC that stores data, instructions, and the control program. Memory size is usually expressed in K values: 1 K, 6 K, 12 K, and so on. The measurement kilo, abbreviated K, normally refers to 1000 units. When dealing with computer or PLC memory, however, 1 K means 1024, because this measurement is based on the binary number system ( $2^{10} \nabla 1024$ ). Depending on memory type, 1 K can mean 1024 bits, 1024 bytes, or 1024 words.

Although it is common for us to measure the memory capacity of PLCs in words, we need to know the number of bits in each word before memory size can be accurately compared. Modern computers usually have a word size of 16, 32, or 64 bits. For example, a PLC that uses 8-bit words has 49,152 bits of storage with a 6 K word capacity ( $8 \times 6 \times 1024 = 49,152$ ), whereas a PLC using 32-bit words has 196,608 bits of storage with the same 6 K memory ( $32 \times 6 \times 1024 = 196,608$ ). The amount



**Figure 1-30** Control management PLC application.



**Table 1-1 Typical PLC Instructions**

Instruction	Operation
XIC (Examine ON) . . . . .	Examine a bit for an ON condition
XIO (Examine OFF) . . . . .	Examine a bit for an OFF condition
OTE (Output Energize). . . . .	Turn ON a bit (nonretentive)
OTL (Output Latch) . . . . .	Latch a bit (retentive)
OTU (Output Unlatch) . . . . .	Unlatch a bit (retentive)
TOF (Timer Off-Delay) . . . . .	Turn an output ON or OFF after its rung has been OFF for a preset time interval
TON (Timer On-Delay) . . . . .	Turn an output ON or OFF after its rung has been ON for a preset time interval
CTD (Count Down) . . . . .	Use a software counter to count down from a specified value
CTU (Count Up) . . . . .	Use a software counter to count up to a specified value

of memory required depends on the application. Factors affecting the memory size needed for a particular PLC installation include:

- Number of I/O points used
- Size of control program
- Data-collecting requirements

- Supervisory functions required
- Future expansion

The *instruction set* for a particular PLC lists the different types of instructions supported. Typically, this ranges from 15 instructions on smaller units up to 100 instructions on larger, more powerful units (see Table 1-1).