5.4 PLCs: Programmable Logic Controllers


Reviewed by B. G. Lipták (1995)

Types of Input/Output (I/O): Discrete I/O: 120 VAC, 220 VAC, 0 to 5 V DC, 0 to 24 V DC, transistor-transistor logic (TTL)
Analog I/O: 4 to 20 mA, 1 to 5 V DC, 0 to 10 V DC, –5 to +5 V DC, –10 to +10 V DC
Special I/O: Thermocouple, RTD, stepper motor (pulses), strain gauge, high-speed counters, PID

Typical Specifications: Scan Time per 1000 Words (1 K) of Logic: 0.1 to 50 ms, depending on manufacturer and enhanced software features
Word Size: 4, 8, 16, or 32 bit (typical)
Amount of I/O: Nano PLC—under 15 I/O; Small or Micro PLC—15 to 128 I/O; Medium PLC—128 to 512 I/O; Large PLC—512 and greater I/O
Size of Memory: Small PLC—256 to 2 K words (K = 1024 bits, bytes, or words of digital data); medium PLC—2 K to 12 K words; large PLC—12 K words and larger
Type of Memory: CMOS (complementary metal oxide semiconductor) RAM (random access memory); BBRAM (battery-backed RAM); EEPROM (electrically erasable PROM); and Flash memory
Environmental Conditions: 0 to 60°C (32 to 140°F), relative humidity, to 95% noncondensing, 115 VAC ± 15% and 230 VAC ± 15%

Costs: Nano PLC Hardware: Basic PLC with CPU, 12 I/O points costs $99 to $200
Small PLC Hardware: Basic PLC with CPU, 2 K to 8 K RAM memory, 20 to 40 I/O points costs $200 to $1000 (extra discrete I/O when available costs $10/point)
Medium PLC Hardware: Basic system with 256 I/O points costs $5,000 to $8,000; with 512 I/O points costs $10,000 to $20,000; and with 1024 points costs $14,000 to $28,000 (RAM quantity adjusted to handle designated I/O count)
Large PLC Hardware: $15,000 to $70,000 (cost driven by network and information display requirements)
Software and Engineering: Software costs range from 50 to 100% of the hardware cost. System engineering costs and documentation costs each range from 25 to 50% of hardware cost. Therefore, the total cost (without installation labor) is about twice (if not more) the hardware cost.

Support Equipment Cost: Handheld programmers cost $100 to $500; PC-supported programming software costs $2500 to $5000

Partial List of PLC Suppliers:
ABB (Elsag-Bailey Controls) (www.ABB.com)
Allen-Bradley/Rockwell Automation (www.AB.com)
Automation Direct (www.Automationdirect.com)
Danaher (Eagle Signal Controls) (www.Dancon.com)
Emerson (Westinghouse) (www.EmersonProcess.com)
Fuji Electric Corp. (www.FujiElectric.com)
G.E. Fanuc Automation (www.GEIndustrial.com)
Giddings & Lewis (www.GLControls.com)
Idec Corp. (www.Idec.com)
5.4 PLCs: Programmable Logic Controllers

International Parallel Machines Inc. (www.ipmi plc.com)
Mitsubishi Electric (www.meau.com)
Modicon/Schneider Electric (www.Modicon.com)
Moeller Corp. (www.Moeller.net)
Omega Engineering (www.Omega.com)
Omron Electronics Inc. (www.Omron.com)
Reliance Electric Co./Rockwell Automation (www.Reliance.com)
Siemens (www.sea.siemens.com)
Toshiba Inc. (www.Toshiba.com)
Triconex/Invensys (www.Triconex.com)
Uticor Technology Inc. (www.Uticor.com)

(Note: The most popular PLCs are Allen-Bradley, Modicon, Siemens, and GE Fanuc.)

Partial List of HMI
Software Suppliers:
GE Fanuc (iFix & Cimplicity) (www.gefanucautomation.com)
Rockwell Software (RSView) (www.software.rockwell.com)
Invensys (Wonderware InTouch) (www.wonderware.com)

INTRODUCTION

A programmable logic controller (PLC) is an industrially hard-
ened computer-based unit that performs discrete or continuous
control functions in a variety of processing plant and factory
environments. Originally intended as relay replacement equip-
ment for the automotive industry, the PLC is now used in vir-
tually every industry imaginable. Though they were commonly
referred to as PCs before 1980, PLC became the accepted abbre-
viation for programmable logic controllers, as the term “PC”
became synonymous with personal computers in recent decades.

PLCs are produced and sold worldwide as stand-alone
equipment by several major control equipment manufacturers.
In addition, a variety of more specialized companies produce
PLCs for original equipment manufacturer (OEM) applications.

Typically, PLC vendors can supply large volumes of application notes for their products. Most major PLC vendors
also publish detailed articles about applications in technical
journals and prepare papers for engineering societies and
industrial symposia on control, automation, and so forth. Each
manufacturer’s software package usually has its own applica-
tion programming techniques. Vendors also are a valuable
source of “how-to” information, providing training courses in
their local office or at the factory as well as actual hands-on
experience to help users gain familiarity with the PLC. Most
vendors offer an applications or programming manual that
provides insight on how to use available programming fea-
tures. Of course, familiarity with one brand of PLC generally
helps the engineer learn to use other brands quickly.

HISTORY

The automotive industry fostered the development of the PLC
primarily because of the massive rewiring required each time a
model change occurred. Solid-state logic is much easier to
change than relay panels, and this advantage was reflected in the
cost of installing and operating PLCs instead of traditional relay
systems. Table 5.4a lists some of the milestones in the develop-
ment of PLCs. Table 5.4b provides a summary of the differences
among various technologies that perform logic control.

Bedford Associates, the forerunner of Modicon, designed
the first PLC in 1968 for the Hydramatic Division of General
Motors Corporation to eliminate costly scrapping of assembly-
line relays during model changeovers and to replace unreliable
electromechanical relays. The objectives of the program were to:

• Extend the advantages of static circuits to 90% of the
machines in the plant
• Reduce machine downtime related to controls problems
• Provide for future expansion
• Include full logic capabilities, except for data reduction
functions

Richard Morley, a Bedford engineer, is credited with
the original PLC design and the creation of ladder logic
programming. Morley says the diagrams on which ladder
logic is based, however, probably originated in Germany years
before to describe relay circuitry. Describing the technology
of the day, Morley explains, “Automatic control of industrial
metalworking and assembly operations in 1969 were mostly
by relays and clock-driven electromechanical devices. Relays
ran hot and tended to beat themselves to death with their
persistent 60-cycle hum and contact arcings. Electromechan-
ical timers/sequencers were much like enlarged music box
mechanisms and were maintenance headaches.” Describing
the early years of the PLC, Morley recounts, “We had some
real problems in the early days of convincing people that a
box of software, albeit cast in iron, could do the same
thing as 50 feet of cabinets, associated relays, and wiring.”

Morley recounted that in 1969, “all computers required a
clean, air-conditioned environment, yet were still prone to
frequent malfunctions. … Thus, even though PLCs were and
are special, dedicated computers, considerable effort was made
to not identify PLCs as computers due to the poor reliability
of computers and the fact that they were not things procured
TABLE 5.4a

History of Programmable Logic Controller (PLCs)

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1968</td>
<td>PLC designed for General Motors Corporation to eliminate costly scrapping of assembly-line relays during model changeovers.</td>
</tr>
<tr>
<td>1969</td>
<td>First commercial PLCs manufactured for automotive industry as electronic equivalent of relays.</td>
</tr>
<tr>
<td>1971</td>
<td>First application of PLCs outside the automotive industry.</td>
</tr>
<tr>
<td>1972</td>
<td>Allen-Bradley (AB) adds the first computer interface to its PLC, read-write controller, and off-line software documentation package.</td>
</tr>
<tr>
<td>1973</td>
<td>Introduction of “smart” PLCs for arithmetic operations, printer control, data move, matrix operations, CRT interface, etc.</td>
</tr>
<tr>
<td>1974</td>
<td>First CRT-based program panel for PLC.</td>
</tr>
<tr>
<td>1975</td>
<td>Introduction of analog proportional, integral, derivative (PID) control, which made possible the accessing of thermocouples, pressure sensors, etc.</td>
</tr>
<tr>
<td>1976</td>
<td>Allen-Bradley coins the PLC acronym.</td>
</tr>
<tr>
<td>1977</td>
<td>First use of PLCs in hierarchical configurations as part of an integrated manufacturing system.</td>
</tr>
<tr>
<td>1978</td>
<td>Introduction of PLCs based on microprocessor technology.</td>
</tr>
<tr>
<td>1979</td>
<td>PLCs gain wide acceptance; sales approach $80 million.</td>
</tr>
<tr>
<td>1980</td>
<td>IBM introduces PC/XT using DOS.</td>
</tr>
<tr>
<td>1981</td>
<td>Data highways enable users to interconnect many PLCs up to 15,000 feet from each other. More 16-bit PLCs become available. Color graphics CRTs are available from several suppliers.</td>
</tr>
<tr>
<td>1982</td>
<td>Larger PLCs with up to 8192 I/O become available.</td>
</tr>
<tr>
<td>1983</td>
<td>“Third-party” peripherals, including graphics CRTs, operators’ interfaces, “smart” I/O networks, panel displays, and documentation packages, become available from many sources.</td>
</tr>
<tr>
<td>1985</td>
<td>IBM-PC programming terminal introduced.</td>
</tr>
<tr>
<td>1986</td>
<td>Various other PC HMI software introduced.</td>
</tr>
<tr>
<td>1987</td>
<td>Wonderware introduces InTouch, the first MS Windows-based HMI application, and starts a major transition to Windows-based “open” technology.</td>
</tr>
<tr>
<td>1988</td>
<td>Small (Brick) PLCs enter the market place.</td>
</tr>
<tr>
<td>1989</td>
<td>Networked block I/O developed.</td>
</tr>
<tr>
<td>1990</td>
<td>Moore Products manufactures APACS, the first control system using IEC 1131-3 standard allowing DCS function blocks, PLC ladder, sequential function chart, and structured text to be used and combined within a single controller.</td>
</tr>
<tr>
<td>1991</td>
<td>Ethernet and TCP/IP connectivity appear on PLCs.</td>
</tr>
<tr>
<td>1992</td>
<td>AB launches DeviceNet, an open device-level network.</td>
</tr>
</tbody>
</table>

by manufacturing operations.”4 Unlike computers of that era, the programmable controller was designed to be reliable. Morley explains, “No fans were used, and outside air was not allowed to enter the system for fear of contamination and corrosion. Mentally, we had imagined the programmable controller being underneath a truck, in the open, and being driven around—driven around in Texas, driven around in Alaska. Under those circumstances, we wanted it to survive. The other requirement was that it stood on a pole, helping run a utility or a microwave station which was not climate controlled, and not serviced at all. Under those circumstances, would it work for the years that it was intended to be? Could it be walled in? Could it be bolted in a system that was expected to last 20 years?” Upon arriving for a client demonstration, the PLC was accidentally dropped on the floor. The client was surprised that not only did the PLC work perfectly, but that Modicon staff expected it to work after being dropped.5

Bedford Associates demonstrated the Modicon 084 solid-state sequential logic solver at GM in 1969.6 The fundamental advantage that PLCs introduced was that they used programming rather than rewiring to configure for a new application. PLC programming could be done much quicker than the traditional rewiring methods. In addition, solid-state devices offered greater reliability, required less maintenance, and had a longer life than mechanical relays, all in a smaller footprint.7,8

The 1970s and 1980s were dominated by proprietary systems and software. Odo Struger, often called the father of Allen-Bradley’s PLC, is credited with the PLC acronym.5 The early
1980s saw a cross pollination between PLCs and distributed control systems (DCSs). PLCs, for example, had already begun incorporating distributed control functions so they could be linked much in the way that DCSs were linked. Building on the trend, software companies sprang up in great numbers during this time.2

During the 90s, standardization and open systems were the main themes. Ethernet peer-to-peer networking became available from virtually all PLC manufacturers. EEPROM and Flash memories replaced the EPROMs of the 1980s. PCs and CRTs in general became accepted and started to replace switches and lights on control system panels. Small PLCs called “Bricks” were introduced to the marketplace. Redundancy for PLCs became a standard product. Many PLC systems were upgraded to address Y2K concerns. The first few years of the 21st century have seen a consolidation of PLC manufacturers. Very small nano or pico PLCs, some as small as industrial relays, have appeared. Safety PLCs featuring triple redundancy were introduced. LCD base operator interface panels have largely displaced CRTs, especially on the plant floor.

PLC SIZES

Today’s PLC is at best a distant relative of the first- and second-generation PLCs built during the 1970s and 1980s. There are now many PLC sizes to select from, ranging from nano- and micro-size devices with 12–30 I/O, to large supervisory control units with built-in PC and networking capabilities.7

The modern medium-sized PLC performs all the relay replacement functions expected of it but also adds many other functions—including counting, timing, and complex mathematical applications—to its repertoire. Most medium-sized PLCs can perform proportional, integral, derivative (PID), feed-forward, and other control functions as well (see Chapter 2). In addition, medium-sized and large-scale PLCs have data highway capabilities and can function well in DCS environments.

Nano PLCs

In the 1990s, small PLCs were introduced with limited functionality. Today’s small PLCs generally contain all of the software and networking power that used to be confined to the larger units.8 A small PLC can be amazingly inexpensive. This small, dedicated controller is enclosed in a single-mounted hardened case. It is intended to be a relay replacement unit and provide reliable control to a stand-alone section of a process. Small PLCs offer many of the same functions as larger models but with limited flexibility of adding I/O points. They are especially suited to limited applications with only basic communications requirements. “For instance, a $99 PLC with four PID loops and two serial ports, matched with a $79 four channel analog plug-in option card, is a cost-effective choice over traditional relays and a single-loop controllers.”9

TABLE 5.4b

<table>
<thead>
<tr>
<th>Cost Advantages over Relay</th>
<th>Relays</th>
<th>Solid-State Controls</th>
<th>Microprocessor</th>
<th>Minicomputer</th>
<th>PLCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware cost</td>
<td>Low</td>
<td>Equal</td>
<td>Low</td>
<td>High</td>
<td>High to low, depending on number of controls</td>
</tr>
<tr>
<td>Versatility</td>
<td>Low</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Usability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Troubleshooting maintainability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Computer-compatible</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Arithmetic capability</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Information gathering</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Industrial environment</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Programming cost</td>
<td>(Wiring) High</td>
<td>(Wiring) High</td>
<td>Very high</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Reusable</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Space required</td>
<td>Largest</td>
<td>Large</td>
<td>Small</td>
<td>OK</td>
<td>Small</td>
</tr>
</tbody>
</table>

BASIC PLC COMPONENTS

A PLC manufactured by virtually any company has several common functional parts. Figure 5.4c illustrates a generic PLC architecture. The diagram shows a central processor, memory, I/O, power supply, and programming and peripheral device subsections. Each is discussed below.
Redundancy The new breed of data highways can provide a process with a “hot” backup PLC to take over in the event of a PLC processor failure. Figure 5.4d depicts a PLC system with redundant CPUs and I/O.

Until recently, it was difficult or impossible to configure an ordinary PLC into a redundant hot backup system. Siemens was one of the first PLC makers to introduce redundant configurations for PLCs. For some people, redundancy is very important. Why is redundancy important? Safety in areas containing explosives or flammables, unattended operation, and environmental liabilities lead the list of concerns addressed in part by redundancy.\textsuperscript{10}

Safety PLCs For several years, Triconex was one of a few manufacturers that offered a triplicate PLCs for use in safety systems. Fault tolerant systems called programmable safety systems (PSSs) are now offered by a number of PLC manufacturers. “The big difference between a regular PLC and a PSS is that all the triple redundant features needed in both hardware and software are built into the latter. That can be done with regular PLCs but it’s a significant engineering challenge.”\textsuperscript{11} See Section 5.8 for further information on safety PLC systems.

It’s important to acknowledge that having redundant CPUs doesn’t necessarily provide system-wide redundancy. Nor does it protect against all types of errors. A programming error, such as divide by zero, or an infinite loop will stop a redundant CPU just as fast as a nonredundant CPU.

Memory Unit The memory unit\textsuperscript{12} of the PLC serves several functions. It is the library where the application program is stored. It is also where the PLC’s executive program is stored. An executive program functions as the operating system of the PLC, which serves as the program that interprets, manages, and executes the user’s application program. Finally, the memory unit is the part of the PLC where process data from the input modules and control data for the output modules are temporarily stored as data tables. This includes I/O status bits, counter values, timer preset and accumulated values, and other stored constants or variables. Typically, an image of these data tables is used by the CPU and, when appropriate, sent to the output modules.

The basic PLC memory element is the word. A word is a collection of 4, 8, 16, or 32 bits that is used to transfer data about the PLC. As word length increases, more information can be stored in a memory location. Even with the ambiguity associated with word length, PLCs that provide the equivalent of 32 K of 8-bit memory locations can execute application programs that are moderately complicated and interact with 50–100 discrete I/O points.

Memory can be volatile or nonvolatile. Volatile memory is erased if power is removed. Obviously, this is undesirable, and most units with volatile memory provide battery backup to ensure there will be no loss of program in the event of a power outage. This is often referred to as battery-backed
RAM. When batteries are used, they must be replaced on a regular basis, typically 1–2 years. Some small PLCs use a large capacitor instead of a battery to avoid this maintenance issue.

Nonvolatile memory does not change state on loss of power and is used in cases in which extended power outages or long transportation times to job sites (after program entry) are anticipated. Core memory was widely used in the 60s and early 70s but is no longer used. In the late 70s a solid-state nonvolatile memory called programmable read-only memory (PROM) became popular. The user programmed the PROM by applying a high current pulse in order to fuse internal links to set individual bits to 1 or 0. This procedure was irreversible. This required a separate piece of equipment called a PROM burner. Then erasable PROMs became available. Like the PROMs, EPROMs could be programmed by the user. The advantage was that EPROMs could be erased optically, through exposure to ultraviolet light through a quartz window on the chip, and then rewritten electrically. A drawback to EPROM devices was that they had to be removed from the equipment to be reprogrammed.

**EEPROM and Flash Memory** New memory technologies, electrically erasable (or Alterable) PROMs (EEPROMs, E’PROMs, or EAPROMs) and Flash memories, became available in the 90s and have essentially replaced the earlier types. EEPROM and Flash memories can be electrically reprogrammed by the user, but without removal from the system, and without the use of exterior programming devices such as PROM programmers. Flash is also used for the firmware used by PLCs and intelligent modules. Revision changes to

---

**FIG. 5.4d**
*Hot backup PLC networking. (Courtesy of Allen-Bradley.)*
the firmware can be “Flash upgraded” in the field, without equipment disassembly. EEPROM and Flash memory are the current choices for nonvolatile memory.

**I/O Systems**

The I/O group includes all types of modules, from dry contact modules to intelligent I/O to remote I/O capabilities. Inputs are defined as real-world electrical signals that give the controller real-time status of process variables such as level, pressure, flow, temperature, weight, analysis results, position, or status (hand switches, pushbuttons, alarm contacts, and so on). These signals can be analog or digital, low or high frequency, maintained or momentary. Typically, they are presented to the PLC as a varying voltage, current, or resistance value.

Based on the status sensed or values measured, the PLC controls various output modules to operate devices such as valves, motors, pumps, and alarms. PLC manufacturers are providing more and more types of input and output capabilities in their products. There are, however, many third-party peripherals that aid the PLC in interfacing to the field devices.

One of the most important functions of the I/O is its ability to filter, condition, and isolate real-world signals (0–120 VAC, 0–24 V DC, 4–20 mA, 0–10 V, and thermocouples) and convert these to the low signal levels (typically 0–5 V DC max) in the PLC I/O bus. This is accomplished by use of optical isolators. A light-emitting diode (LED) generates a light that optically turns on a phototransistor. Because the only thing normally connecting the input to the output is light, there is an excellent isolation between input and output voltages. Only the electrical breakdown of the isolator (typically over 1000 V) limits the voltage isolation achieved with these devices. Typical discrete I/O schematics are shown in Figures 5.4e, 5.4f, and 5.4g.

Most I/O systems are modular in nature; that is, a system can be arranged by use of modules that contain multiples of I/O points. These modules can be composed of 1, 4, 8, 16, or 32 points and plug into the existing bus structure. The bus structure is really a high-speed multiplexer that carries information back and forth between the I/O modules and the central processor unit. Higher point densities are possible, but their selection may involve a trade-off in wire size used, as well as the ease of wire harness installation to the module.

In small PLCs, the CPU and I/O are generally contained in a single enclosure and may or may not be expandable. These have been nicknamed “Bricks” because the size of the enclosure is similar to the size of a brick. In most medium and large PLCs, plug-in I/O modules are used to convert the I/O signal level to one that is compatible with the bus architecture. These modules can be composed of 1, 4, 8, 16, or 32 points, depending on the manufacturer’s standard design. For small projects (20–256 I/O), I/O requirements are usually easy to define and group. A systematic approach is required for medium-sized projects (256–1024 I/O), however, in order to avoid confusion of I/O allocation. Obviously, the organization of I/O for large systems (1024 I/O and above) requires careful planning.

PLC manufacturers are typically very good at backward compatibility. When new models of CPUs are introduced,
Multiple processors in a single rack. Some manufacturers allow expansion racks that are close by (usually within the same structure to “clean up” the panel design. As each new generation of I/O becomes increasingly dense, the systems become progressively harder to wire. Interposing terminal blocks between the field connections and the rack may become standard in the future. Some manufacturers already offer prewired field terminals that mount in a panel and are equipped with a cable that plugs directly onto the I/O module.

Digital Inputs Pushbuttons, limit switches, or even electromechanical relay contacts are familiar examples of digital, contact closure type signals. Input modules are typically transistor-triggered and have built-in time delays to protect against contact bounce. The input signal from a field device (limit switch) has to be energized for some amount of time in order for the module to notify the processor of a true “on” condition. The modules are available in a number of voltages including DC voltages (14, 15–30, 24, 120, 240), TTL (5 V DC), and AC voltages (24, 48, 120, 240). They come in 8, 16, or 32 points per module.

Discrete I/O modules come equipped with an LED indicator light to indicate the status of each I/O point in the module (on or off) for troubleshooting. The input module LED indicates field side status of the pushbutton.

Modules, especially high-density modules, often share a common neutral between multiple inputs. This means that the input voltages must be referenced to the same neutral. AC signals powered from different power sources cannot be mixed within the shared group. Using a different module for the different sources or using isolating relays solves the problem.

Digital Outputs Digital or discrete outputs (DOs) can be used to power pilot lights, solenoid valves, or annunciator windows (lamp box). The discrete output module uses a solid-state switch (triac) to power a field device, such as a motor starter, a valve, or lights. Outputs are available for voltage ranges of 5–240 V at currents up to 5 A, with typical 120 V outputs operating at 1–2 A maximum. Solid-state drivers of this type are not intended to drive large loads directly (e.g., a large motor starter). Highly inductive loads or those with a high surge current may also require an interposing dry contact relay in order to power the field device or a resistor-capacitor (RC) network to control transients.
Both manufacturers and third-party vendors offer dry contact modules. These modules solve the problems normally associated with triacs: low power and uncertainty of failure state. Triac outputs also "leak" some current when they are off. When attached to high impedance devices such as Sonalerts or LED pilot lights, this leakage current can cause continuous or intermittent operation. A voltage-dropping resistor can be added across the load to solve this, but the resistors are often large and run hot. Some designers simply add inexpensive isolation relays as needed to address this problem.

Manufacturers rate their equipment output current at different temperatures. All current ratings for solid-state outputs will vary with ambient temperatures. Therefore, one should be sure to check the PLC output rating for each application and manufacturer. Built-in fusing on output modules is becoming standard in the industry and provides good protection for overload conditions. The type of fuse depends on the module, and the way in which fuses are accessed varies from one PLC manufacturer to another. Some wiring arms have individual fuses for each output.

Discrete I/O modules come equipped with an LED indicator light to indicate the status of each output on the module (on or off) for troubleshooting. The output module LED generally indicates logic side status.

Analog Inputs Analog modules are sometimes referred to as A/D (analog-to-digital) modules. They provide optical isolation for electrical noise protection and are typically arranged in a quad module or an eight-point module. The analog I/O system is designed to interface with analog field devices such as flowmeters, pressure transmitters, and valve positioners. This system accepts inputs of 0–5 V DC, 0–10 V DC, –5 to +5 V DC, –10 to +10 V DC, 0–20 mA, or 4–20 mA. Precisions of 1 in 4,096 (12-bit), 1 in 32,768 (15-bit), or 1 in 65,536 (16-bit) are commonly available.

Direct thermocouple and RTD input modules typically accept eight to ten points each. On-board cold junction compensation is often included with thermocouple input modules, and many different types of thermocouples (B, C, E, J, K, N, R, S, T), RTD (100 Ω platinum, 120 Ω nickel, 10 Ω copper), and mV signals (0–50 mV, 0–100 mV) can be accommodated.

Analog Outputs Analog outputs can drive signals to variable-speed drives, control valves, and other analog devices. The output analog module can output a signal of 4–20 mA current loop as well as 0–10 V DC, 0–5 V DC, –5 to +5 V DC, and –10 to +10 V DC. Most source their power from the back plane. Some loops require external loop power supplies.

Register I/O Systems One additional type of input signal, the register input, reflects the computerized nature of the PLC. The register input is particularly useful when the process condition is represented by a collection of digital signals delivered to the PLC at the same time. A binary-coded decimal (BCD) thumbwheel is a good example of an input device that is compatible with a register input port. If the thumbwheel represents three and one-half digits of process data, then all 14 data output wires from the thumbwheel would provide their digital signal directly to the PLC register input unit, which would, in turn, signal the condition and transfer the data to the central processor unit.

The register I/O system provides direct interface to multi-bit data field devices, such as thumbwheel switches, position encoders, and digital readouts to the PLC. These devices are typically TTL level, which allows interface to other types of electronic hardware as well. Intelligent I/O and other special-purpose I/O requirements are becoming increasingly common.

Specialty Modules Specialty modules are designed to solve a single interface problem. X/Y positioner modules can be included in this category, as well as servo axis controllers, stepper motor outputs, and even maintenance access modules. These modules are a further extension of the distributed technology. Combination I/O modules combine either digital or analog inputs and outputs on a single module. They are particularly useful in small systems.

High-Speed Modules Fast-response I/O is currently offered in both discrete and analog versions. Discrete rapid response modules are facilitated by the PLC logic, but the output does not rely on ladder logic scan times to get updated. High-speed analog modules provide quicker analog-to-digital and digital-to-analog (D/A) conversions. This gives PLCs the ability to control faster PID loops and to make analog measurements of assembly-line parts (weight, for example). High-speed pulse counter modules provide the ability to interface with turbine meters, stepper motors, and optical encoders. High-speed pulses cannot normally be interfaced to PLC inputs because of the scan time of the ladder logic. These modules provide an interface that does not rely on the scan time, so the PLC is able to monitor fast pulses that indicate position or flow. They can often be configured as high-speed up counters, up/down counters for quadrature use, high-speed interrupts, pulse catchers for very fast pulses, and adjustable filter inputs. PID loop coprocessor and temperature controller modules are also available.

Intelligent I/O Modules Intelligent I/O modules include all modules that are able to perform processing functions. Because the tasks performed by the PLC are further distributed, greater speed and reliability for the overall system can be realized. Intelligent I/O modules give the PLC multiple additional capabilities, which may include memory storage and retrieval, computing tasks, and communications. Memory modules provide additional room to store data points, alarm messages, lookup tables, and the like. This approach leaves the main operating memory free for the control tasks. Computing modules give PLCs the ability to perform true computer functions using a language like Basic. Basic modules can be used for a variety of things, such as complex math abilities.
Because they usually come with two or more serial ports, they can be used for handling specialized ASCII protocols and they can demultiplex a multiplexer data stream. Of course they can be programmed as Modbus masters or slaves. Again, the real-time tasks are left in the main memory, but tasks such as setpoint calculation, formation of data, and some operator interface tasks may be placed in the computer module. Other in-rack modules include complete computers, state language coprocessors, and real-time interrupt modules.

Clock modules that fit into the I/O bus may be considered part of this group. These modules provide real-time and day/date functions upon interrogation from the PLC. Most are backed up by a battery to ensure time keeping during power outages.

I/O Simulators Some I/O simulators used to develop and debug programs can be categorized in the I/O enhancement group. These specific devices are typically hardware modules that can be plugged into the PLC. I/O simulators generally look like input cards except switches are built into the module for testing.

PLC Power Supply

The power supply may be integral or separately mounted. It always provides the isolation necessary to protect solid-state components from most high-voltage line spikes. The power supply converts power line voltages to those required by the solid-state components. All PLC manufacturers provide the option to specify line voltage conditions (typically 120 VAC, 240 VAC, or 12/24 V DC). In addition, the power supply is rated for heat dissipation requirements for plant floor operation. This dissipation capability allows PLCs to have high-ambient-temperature specifications and represents an important difference between PLCs and PCs for industrial applications.

The power supply drives the I/O logic signals, the central processor unit, the memory unit, and possibly some peripheral devices. As I/O is expanded, some PLCs may require additional power supplies in order to maintain proper power levels. The power supply generally does not provide power to field devices, though some manufacturers provide external terminals to permit this use. The additional power supplies may also be separate or part of the I/O structure. The power requirements of all the I/O modules and CPU or remote I/O adapters are added together for each voltage used to make sure that the power supply is large enough.

ADDITIONAL PLC COMPONENTS

Communications Modules

Communications modules can provide the PLC with a range of capabilities, from simple ASCII output strings to communications networking. The storage of ASCII messages for a printer or display can be contained outside the main memory of the PLC, and the data can be output when required. Full-system communications networking capabilities are provided with network modules, giving the designer the ability to multiplex PLCs off a single operator interface device or a supervisory computer. A wide variety of devices communicate using ASCII, including weight scales, bar code scanners, some operator interfaces, and modems. Modules included in this category also include telephone and radio modem modules. Remote I/O and peer-to-peer are specialized communications modules that are discussed below.

Remote I/O

Robust industrial networks make distributed I/O feasible. Networks enable I/O modules to be located close to field devices, resulting in labor and material savings for wiring. Often manufacturers provide a remote I/O distribution panel or module to serve the efficient multiplexing of the modules on the remote I/O rack back to the CPU. A remote I/O base is similar to the local I/O base except it holds a remote I/O adapter (or has it built-in) instead of the CPU. Most medium-sized PLCs can support several remote racks, which in turn contain 4, 8, or 16 I/O modules. It may or may not be possible to mix various modules in a remote rack. Specific information about module compatibility and remote I/O multiplexing is available from the manufacturer. This information is required to facilitate PLC selection and sizing for specific applications.

Whereas in most systems the module has the intelligence to communicate with the CPU, some systems require the use of serial interface modules. In any case, some provision is made to accept register input data from the input modules and to send this data (on or off status of field device) in serial format to the PLC processor. Serial data are also converted into register data to be sent to the output module. This is normally a proprietary protocol.

Remote I/O adapters need to be configured for a particular drop, either by programming or a switch setting on the module itself. Remote I/O is very often an RS-485-based system. The end modules in the RIO network are usually terminated with a resistor that matches the impedance of the cable used. RIO communications speeds range from 56 K to 230 Kbps. Remote I/O is broken down into two distinct types: the integral type, which allows a limited transmission distance (up to 15,000 ft, or 4,500 m); and the transmitter/receiver type, which allows virtually unlimited transmission capability. Most PLC manufacturers and third-party peripherals manufacturers can provide some form of either type. Technology is advancing rapidly in this area, as systems change from fiber optics to microwave and radio transmission. Many manufacturers are now offering flexible remote I/O solutions. Often these are modular, plug-together systems instead of being rack based. Designed to be located next to the wireway, they incorporate built-in terminal blocks for direct wiring.
Open vs. Proprietary I/O Networks

Proprietary networks have a compelling advantage: They are easy to put together. If you buy all your network components from one supplier, even if they’re for an open network like DeviceNet, you avoid potential configuration problems. Even so, offering both open and proprietary systems is important to most manufacturers.

Besides proprietary remote I/O networks, a variety of open protocols exist to enable I/O data interchange. These include Modbus, DeviceNet, Profibus, HART, and Fieldbus, among others. Because these are open, a number of different manufacturers offer products based on them. Ethernet is becoming a major factor by providing a backbone that can carry the diverse protocols of nonproprietary I/O communications.

Peer-to-Peer Communications

Many PLC manufacturers have responded to the increasing industry demands for equipment that allows communications among multiple process areas. The original master control layout shown in Figure 5.4h is adequate for PLC control of moderate-size applications of perhaps 100–500 I/O points, but as the application grows the PLC becomes overburdened under this arrangement. The hierarchical plan illustrated in Figure 5.4i provides a supervisory PLC controlling network that reduces this burden. The plan allows for a “master” PLC that oversees the process and controls a set of “slave” PLCs that control the actual process activities in the plant.

An alternate approach is illustrated in Figure 5.4j. This distributed control system allows dedicated PLCs to control sections of the process with an interface to management. As indicated in the figure, this interface could be to a human operator. This person would monitor selected data from the collection of PLCs. Any management decisions generated from this data are passed back to the PLCs by an input terminal. Under normal conditions it is expected that each dedicated PLC can monitor and control its section of the process.

FIG. 5.4h
With master control, one PLC controls a number of related machines and processes. This system is simple but may require long runs of multiple-wire cables, and the entire system is vulnerable to the failure of the one PLC. (Courtesy of Allen-Bradley.)

FIG. 5.4i
Hierarchical control. (Courtesy of Allen-Bradley.)

Most major PLC manufacturers currently offer their own network communications for their products (e.g., DatHighway+, ControlNet, Modbus+, and TI-WAY). Various network protocols exist, and efforts to generate standards are gaining momentum. Instrument and control organizations such as the Instrumentation, Systems, and Automation (ISA) Society constantly exert pressure on the PLC industry to agree on one or at least a few communications standards. In any event, there are several communications philosophies of interest. Some of these are outlined below.

FIG. 5.4j
Distributed control. (Courtesy of Allen-Bradley.)
The computer interface device group is a rapidly expanding section of PLC peripheral devices. These devices allow peer-to-peer communications (i.e., one PLC connected directly to another), as well as network interaction with various computer systems. In fact, this group of devices will certainly expand in number as communications standards become commonly accepted and more and more products are provided to facilitate such network interactions.

Most PLC manufacturers have already addressed the need for peer-to-peer communications among PLCs on a distributed network. Without this feature, every time one PLC needs to know the status of another part of the machine or plant, it must interrupt the activities of the supervisory computer in order to get the information (Figure 5.4k) or interface through status I/O points. The ability of one PLC to “talk” to another along the data highway greatly speeds the control activities of each machine and allows the supervisory computer to “concentrate” on its tasks.

Some PLC manufacturers and third parties are offering universal communications networking based on open protocols such as Modbus. This is likely to be the favored method in the future, because the need for networking different brands of PLCs together will increase.

More PLC vendors are adding Ethernet compatibility to their systems. Ethernet has become the open architecture of choice for a variety of industries. However, care must be taken to ensure that the PLC is never connected to the Internet. Doing so opens the door to all sorts of security issues, especially hackers. Using a PC in combination with firewalls, encryption, and passwords helps address some of these issues.

**Peripheral Devices**

The popularity of PLCs has led to the creation of a strong third-party peripheral manufacturing industry. These companies are always developing new products that assist the PLC user with interfacing a specific application to a PLC. Two categories of these products — operator stations and programming/documentation tools — are presented below. The operator stations facilitate operator interface with the PLC-controlled process to monitor process variables, to alter program parameters, to conduct on-line program alterations, and to conduct troubleshooting procedures. Programming and documentation tools include products supplied by the manufacturers or made available by third-party vendors.

**Local Operator Interface**

Operational aids include a variety of resources that range from switches and lights, to color graphics CRTs, to equipment or support programs that can give the operator specific...
access to processor parameters. In this situation the operator is usually allowed to read and modify timer, counter, and loop parameters but not access the program itself. Some aids facilitate the interaction between the PLC and printers to deliver process information in a desired format. Some devices have the ability to set up an entire panel and plug into the PLC through external RS-232-C ports, saving enormous panel and wiring costs.

The most basic of operator interfaces are switches and lights. These are directly wired to the input and output cards and relay information back and forth to the PLC. Analog signals can also be used with potentiometers, panel meters, and chart recorders. Individual loop controllers can be wired to accept a remote set point from the PLC. Register I/O is used with BCD thumbwheels and BCD displays. Annunciator panels can also be used. Although just about all control panels have some switches and lights, the cost of using a display has fallen to such an extent that it is often not cost-effective to use switches and lights as the primary interface. PLC backup systems that are used if the PLC is being serviced still use these simple devices.

Annunciator panels are still used in some industries. They are light boxes that have alarm descriptions illuminated when an output is activated. The PLC generally supplies one output point per alarm. The lamp test, silence, and acknowledge functions are built into the annunciator assembly. Text display units are also available in a number of sizes, which display a preprogrammed message when signaled by the PLC. Other message systems are available that can dial a preprogrammed phone number and deliver a page, a text message, or a verbal description of the triggering event. Some switches and light assemblies became available in standard arrangements that connected directly using remote I/O technology. Some are available that communicate using the Modbus protocol.

Operator stations include those provided by manufacturers intended to be used with their particular PLC and those offered by third parties for use with either a particular brand or anyone’s PLC. These stations may include devices such as timer/counter access modules (TCAMs), loop access modules (LAMs), data terminals, color graphics consoles, computers, printers, and manual backup stations. Most PLC manufacturers provide an operator interface unit (OIU) designed specifically for their PLC. These are either part of the standard system or offered as an option. They are sometimes mounted directly on the PLC or can be panel-mounted and cabled back to the controller. Functions include access to read/write register data, simple programming, and diagnostics.

Some specialized devices, such as TCAMs, LAMs, and OIUs, provide operator interaction with PLC internal registers and loop tables. This gives the systems designer the ability to provide real-time changing of variables, loop tuning and inspection, manual control of analog outputs, and the ability to provide batch- or menu-type information at low cost. Communications with the PLC are multidropped over an RS-422 or a similar differential line. Unauthorized data entry is prevented with software locks, keylock protection, or both. TCAMs and LAMs have become less popular as the cost of graphic interface panels has dropped, because their functions are easily added to these newer interface panels. Some PLCs can support communications directly with dumb data terminals. However, dumb terminals have been replaced almost completely by PCs.

Many PLCs are able to provide communications directly to printers. A stand-alone PLC system can often provide performance reports, alarm logging, and the like without ever involving a computer. This feature is usually somewhat limited, because PLCs were designed primarily to control the process machine. Large amounts of data, sophisticated print logs, and multiple alarms are not really within the realm of a stand-alone PLC system. This type of data manipulation is too cumbersome and requires too much memory for most PLCs. In addition, many PLCs are located in the process floor, which is generally not a good environment for printers.

Small touch screen panels have become very popular in recent years. They provide basic reporting of status information and commands in a small format. Some also provide built-in alarm management functions. They are meant to replace the traditional switches, lights, and meters. The main disadvantage these small screens have is that they are fundamentally different from the computers in the control room. The screen graphics for the local panels must be developed separately. They generally are limited to the individual control panel they are associated with.

With the advent of plant floor Ethernet and relatively inexpensive computers, it has become practical to provide the same screens at the local control panel that are displayed in the control room. An industrial computer or a regular desktop computer in a sealed computer enclosure is used. The same software that is used in the control room is loaded onto the local computer, which uses the local area network (LAN) to communicate with the supervisory control and data acquisition (SCADA) server. This has many advantages. Because the screens are shown on the same type of hardware, they can be simply copied from the control room computers. Obviously, this helps reduce development and maintenance costs. Because these communicate with the entire plant, an alarm can be investigated at any control panel throughout the plant.

**Human-Machine Interface**

Intelligent human-machine interfaces (HMIs) can be networked into a total distributed system to give a redundant or local interface to the system (Figure 5.41). This technique can be used to provide the control room interface, while the supervisory computer supports its own interfaces. Another powerful technique is to allow the networking of a few PLCs together solely for the purpose of providing a single human-machine interface to all parts of the system (Figure 5.4m). This solution is ideal for small PLC users because it is very economical yet allows expansion as the plant grows.
Color graphics consoles offer process graphics and communications facilities simultaneously to many brands of PLCs. Because of this ability, HMIs are sometimes used to tie multiple PLC systems together. These systems range from those that can simply be purchased and put on-line with a minimum of engineering effort to those that require some programming. The basic differences are in flexibility. Those that do not require programming may not be able to provide the custom menus and graphics that are required. The ease of communications with different types of PLCs also varies according to manufacturer. Finally, the method of generating the graphics pages differs greatly. Most color graphics consoles offer multiple graphics pages that are animated by reading data tables in the PLCs. Operators enter the data by means of standard keyboards, user-configurable industrial keyboards, light pens, touch screens, and the like. Different graphics pages may be selected with preformatted menus or custom menus programmed by the user or the systems house. Development stations are often required to give the final user the ability to change graphics menus or key commands after the initial project is completed.

Computer systems can be made to perform human-machine interface functions. Indeed, the color graphics consoles described in the previous paragraph are simply computers with standard graphics and communications software packages. Most PLC manufacturers provide board-level additions or modules that give the PLC the ability to converse via the RS-232 or Ethernet with nearly any computer. Of course, both the communications software and the particular applications software must be generated to provide an interface. Many vendors and systems houses are providing communications packages for various PLCs to run on microcomputers and PCs. These small systems offer low-cost operator interfaces to PLCs, providing data handling capabilities and the ability to be networked into a true distributed architecture. Microcomputers that have the ability to multitask, access large amounts of both RAM and nonvolatile memory, have proper software support, and are able to be networked will provide a good investment in terms of operator interface functions as well as total system capability.

**Printers**

Printers have always been an important part of the PLC system both as a development tool and for handling some of the operator interface functions. Two types of printers are commonly found in control rooms: dot matrix printers and laser printers. The dot matrix printers are used to print alarms and events as they occur. They are unique because as soon as a line of text is sent to the printer, the operator can read it. This is not possible with most types of printers currently popular. The laser printers are used for reports or screen printout and are either color or black and white.

**Programmers and Workstations**

The programmer unit provides an interface between the PLC and the user during program development, start-up, and troubleshooting. The instructions to be performed during each scan are coded and inserted into memory with the programmer.

Programmers vary from small handheld units the size of a large calculator to desktop stand-alone intelligent CRT-based units. These units come complete with documentation, reproduction, I/O status, and on-line and off-line programming ability. Most PLCs can use a PC as the programming tool. A program for the PC allows it to interface with a serial or Ethernet port in the PLC.

Programming units are the liaison between what the PLC understands (words) and what the engineer desires to occur during the control sequence. Some programmers have the ability to store programs on floppy disks or on a hard drive. Another desirable feature is automatic documentation of the existing program. This is accomplished by a printer attached to the programmer. With off-line programming, the user can write a control program on the programming unit, then take the unit to the PLC in the field and load the memory with the new program, all without removing the PLC. Selection
PLCs and Other Logic Devices

of these features depends on user requirements and budget. On-line programming allows cautious modification of the program while the PLC is controlling the process or the machine.

Manufacturers of PLCs provide two basic programming tools. These are handheld programmers and programming software that runs on PCs. The programming software may not be an option if a small PLC is purchased. Dedicated CRT programmers, once manufactured for the sole purpose of programming PLCs, have been replaced with PCs running PLC programming software.

The handheld programmer enables the operator to enter a program one contact at a time. These units are widely used because they are rugged, portable, and easy to operate. They are very cost-effective and give an engineer the capability to enter a program and to diagnose trouble in logic and field devices.

The programming software on a PC provides the engineer with a visual picture of the program in the PLC. Ladder diagrams are drawn on the screen, just as they would be drawn on paper. Design and troubleshooting time is reduced. With menu-driven software, programming training time is decreased. Information is stored on either floppy disks or the hard drive. Programs can be copied from one disk to another and then verified without need for loading the PLC’s memory. With stand-alone programming an engineer can develop a program on the PC and then load it into the PLC.

PLC programming software generally provides complete documentation capability, including ladder diagrams, cross-reference listing, I/O listing, user-defined contacts, and coil names along with commentary above each network or rung. All of this is can be sent to a printer for hard copy documentation. The screen typically shows eight rungs of ladder logic by 11 contacts across. The ladder diagrams can be placed into the real-time mode, which allows visual contact status. A whole screen of contacts and coils can be updated in as fast as a few seconds. Laptop PCs are often used for portability. With a modem connection, these programs can be used at remote locations for programming and troubleshooting.

PLC programming software may be restricted for use on only one PC. Security keys are one way to obtain this isolation. Security keys are devices that plug into the back of the computer or a special key disk. Without the key the software will not run. Some manufacturers use alternate security schemes such as providing extra keys at group prices or issuing site license agreements. In any case, it is unlikely that a PLC manufacturer will ever allow the use of programming software on an unlimited number of PCs.

Some PC-compatible software allows the PLC to be emulated by the PC. This software is sold by the PLC manufacturer or a licensee and is often model-specific. If the software also offers on-line programming and troubleshooting characteristics, it might be usable on only a single specific PLC. This restriction is achieved by means of software or hardware keys that come with each copy of the software purchased.

On-line engineering of PLC systems can be configured from a remote location like a control room with the combination of the programming/documentation tools and the distributed network. Systematic start-up and debugging of processes are available with this technique. Figure 5.4n depicts on-line engineering as part of a distributed system.

JUSTIFICATION FOR THE USE OF PLCs

For a given control problem, several technologies besides PLCs can be applied to achieve a solution: relays and stand-alone controllers, DCSs, and software running on PCs. PLCs differ from these technologies as described below.

PLCs vs. Relays and Stand-Alone Controllers

The main purpose of any control system is to get the process variable under control effectively and reliably. When control options are available, several factors can be taken into consideration in making an implementation decision. Some of these factors include the controller’s cost, versatility, flexibility, and expandability.

Cost  The ideas associated with cost are discussed first because this is usually the first issue users consider when selecting a control technology. In most cases, the two things initially known about a problem are the results desired and the budget available for fixing the problem. Unfortunately, at the beginning of a project the understanding of the problem is often limited. If the user’s focus on the problem is too narrow, the solution might solve only the perceived symptoms
of the problem while the problem itself may reappear in an alternate form at some other point in the process.

The budget assigned to solve a problem should be based not only on the initial investment required for the PLC hardware, but also on the costs associated with labor, maintenance, and downtime. Purchasing a larger and therefore more expensive PLC might in some situations be cost-effective when labor, maintenance, and downtime costs are also considered. Similarly, while PLCs may be more expensive than individual solid-state control units, if the indirect costs are also considered then the PLC becomes a cost-competitive alternative. As the cost of PLCs continues to drop, the cost of small PLCs may be equal to or even be less expensive than the cost of industrial relays.

Although significant, cost reductions alone are not the only reason that PLCs are the major replacement candidate for traditional relay logic. Compared with electromechanical relay systems, PLCs offer the following additional advantages:

- Ease of programming and reprogramming in the plant
- A programming language that is based on relay wiring symbols familiar to most plant electrical personnel
- High reliability and minimal maintenance
- Small physical size
- Ability to communicate with computer systems in the plant
- Moderate to low initial investment cost
- Rugged construction
- Modular design

**Versatility** The multifunction capability of a PLC allows control logic decision-making, enabling versatility rarely possible with other systems. The ability to combine discrete and analog logic is a powerful tool for the controls engineer. This is particularly evident in the control of batch processes. Entire start-up and shutdown sequences can be performed by the sequencer logic, and analog logic can be brought in during the batch run. Control of critical start-up parameters, such as temperature and pressure, can be precisely preprogrammed for each start-up step. Temperature stepping is easily programmed, as are the feedforward calculations that are used in some polymer reactors. All of these types of PLC applications are currently in use today and are well documented.

**Flexibility** As a process goes on-line and is refined, the control equipment should be easily reconfigured to accommodate such modifications. The multifunction use of the PLC has already been discussed. In addition, digital blending applications, boiler control of either carbon monoxide or excess oxygen, and some other forms of optimizing control are also within the capabilities of PLCs. Because one common device performs multiple functions in a plant, fewer spare parts are needed, and the programming language is technician-friendly. In addition, the digital nature and self-diagnostics capabilities are strong additional justification for the PLC.

**Expandability** As a process matures, it is inevitable that enhancements will be added. These usually require more inputs and outputs. For hand-wired relay systems this usually necessitates extensive panel changes, which generally are problematic. A PLC easily accommodates the additional I/O without requiring changes in the existing wiring. The new points are merely placed in the system. If a PID loop or two is being added, no panel rework is necessary; only the wiring of the new points and some reprogramming to incorporate them are required. Conversely, if the initially selected PLC is “tight,” additional I/O bases might be necessary. For this reason, most manufacturers recommend sizing the system to allow for 10 to 20% expansion.

Another advantage of the PLC is that it allows piecemeal implementation of projects. Systems can be brought on-line quickly and can be gradually converted to the PLC while online. The ability of the PLC to be reprogrammed while operating permits automation of processes that are too expensive to shut down. This technique is valuable to new as well as retrofit projects (revamps).

**PLC vs. DCS**

The capabilities of PLCs and DCSs have changed to the extent that today many applications that used to be the exclusive province of one or the other can be handled by both. The technology differences between PLCs and DCSs have evolved to be subtle or nonexistent. Manufacturers that had been making relays for logic and interlock applications developed PLCs, while DCS systems were developed by process control manufacturers with substantial experience in PID-type analog control. In the past it made good sense to use each type of controller in its area of superior experience. If the bulk of the I/O was digital (discrete), the logical choice was to use a PLC. If the I/O was mostly analog, a DCS system was selected. This logic is no longer universally true, and personal preference and end-user familiarity have become decisive factors in system selection. PLC and DCS comparisons have been taking place for as long as the two systems have existed. What’s odd about this debate is that PLCs define only the controllers, whereas DCSs also include the software, operator interface, and network.

In terms of pros and cons between the two designs, PLC I/O is likely to be more rugged and PLCs are likely to handle discrete logic faster than DCS systems. PLCs are also likely to be more desirable because their languages, such as ladder logic, are usually more familiar to plant personnel, and therefore there is less resistance to using them. On the other hand, ladder logic-type languages can be undesirable in some situations because they are not well suited to analog process control.

PLCs tend to be sold in a more piecemeal fashion as compared to the integrated systems approach common to DCSs. While PLCs are less expensive than DCSs, the end user often becomes more deeply involved with programming and systems integration issues. Nonetheless, many users find
PLCs to be a cost-effective solution for small to medium-sized process plants. Some users have overcome the limitations of PLCs by coupling them to PCs using custom-coded programming. The disadvantage of this approach is that such a nonstandardized system is usually understood fully only by its designer, and when that individual leaves the organization, the system can be ruined. When it comes to communications redundancy and data security, the DCS systems are superior. The DCS systems are also superior in their programming library, in advanced or optimizing control, in self-tuning algorithms, and, particularly, in their total plant architecture and information management capabilities.

**PLC vs. Personal Computers**

The PC is designed to be flexible and handle thousands of different types of applications, but PLCs are especially designed for control. PCs started showing up on the factory floor in the mid-80s in programming and HMI applications. However, now the PC is migrating toward actual control. In response, PLC manufacturers have adapted PC technology, such as Ethernet. Some systems actually provide a “PC on a card” to plug into the PLC back plane. PLCs are often thought of as computers. To a certain extent this is true; however, there are important differences between PLCs and computers.

**Real-Time Operation/Orientation**

The PLC is designed to operate in a real-time control environment. The first priority of the CPU is to scan the I/O for status, make sequential control decisions (as defined by the program), implement those decisions, and repeat this procedure all within the allotted scan time. Most PLCs have internal clocks and “watchdog timers” built into their operations to ensure that a software error like “divide by zero” or an endless loop does not send the central processor into an undefined state. When the watchdog time is exceeded, the processor shuts down in a predetermined manner and usually turns off all outputs. In real-time systems, reliability is a big concern. PLC manufacturers’ experience shows mean time between failures (MTBF) ranging from 20,000 to 400,000 hours. “This is far in excess of almost any other type of electronic or control equipment.”

**Environmental Considerations**

PLCs are designed to operate near the equipment they control. This means they function in hot, humid, dirty, noisy, and dusty industrial environments. Typical PLCs can operate in temperatures as high as 140°F (60°C) and as low as 32°F (0°C), with tolerable relative humidity ranging from 0 to 95% noncondensing. In addition, they have electrical noise immunities comparable with those required in military specifications.

**Programming Languages and Techniques**

PLC languages are designed to emulate the popular relay ladder diagram format. This format is read and understood worldwide by maintenance technicians as well as engineers. Unlike computer programming, PLC programming does not require extensive special training. Applications know-how is much more important. Although certain special techniques are important for programming efficiency, they are easily learned. The major goal is the control program performance. Another difference between computers and PLCs is the sequential operation of the PLC. Program operations are performed by the PLC in the order they were programmed (Figure 5.4o). This is an extremely useful feature that allows easy programming of shift registers, ring counters, drum timers, and other useful indexing techniques for real-time control applications.

**Maintenance and Troubleshooting**

As a plant floor controller, the plant electrician or the instrument technician must maintain the PLC. It would be highly impractical to require computer-type maintenance service. To this end, PLC manufacturers build in self-diagnostics to allow for easy troubleshooting and repair of problems. Most PLC components are modular and simple to isolate; remove-and-replace (system modules) diagnostic techniques are usually implemented.

**SUMMARY**

Table 5.4p together with Figures 5.4q and 5.4r provide illustrative summaries of the PLC characteristics discussed thus far. Table 5.4p illustrates an I/O list for a milling application, while Figure 5.4q shows in some detail the PLC and all its key parts. The CPU is shown in the center of the figure as the PLC processor. Power for this unit is delivered by the
power supply shown to its right. The programming unit connected directly to it on the left is the way a ladder logic program line, like the one shown just above the CPU, is entered into the controller.

Representations of two I/O modules are shown in Figure 5.4. The input module is on the left of the drawing and indicates a variety of contacts directly attached to seven of the eight points on the module. The point at address zero is shown as a spare for use as a replacement or future enhancement site. The output module shown has all eight of its output addresses (from address 140 through 147) in use.

Figure 5.4q shows an example of a remote I/O rack. Although the rack shown (rack No. 2) is powered by the main power supply, that is not a requirement. A remote I/O rack functions the same way as its direct I/O counterpart does. Various modules can be inserted in the rack to match the application’s control needs.

### Table 5.4
Example of a Detailed I/O List for a Milling Machine

<table>
<thead>
<tr>
<th>Input</th>
<th>Definition</th>
<th>Used in Rung(s)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>MOA Automatic</td>
<td>1–46</td>
</tr>
<tr>
<td>X1</td>
<td>Right Shot Pin In L.S. 1</td>
<td>3 – 3 9</td>
</tr>
<tr>
<td>X2</td>
<td>Left Shot Pin In L.S. 2</td>
<td>3 – 3 9</td>
</tr>
<tr>
<td>X3</td>
<td>Machine Slide Adv L.S. 3</td>
<td>7 – 22</td>
</tr>
<tr>
<td>X4</td>
<td>Machine Slide Ret L.S. 4</td>
<td>3 – 8 – 39</td>
</tr>
<tr>
<td>X5</td>
<td>Shot Pins Out L.S. 5 &amp; 6</td>
<td>6 – 52</td>
</tr>
<tr>
<td>X6</td>
<td>Pump #1 3M 1</td>
<td></td>
</tr>
<tr>
<td>X7</td>
<td>Right Bore Slide Ret L.S. 7</td>
<td>11 – 46</td>
</tr>
<tr>
<td>X8</td>
<td>Left Bore Slide Ret L.S. 8</td>
<td>12 – 47</td>
</tr>
<tr>
<td>X9</td>
<td>Right Bore Slide Adv L.S. 9</td>
<td>3 – 15 – 19</td>
</tr>
<tr>
<td>X10</td>
<td>Left Bore Slide Adv L.S. 10</td>
<td>3 – 15 – 21</td>
</tr>
<tr>
<td>X11</td>
<td>Swing Clamp On P.S. 1</td>
<td>6</td>
</tr>
<tr>
<td>X12</td>
<td>Pull Clamp On P.S. 2</td>
<td>5</td>
</tr>
<tr>
<td>X13</td>
<td>Swing Clamp Off P.S. 3</td>
<td>16</td>
</tr>
<tr>
<td>X15</td>
<td>Cycle Stop 1 2</td>
<td></td>
</tr>
<tr>
<td>X16</td>
<td>Auto Lube On/Off 40 48</td>
<td></td>
</tr>
<tr>
<td>X17</td>
<td>Lube Complete Pin – 40 – 48</td>
<td></td>
</tr>
<tr>
<td>X18</td>
<td>Hi Lube Level 37 – 40</td>
<td></td>
</tr>
<tr>
<td>X19</td>
<td>Lo Lube Level – 40 – 48</td>
<td></td>
</tr>
<tr>
<td>X20</td>
<td>Manual Light – 1 49 50 51 52</td>
<td></td>
</tr>
<tr>
<td>X21</td>
<td>Pull Clamp On/Off 49 – 52</td>
<td></td>
</tr>
<tr>
<td>X22</td>
<td>Shot Pins In/Out 50 – 52</td>
<td></td>
</tr>
<tr>
<td>X23</td>
<td>Swing Clamp On/Off 51 – 52</td>
<td></td>
</tr>
</tbody>
</table>

* Rung is a grouping of PLC instructions that controls one output or storage bit. This is represented as one section of a logic ladder diagram.

### PROJECT EXECUTION

Like any major enterprise, the PLC project must take into account the important considerations of schedule and budget. The PLC can facilitate the transition, however, by simultaneously pursuing several activities, thereby condensing the overall project schedule. A review of each major activity is presented in the following paragraphs.

### Systems Analysis

The control system should be analyzed as a whole to determine plant control requirements. The PLC plays an integral part in these analyses, and its capabilities should be thoroughly understood by the controls engineer. Vital to systems analysis are the process and instrument diagram (P&ID), the descriptive operational sequence, and a logic diagram.
or electrical schematic. Part of this evaluation will be system sizing and selection. Once the appropriate PLC is selected and purchase orders are placed, two activities should begin immediately: engineering design and software development.

**Systems Integrators**  One of the first decisions is whether to perform the work in-house or have a systems integrator do it. The general trend in the industry is that systems are more likely to be sold to OEMs and system integrators than to end users. If an integrator will be used, it should be selected carefully. Some things to consider include:

- How well does the integrator know the PLC/HMI/reporting software?
- How well does it know your industry?
- How familiar is the integrator with your operations?
- How well can you define the operation of your project?
- How good is its training?
- Will it have time and budget to fine-tune the system?
5.4 PLCs: Programmable Logic Controllers

- After warranty, will you be able to make changes without the integrator or will you have to pay for it to return?\(^{36}\)

For budgeting, whether the work is done in-house or by an integrator, software costs range from 50 to 100% of the hardware cost. System engineering costs and documentation costs each range from 25 to 50% of hardware cost. Therefore, the total cost (without installation labor) is about twice (if not more) the hardware cost.

**Open Systems**

Another consideration is whether to use open systems or proprietary. A major benefit of using open systems is that a number of different manufacturers can supply equipment for the system. “The trend today is definitely towards open systems and open languages, but the bulk of the PLC market is still proprietary—about 75% proprietary and 25% open or standards-based. Even so, a large number of proprietary PLCs are shipped with Profibus, DeviceNet, or other open networking capabilities.”\(^{37}\)

**Distributed Control** “One of the effects of better networks and intelligent devices is that control function no longer has to reside in a central controller. It can be scattered about, either in a network of small PLCs or in intelligent I/O bases… Some distributed processing is good to have because you can divide an operation into logical pieces, but it becomes difficult to partition programs and logic and so forth into the machines.”\(^{11}\) A potential benefit dividing a single large system into separate areas of control is that starting and troubleshooting a smaller piece of the system is more manageable.\(^{37}\)

**Redundancy** Another issue is the amount of redundancy that will be built into the control system. Some areas to consider when evaluating the need for redundancy include preventing production impacts, equipment damage, business interruption and quality impacts, liability costs, public health and safety, site safety, and environmental impacts.\(^{38}\) Redundant PLCs are available and I/O can be made redundant but it is often better to provide diversity in redundancy planning. Instead of using PLCs, manual controllers can be used. Manual control stations are important as backups in case of failure.
of the PLC controlling PID loops. Operator stations and HMI often provide manual control capabilities but still rely on the integrity of the PLC, so they are not truly manual in the hardwired sense. A manual control station is an important part of the distributed control system because it gives true manual control of the loops locally or in the control room, even when the local control systems are down. It is important to look at the overall system when evaluating redundancy. Unfortunately, the term “control-system redundancy” has come to refer to only the CPU, possibly the I/O but not the field devices. Even redundant PLCs can “fail” if they are powered from a single power source.

**PLC Hardware, System Sizing, and Selection**

Despite the variety of available PLC models, system sizing is relatively simple. Hardware and system size can be determined by an analysis of the following system characteristics:

1. I/O quantity and type
2. Remote I/O requirements
3. Memory quantity and type
4. Programming requirements
5. Programmers
6. Peripheral requirements

Although sizing is generally straightforward, selection of the right PLC requires considerable judgment regarding trade-offs between future requirements and present cost. The first three are examined below. The others are addressed elsewhere in this chapter.

**I/O Quantity and Type (Example 1)** For this simple system, the first step is development of the I/O list (Table 5.4p). This detailed document will be used extensively and should be developed with great care. (Once I/O numbers are assigned, it becomes very difficult to change all references to these numbers.) If switches and pilot lights are used to operate the controls, include I/O for them. Include any alarm outputs used for alarm annunciators. An intrusion switch on the panel door is useful for larger control panels. Also, consider allowing 20% spare I/O of each type used. The I/O list is followed by the configuration drawing.

The configuration drawing (Figure 5.4q) shows the arrangement of the I/O and support hardware. The point-to-point wiring diagram (Figure 5.4r) is used by the panel shop and the installation contractor to make the I/O device interconnect. Panel, or enclosure, design should now be coordinated with the addition of panel instrumentation, such as light switches, meters, and recorders. Once these steps are completed, panel fabrication and assembly can begin.

**I/O Quantity and Type (Example 2)** In this somewhat more complicated example, the user should arrange any special I/O types as well as the commonly available modules according to an I/O matrix by logical area, as shown in Table 5.4s. In this table, I/O types are listed across the first row and plant areas are listed down the first column. In this way, one can accurately reconstruct the decision-making process concerning I/O quantity and type. It is important to include at least 10 to 20% spare rack space in all I/O considerations.

For example, consider a typical process application. Assume a total I/O count of 764, broken down into 436 inputs and 328 outputs. This application falls into the medium PLC category. Because the majority of the field devices are located a good distance from the CPU, a PLC with remote I/O is desirable. The I/O requirements by locations are as follows:

**Process Area**: Total of 390 inputs and outputs, of which 230 are 24 V (discrete DC inputs), 24 are 4–20 mA analog inputs, and 136 are 24 V DC discrete outputs.

**Tank Farm #1**: Total of 98 I/O, of which 62 are 120 VAC discrete inputs (DIIs) and 36 are 120 VAC discrete outputs.

**Tank Farm #2**: Total of 86 I/O, of which 52 are 120 VAC discrete inputs and 34 are 120 VAC discrete outputs.

**Loading Station**: Total of 190 I/O, of which 68 are 120 VAC discrete inputs and 122 are discrete outputs (of which 24 are 120 VAC and 98 are 240 VAC).

Let us assume that the PLC system being considered has the following features: (1) No constraints on input and output mixture; (2) the I/O modules are available in two formats, 16 points per module and 8 points per module; and (3) 10% spare I/O is required. For the sake of illustration, we will use the 16 points per module I/O structure in the process area and the 8 points per module structure in the tank farms and loading station. The I/O distribution (including spares) per location would now be as follows:

**Process Area**: Total of 390 I/O points, which require the following modules:
- 230 + 10% = 253 24 V DC inputs with 16 points per module = 15.8 modules; use 16 modules
- 24 + 10% = 26.4 4–20 mA analog inputs at 16 points per module = 1.6; use 2 modules
- 136 + 10% = 149.6 24 V DC discrete outputs at 16 points per module = 9.35; use 10 modules

**Tank Farm #1**: Total of 98 I/O points, which require the following modules:
- 62 + 10% = 68.2 are 120 VAC discrete inputs at 8 points per module = 8.5; use 9 modules
- 36 + 10% = 39.6 are 120 VAC discrete outputs at 8 points per module = 4.9; use 5 modules

**Tank Farm #2**: Total of 86 I/O points, which require the following modules:
- 52 + 10% = 57.2 are 120 VAC discrete inputs at 8 points per module = 7.2; use 8 modules
- 34 + 10% = 37.4 are 120 VAC discrete outputs at 8 points per module = 4.7; use 5 modules
TABLE 5.4s
I/O Matrix

<table>
<thead>
<tr>
<th>Plant Area</th>
<th>Analog In</th>
<th>Analog Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process area 1</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>Tank Farm #1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tank Farm #2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Loading station</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plant Area</th>
<th>Analog In</th>
<th>Analog Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process area 1</td>
<td>4-20 mA</td>
<td>2</td>
</tr>
<tr>
<td>Tank Farm #1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tank Farm #2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Loading station</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model No.</th>
<th>Qty</th>
<th>Discrete In Voltage</th>
<th>Model No.</th>
<th>Qty</th>
<th>Discrete Out Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>230</td>
<td>24 VDC</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>120 VAC</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>120 VAC</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>120 VAC</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>240 VAC</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
(1) Discrete in 16 pts/card
Discrete out 16 pts/card
Analog in 16 pts/card
Analog out
(2) Add model numbers after award of contract.
(3) DI and DO are 8 points per card

Loading Station: Total of 190 I/O points, which require the following modules:
- 68 + 10% = 74.8 are 120 VAC discrete inputs at 8 points per module = 9.4; use 10 modules
- 24 + 10% = 26.4 are 120 VAC discrete outputs at 8 points per module = 3.3; use 4 modules
- 98 + 10% = 107.8 are 240 VAC discrete outputs at 8 points per module = 13.5; use 14 modules

If we assume that one remote communications channel can service up to 128 I/O in groups of sixteen 8-point modules or eight 16-point modules, the system becomes:
- Process Area—390 I/O; 28 16-point modules; four remote channels
- Tank Farm #1—98 I/O; 14 8-point modules; one remote channel
- Tank Farm #2—86 I/O; 13 8-point modules; one remote channel
- Loading Station—190 I/O; 28 8-point modules; two remote channels

Remote I/O vs. Distributed Control A unique feature of the PLC is the multiplexed nature of the I/O bus. This can be used to great advantage to reduce overall wiring cost. If I/O racks are centralized in logical clusters, plant wiring requirements can be greatly reduced. Wiring between racks and the CPU can be reduced to a few twisted pairs of wires or a single cable. The tremendous cost savings that result can be realized without a compromise of control accuracy or capability.

A system configuration diagram (such as that shown in Figure 5.4q), when used in conjunction with the I/O matrix in Table 5.4s, aids in keeping track of the overall system configuration.

It is important to remember the major weakness of remote I/O systems. If the bus is cut or interrupted, the effects of I/O failure will be relatively unpredictable. One must consider the effect of a possible system failure on each step in the sequence. Some users install a redundant version of remote I/O communications to guard against the loss of remote I/O communications. For this reason, duplication of smaller CPUs at each remote location is often considered preferable to a large central CPU. This is actually an extension of distributed control within the network of the PLC itself. This approach can be very cost-effective, because requirements for the central unit size can be reduced. Serious consideration should be given to distributed versus centralized architecture in remote I/O systems in which control system integrity is important.

Memory Quantity and Type The type and quantity of PLC memory used depends on the controller’s size and the company that manufactured it. Most small PLCs come with...
a fixed quantity of RAM. Although this is usually 2 K to 4 K of memory, the actual number of memory locations is not as important as the average size application program the PLC can be expected to handle. (In this case, size refers to the number of I/O points that are to be controlled and the average number of logic, timer, counter, and mathematics operations that are to be performed.) Some manufacturers may provide an extra-expense option of EEPROM or Flash memory with their small PLCs.

Midsize and large PLCs provide users an option for almost any type of memory desired. This includes various types of nonvolatile memory. Quantity limits imposed by the PLC will exceed most application demands. When this is not the case, it usually suggests that a more efficient control scheme is in order or that the application really does not belong on a PLC in the first place.

Total memory, as stated in the manufacturer’s literature, does not necessarily mean the entire content is available to the user. Some manufacturers reserve large blocks for the PLC executive. A system with 4 K of 16-bit words of user memory may comfortably accommodate a program, whereas another system with 8 K of 8-bit words may have too little memory for the same program.

Special programming language features are an important aspect of memory sizing, especially in process control. The PID algorithm is a perfect example: One manufacturer requires 33 words of user-available memory, whereas another may need in excess of 1000 words. Obviously, the memory sizing for a loop control program would vary in these two systems. Another example is the use of special functions, such as shift registers. An alternative way of developing a shift register in ladder logic is to use a special function shift register or handling data to require less user memory. Word (or register) moves are also powerful in terms of memory efficiency. Programming languages, which can be binary- or octal-based or alphanumeric Boolean, affect memory use. The closer the language is to a machine code (binary-based), the more user memory is required to perform the more complex functions. The closer the language is to alphanumeric Boolean, the less memory will be required for complex functions.

The best way to determine program memory prerequisites is to write a representative sample program reflecting some actual project requirements and to request information about user memory size from the various manufacturers. If the manufacturer’s suggestions are followed, the user can be reasonably assured that the memory will not be undersized.

The final area of caution about memory size concerns the consideration of data storage. Data tables, scratch pads, and historical data retrieval requirements can inflate the size of the PLC memory. It should be remembered that the primary task of a PLC is control of the process. If data requirements are large, connection to auxiliary devices, such as mini- and microcomputers, should be given serious consideration. Many of these devices are currently available and are of an industrial grade; furthermore, the price of these systems is coming down rapidly. It is not good engineering practice to degrade control capabilities by burdening the PLC with excessive data acquisition functions. As a plant goes online, operational requirements for data generally increase astronomically. These will be easily accommodated by a mini- or microcomputer but not by the PLC memory.

**PLC Installation and Panel Design**

Installation of PLC systems is not a difficult or mysterious procedure, but the following general rules will save time and trouble for the systems designer or installer. The basic principles of PLC installation are the same as those for the installation of relay or other control systems. Safety rules and practices governing proper use of electrical control equipment in general should be observed. These include correct grounding techniques, placement of disconnect devices, proper selection of wire gauge, fusing, and logical layout of the device. PLCs can often be retrofitted into existing hardwired relay enclosures because they are designed to withstand the typical plant environment. PLC vendors provide installation manuals upon request.

After the PLC equipment is selected, the support equipment will need to be identified. Some panel shops that build PLC panels can assist or perform this additional design. At the panel level this includes terminal blocks, wireway, wire type and size both for discrete and analog, loop power supply (usually 24 V DC), circuit breakers and fuses, isolation relays, intrinsic safety barriers (if I/O wiring enters flammable or explosive environments as defined by national electric code), main isolation transformer, pilot lights, pushbuttons and switches, internal panel work lights, and receptacle. Also included are any door-mounted equipment like chart recorders, digital panel meters, and operator interface panels. A UPS may be required. Separate isolated grounding is sometimes used for loop shields. Motor starters and small variable frequency drives may also be in the same enclosure.

After all the equipment in the panel is determined, a heat calculation should be done for the ambient extremes expected, particularly if the cabinet is outdoors, to verify the internal temperature is well below the maximum temperature of all components. Generally the PLC is not the limiting factor here, but other electronics and power supplies may limit the maximum allowable temperature. If the interior temperature is too high, a heat exchanger or air conditioner may be required. If the panel is installed in a cold area, a heater may be needed to prevent condensation, which can ruin electronic equipment.

**Safety Considerations** Perhaps the most important safety feature, which is often neglected in PLC system design, is emergency stop and master control relays. This feature must be included whenever a hardwired device is used in order to ensure operator protection against the unwanted application of power. Emergency stop functions should be completely hard-wired (Figure 5.4t). In no way should any software functions...
be relied upon to shut off the process or the machine. Disconnect switches and master control relays should be hardwired to cut off power to the output supply of the PLC. This is necessary because most PLC manufacturers use triacs for their output switching devices, and triacs are just as likely to fail on as off. This feature is often required by local or national codes. In the United States, NFPA 79, “Electrical Standard for Industrial Machinery,” covers this subject.

Implementation Planning ahead is every bit as important in designing a complete PLC system as in laying out a relay logic panel. Care in counting I/O points in the beginning— and leaving a safety factor—will save headaches in the panel fabrication stage. Panels should always have plenty of expansion room left over, because I/O is invariably added as the job progresses and the operators see the advantages of PLCs. The designer should refer to the layout considerations provided by the manufacturer. Extra space should be left to provide access to the boards and connectors of the PLC. The diagnostic and status indicators should all be visible. The designer should leave room between I/O racks for wireways and large hands.

One good technique for ensuring efficient panel layouts is to involve maintenance personnel in the design procedure. This not only optimizes the layout but also introduces the staff to the hardware (Figure 5.4u).

In general, the best defense against creating a tangled mess when designing a PLC system is to follow proper documentation techniques. A little more time spent documenting panel layout, I/O counts, and wiring diagrams results in a lot less time spent starting up the system. PLCs can handle large amounts of I/O points with varying electrical characteristics, so things can get pretty confusing in a hurry. Cable requirements between hardware boxes vary from one type of PLC to another, so this is an important consideration in panel layout.

Enclosure Enclosures should nearly always be provided for the PLCs themselves. This protects the electronics from moisture, oil, dust particles, and unwanted tampering. Most
manufacturers recommend a NEMA 12 enclosure for the standard industrial environment or a NEMA 4X for outdoor or corrosive environments. A NEMA 3R is not recommended because it is not sealed against dust and moisture. This type of enclosure is readily available in a variety of sizes and, in fact, may be already included with a new system.

PLCs are designed to be located close to the machine or the process under control. This keeps the wiring runs short and aids in the troubleshooting procedure. At times, however, mounting the PLC directly on the machine or too close to the process is not advisable, such as in cases of vibration inherent in the machine, electrical noise interference, or excessive heat problems. In these situations, the PLC must be either moved away or successfully protected against these environmental conditions.

**Temperature Considerations** Installing any solid-state device requires paying attention to ambient temperatures, radiant heat bombardment, and the heat generated by the device itself. PLCs are typically designed for operation over a broad range...
of temperatures, usually from 0 to 60°C. When analyzing the proposed PLC environment, however, one should remember that enclosure temperatures usually run a few degrees higher than ambient temperatures. Radiant heat on an enclosure from surrounding tanks can raise the internal temperature beyond that specified by the manufacturer.

Heat generated by the PLC is a key issue when the device is placed in ambient temperatures close to the extreme mentioned in the specifications. The temperature rise caused by the power consumption of the PLC itself is not hard to estimate. In addition, most manufacturers will provide a notation of the power consumption of the triacs driving field loads. When designing the hardware layout within the panel, one should adhere to the manufacturer’s suggestions regarding ways to minimize heating problems. Most PLCs use convection over fins to take heat away from particular areas within the hardware. Care must be taken to ensure that no obstruction to air flow over these fins is introduced by placement of the PLC in the enclosure. Wireways are typically provided with holes to allow air to pass through. Generally, one can avoid problems with PLC enclosures by simply leaving plenty of air space around the heat producers.

Should all of these factors combine to cause a temperature problem, the panel can be vented, air conditioned, or moved to another location. Usually, simply blowing filtered air through the enclosure will resolve minor difficulties. If air conditioning is required, small units that are designed for cooling electronic enclosures are readily available.

Noise Noise or unwanted electrical signals can generate problems for all solid-state circuits, particularly microprocessors. Each PLC manufacturer suggests methods for designing a noise-immune system. These guidelines should be strictly followed in the design and installation phases, because noise problems can be very difficult to isolate after the system is up and running. I/O systems are isolated from the field, but voltage spikes can still appear within the low-voltage environment of the PLC if proper grounding practices are not followed.

A well-grounded enclosure can provide a barrier to noise bombardment from outside. Metal-to-metal contact between the PLC and the panel is a must, as is a good connection from the panel to the ground (see Figure 5.4v). Noise producers within the panel should be noted during the panel design phase, and the PLC must not be located too close to these devices. Wiring within the panel should also be diverted around noise producers to avoid picking up any stray signals. Often, it is necessary to keep AC and DC wiring bundles apart, particularly when high-voltage AC is used at the same time that low-level analog signals are present. Refer to Tables 5.4w and 5.4x and Figure 5.4y for recommendations.

### TABLE 5.4w

*Follow These Guidelines for Grouping Conductors with Respect to Noise (Courtesy of Allen-Bradley)*

<table>
<thead>
<tr>
<th>Group Conductor Cables Fitting This Description</th>
<th>Into This Category:</th>
<th>Examples:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control and AC power—High-power conductors that are more tolerant of electrical noise than category 2 conductors and may also cause more noise to be picked up by adjacent conductors</td>
<td>Category 1</td>
<td>AC power lines for power supplies and I/O circuits.</td>
</tr>
<tr>
<td>Corresponds to IEEE levels 3 (low susceptibility) and 4 (power)</td>
<td></td>
<td>High-power digital AC I/O lines—to connect AC I/O modules rated for high power and high noise immunity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High-power digital DC I/O lines—to connect DC I/O modules rated for high power or with input circuits with long time-constant filters for high noise rejection. They typically connect devices such as hard-contact switches, relays, and solenoids.</td>
</tr>
<tr>
<td>Signal and communications—Low-power conductors that are less tolerant of electrical noise than category 1 conductors and should also cause less noise to be picked up by adjacent conductors (they connect to sensors and actuators relatively close to the I/O modules)</td>
<td>Category 2</td>
<td>Analog I/O lines and DC power lines for analog circuits.</td>
</tr>
<tr>
<td>Corresponds to IEEE levels 1 (high susceptibility) and 2 (medium susceptibility)</td>
<td></td>
<td>Low-power digital AC/DC I/O lines—to connect to I/O modules that are rated for low power such as low-power contact-output modules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low-power digital DC I/O lines—to connect to DC I/O modules that are rated for low power and have input circuits with short time-constant filters to detect short pulses. They typically connect to devices such as proximity switches, photoelectric sensors, TTL devices, and encoders. Communications cables (ControlNet, DeviceNet, Universal remote I/O, extended-local I/O, DH+, DH-485, RS-232-C, RS-422, RS-423 cables)—to connect between processors or to I/O adapter modules, programming terminals, computers, or data terminals.</td>
</tr>
<tr>
<td>Intra-enclosure—Interconnect the system components within an enclosure</td>
<td>Category 3</td>
<td>Low-voltage DC power cables—to provide backplane power to the system components.</td>
</tr>
<tr>
<td>Corresponds to IEEE levels 1 (high susceptibility) and 2 (medium susceptibility)</td>
<td></td>
<td>Communications cables—to connect between system components within the same enclosure.</td>
</tr>
</tbody>
</table>
Line voltage variations can cause hard-to-trace problems in the operation of any computer-based system. PLCs are no exception, even though they are designed to operate over a much larger variation in supply voltage. Large spikes or brownout conditions can cause errors in program execution. Most manufacturers protect against this, enabling the controller to come up running after a brownout, but these measures may not be acceptable in all applications. The designer may wish to add an isolation transformer to a proposed PLC system, sized for twice the anticipated load. This is cheap insurance, and PLC manufacturers will help determine the required load.

Triac outputs require some special attention that will be new to relay users. Triacs used for AC loads typically leak a small amount of current. In the case of triac outputs from a PLC, this leakage may be enough to keep panel lamps glowing or small relays energized. When a triac is used to switch the input on a PLC, the leakage may be enough to make the PLCs “think” the input is on. A dummy load (shown in Figure 5.4z) can be used to drain this leakage when the input should be off. Whenever a mechanical contact is used in series with a load energized by a triac (as shown in Figure 5.4aa), a resistance-capacitance (RC) network should be used as shown to protect the triac from inductive kickback. A varistor should be provided in parallel with a load whenever the load can be “hot-wired” around the triac (Figure 5.4bb). The user should check with the PLC manufacturer for the suggested RC and metal oxide varistor (MOV) types for the particular application. Triacs cannot directly drive large motor starters and similar devices. PLC manufacturers provide surge specifications for the various I/O cards. Sometimes an interposing relay or dry contacts will be required for large loads.

PLCs are similar to most electrical control systems. To be sure, solid-state devices, microprocessors, and triacs require some special considerations during the design, installation, and start-up phases of a project, but these concepts are not too complex or difficult to assimilate. As always, good design habits in the beginning will ensure a safe and reliable control system.

**Hookup**  PLC panels can be very neat and orderly if all the terminals are arranged in a logical fashion. The actual result is a direct function of the time spent during the design process. Interposing terminal blocks between the PLC I/O structure and the field is suggested, because the terminations provided by PLC manufacturers are shrinking in the race to provide higher-density I/O. This also gives the panel designer the ability to place the field termination points where they are easily accessed. Wiring ducts keep the panel neat and protect the wire from mishap.

Following good wiring practices can avert many noise problems. Low-voltage signal wiring should be kept away from noise sources. Analog signals should be shielded, with the shield terminated at an isolated ground in the panel only (to prevent shield ground loops). Again, these analog signals should be separated from power wiring.

**Software (Program) Development**

The I/O list mentioned previously will be used to begin the program development. Basic control philosophy decisions need to be made at this point. Should valves fail open or closed? What fail-safe provisions are necessary for analog control? These philosophical decisions should be documented and included with the process operational descriptions. Often this document will be referred to as the software functional specification. Its purpose is to define, as precisely as possible, the operation of the controls. It also has several other functions:

1. It communicates the functional requirements of the control system to those writing the PLC code.
2. It records the thought process (regarding control) of the system designer to be used in the event of a personnel change. Such information can be invaluable.

3. It provides a review document for personnel working in other capacities (mechanical, process, and project management) to ensure that they understand the operation of the controls.

4. It provides a guide for developing the operational description for the operator’s manual.

After the functional specification has been reviewed and approved, a detailed operational sequence chart, timing diagram, logic diagram, flowchart, or electrical schematic is developed from it. This schematic is translated or coded into the appropriate PLC language, cross-referencing I/O with PLC designator tags. The piping and instrument diagram is also cross-referenced with PLC designators. In this way, future cross-referencing of system drawings and PLC codes is facilitated.

As the code is entered, a memory map or register index is kept by the programmer (Table 5.4cc). This map is useful for organizing program data in logical arrangements and will prove invaluable during start-up, when the programmer may need to locate available blocks of memory quickly for program revisions. Most PLC programming software can generate a cross-reference; however, certain types of instructions are difficult for the program to cross-reference, so a list is still desirable.

The use and understanding of PLC programming depends on knowledge of the process to be controlled, an understanding of electrical schematics, and an appreciation for logic operations and for various types of logic and relay devices. Other sections of this chapter provide information on each of these topics. A review of those topics might be useful at this point.

Although the programming style and language used is, to some extent, dictated by the size of the PLC used, there are fundamental programming elements, including logic operations, timers, counters, and arithmetic capabilities, that are provided in all models. Some of the characteristics of these important elements are briefly discussed below (see Sections 5.5 and 5.6 for more information). Ladder logic continues to be popular because it was purposely created to look like hardwired relay logic drawings already in use. It is this similarity between ladder diagrams used for relays and the programming version that eased the change from using relays to using PLCs. In addition, ladder logic is relatively easy to learn.

**Boolean Logic** Relay type instructions are the most basic of all PLC instructions. Boolean logic functions (AND, OR, and NOT) are diagrammed as combinations of normally open and normally closed contacts. A coil symbol represents the result of the logic function. In ladder logic, contacts can be shown in series or parallel to achieve the “AND” or “OR” functions, respectively. A normally closed contact represents a “NOT” function.
PLCs and Other Logic Devices

One popular programming technique involves defining the sequential logic in electrical schematic format, using actual tag numbers, and then translating this diagram into the appropriate programming language. Figure 5.4dd shows the translation of some examples of typical circuits to ladder diagrams, Boolean algebra, and mnemonics. Because this translation is relatively simple, maintenance and engineering personnel have accepted PLCs, although they have not accepted computers as readily. The unknown has been replaced with the familiar.

Timing and Counting Figure 5.4ee is a schematic representation of a timer and a counter. Although their formats differ, the principles are the same. The legs of the timer represent start/stop and reset. Timers can be on-delay or off-delay and can be cascaded (that is, linked together in series). Counters can be up or down and have a count leg (in which the number of switch closures is the count) and an up/down leg (in which the position of the switch determines up count or down count). A PLC with arithmetic capability can use a combination timer and counter as an integrator. Figure 5.4ff shows a turbine meter pulse counter turned into a low-cost integrator within the PLC. Obviously, the scan rate of the PLC (scans per second) must be twice the pulse rate of the turbine (pulses per second).

Arithmetic Capabilities Figure 5.4ff shows an arithmetic program that permits the rapid addition of pulse counts from two counters hooked to two electric meters. The resulting sum is displayed through a panel meter. This logical addition is performed using integer mathematics (that is, no decimal calculations can be performed). Most PLCs use an approximation technique called “double-precision integer mathematics” to do calculations of greater complexity (such as PID). Some PLCs have true floating-point mathematics capability.

Floating-point mathematics is a powerful tool for process applications. For example, in the integrator example in

FIG. 5.4aa
Examples of where to use suppression. (Courtesy of Allen-Bradley.)

Example 1: An AC output module controls a motor starter whose contacts control the motor. The motor needs suppressors because it is an inductive load switched by hard contacts.

Example 2: An AC output module controls an interposing relay, but the circuit can be opened by hard contacts. The relay contacts control a solenoid. The interposing relay needs a suppressor because it is an inductive load switched by hard contacts.

Example 3: A contact output module controls an inductive load.

The motor needs suppressors because it is an inductive load switched by hard contacts.

The solenoid needs a suppressor because it is an inductive load switched by hard contacts.

The pilot light needs a suppressor because it is an inductive load switched by hard contacts.

FIG. 5.4bb
Surge suppression for inductive AC load devices. (Courtesy of Allen-Bradley.)

© 2006 by Béla Lipták
Figure 5.4gg, the division of pulses by elapsed time can be expressed as a decimal number rather than as a truncated integer. Feedforward calculations and PID can be performed in double-precision integer mathematics but are more memory-intense than in floating-point mathematics. Floating-point mathematics may require the use of a separate microprocessor within the CPU and usually involves two adjacent memory locations to store the mantissa and the abscissa in a form of scientific notation. The programmer automatically translates when the memory location is followed by a period (.), indicating a floating point number.

| TABLE 5.4cc |
| Example of Memory Map for Milling Machine |

<table>
<thead>
<tr>
<th>Coil</th>
<th>Definition</th>
<th>Used in Rung(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR0</td>
<td>R.H. Head Retract</td>
<td>13 15 19 –20 –33 –46</td>
</tr>
<tr>
<td>CR1</td>
<td>L.H. Head Retract</td>
<td>14 15 21 –33 –38 –47</td>
</tr>
<tr>
<td>CR2</td>
<td>Clamp Part</td>
<td>4 5 –34</td>
</tr>
<tr>
<td>CR3</td>
<td>Cycle Part Unclamp</td>
<td>2 2 25 25</td>
</tr>
<tr>
<td>CR4</td>
<td>Shot Pins Out</td>
<td>5 5 6 27 –28</td>
</tr>
<tr>
<td>CR5</td>
<td>Machine Slide Adv Milling</td>
<td>6 6 7 22 24 29</td>
</tr>
<tr>
<td>CR6</td>
<td>Machine Slide Retract</td>
<td>7 7 8 –22 39</td>
</tr>
<tr>
<td>CR7</td>
<td>Milling Spindles Off</td>
<td>9 9 10 –24 –24 –32</td>
</tr>
<tr>
<td>CR8</td>
<td>Start Boring Spindles</td>
<td>10 10 11 12 20 23</td>
</tr>
<tr>
<td>CR9</td>
<td>R.H. Bore Complete</td>
<td>11 11 13 13 46</td>
</tr>
<tr>
<td>CR12</td>
<td>Unclamp Pull Clamps</td>
<td>–3 16 16 17 26</td>
</tr>
<tr>
<td>CR13</td>
<td>Hold Light On</td>
<td>3 3 4 4 –26 34</td>
</tr>
<tr>
<td>CR15</td>
<td>Milling Complete Shot Pins In</td>
<td>8 8 9 –27 –39</td>
</tr>
<tr>
<td>CR17</td>
<td>Pump #1 Pressure</td>
<td>25 26 28 52</td>
</tr>
<tr>
<td>CR18</td>
<td>Unclamp Pull Clamp</td>
<td>26 49</td>
</tr>
<tr>
<td>CR19</td>
<td>Retract Shot Pins</td>
<td>27 50</td>
</tr>
<tr>
<td>CR20</td>
<td>Unclamp Swing Clamps</td>
<td>28 51</td>
</tr>
<tr>
<td>CR29</td>
<td>L.H. Bore Complete</td>
<td>12 12 14 14 47</td>
</tr>
<tr>
<td>CR31</td>
<td>Pressure Off Pump Time Start</td>
<td>17 17 18 18</td>
</tr>
<tr>
<td>CR32</td>
<td>Pressure Off Pump #1</td>
<td>18 –52</td>
</tr>
<tr>
<td>CR101</td>
<td>Auto Lube Cycle</td>
<td>36 36 40 40 41 41</td>
</tr>
<tr>
<td>CR102</td>
<td>Lube Off</td>
<td>41 42 42 –43 –44 45</td>
</tr>
<tr>
<td>CR103</td>
<td>Lube On</td>
<td>–48</td>
</tr>
<tr>
<td>CR104</td>
<td>60 Min. Lube Restart</td>
<td>41 –41 42</td>
</tr>
<tr>
<td>CR105</td>
<td>Start/ Disable</td>
<td>43 44</td>
</tr>
<tr>
<td>CR106</td>
<td>Contin. Lube Restart</td>
<td>40 45 45</td>
</tr>
</tbody>
</table>

Source: Xcel

The following technique can be used for PLC programming applications. There are many advantages to this approach.

1. Develop detailed I/O lists. Table 5.4p shows an I/O cross-reference relating tag numbers to I/O points. This list should be used extensively; starting without it will cause confusion and errors resulting from inevitable changes.

2. Develop a detailed descriptive operational sequence of events. Figure 5.4ii shows a sample sequence using a process batch application.
PLCs and Other Logic Devices

**FIG. 5.4dd**
Ladder translation. Here is a comparison of programming languages that are used with various PLCs. The most popular is still the relay ladder diagram because plant personnel are more familiar with it.

**FIG. 5.4ee**
Timer/counter schematic. A key part of any PLC programming is its capability to do timer/counter functions. The instructions are entered either horizontally or vertically, depending on the make. Horizontal programming, however, is more commonly used.

**FIG. 5.4ff**
Pulse counter totalizer.

© 2006 by Béla Lipták
3. Develop electrical schematics or ladder diagrams for sequential control.

4. Develop piping and instrumentation drawings (or a logic diagram) for process control. Figure 5.4jj shows a diagram that will allow maintenance personnel to find important activities quickly. Note the I/O matrix index, showing what happens inside the PLC software, and the cross-referenced I/O memory locations and tag numbers.

5. Translate the drawings in steps 3 and 4 to the PLC language.

6. Enter the program code using a memory map (see Table 5.4cc).

7. Debug the program at the programmer’s facility. Use a simulator to debug the program. Run through the operational sequence defined in step 2. If it has changed, be sure to ascertain how that change has affected other parts of the program. Rewrite the sequence description to reflect current operations, if necessary.

8. Save and document the program. Reproduce the program on transportable media, such as floppy disks. (Do this daily.) Document programming changes using printouts.

9. Enter and debug the program in the field. It is essential to note all changes made on the documentation. Some of the biggest problems with relay systems are undocumented field modifications.

10. Redocument and reproduce the final program.

The final documentation package should include the following: I/O list and cross-reference, descriptive operational sequence, electrical schematics, process schematics, program listing (see Figure 5.4gg for annotated final document), memory maps (showing the memory areas that have been used and those that are available), and notes for future program changes or additions.

One important word about the documentation package: A major advantage of the PLC is its ability to be reprogrammed as plant requirements change. Without proper documentation, previous programming efforts will have to be reproduced in order to make the changes that are required. Poor documentation results in wasteful efforts at reconstruction. The PLC, like all engineering tools, requires good control systems engineering practices in order for its full potential to be realized. Good documentation is an essential element of any PLC project.

### Programming and Documentation Tools

Both PLC and after-market parties offer programming and documentation tools for the system designer or user. Programming software is typically provided by the PLC manufacturer and is designed to program a specific machine or family of machines. Some third parties are offering universal programming software. These software packages vary greatly in price and capabilities but offer on- and off-line programming to many different types of PLCs, real-time status, and some very sophisticated annotation. Communications to different PLCs are usually supported by different software packages. Each vendor’s product offers different types and amounts of ladder and contact comments. Again, many types of cross-references are available to be printed out. Often, other PLC design documentation problems may be solved, such as the generation of panel configuration drawings, point-to-point wiring diagrams, and I/O layout. One system even prints out the wire labels.

### HMI Software (Program) Configuration

The operator interface needs to be designed and implemented. Generally, the same process is followed whether the interface is a local panel or the SCADA server in the control room. Sketches of the displays should be made and reviewed with operations staff. These should emphasize operator-friendly graphics for the plant operators, who will spend the most time with them. Most of the time, operators need to view an operations-oriented view of the process. Only when an abnormal event occurs will they go to the detail screens to find the cause.

Once a consensus is reached on how the system should be operated, the screens can be configured on the interface equipment. “Linking” is the term used for defining how the screen will change based on the contents of PLC memory locations.
Reports Configuration  Some simple reports can be directly set up in the operator interface software. More extensive reports usually require separate reporting or database software. These reports should be set up early so that they can be checked before installation of the equipment.

Software/Hardware Integration

Once the program is entered, a simulation is recommended, and the program checkout process begins “on the bench.” This process uses the functional specification to prove the software is acceptable. A large percentage of the program can be proved in this manner. Program debugging can be completed before field installation. Field corrections will then be minimized, and high-salaried electrical and installation personnel will not be standing around waiting. The savings that can be realized are quite significant. In addition, it’s important to keep in mind that there’s no substitute for a bench-simulated program check. The software simulation proves the program and allows acceptance by the customer. The program should be reproduced and documented after it has been checked.

Once the panels are built, and all configuration and programming are complete, as much of the system as possible should be set up in a staging area for a complete checkout. Final control elements (valves, motor starters) function can be checked for proper outputs. Analog sensors (pressure, level, flow, and so on) are usually simulated with a 4–20 mA calibrator. Using the PLC and programming aids, the panel wiring can be “rung out” (that is, checked point by point) through the PLC. Each I/O point should be activated separately to the terminal block or from the panel controls (buttons, lights, switches). In this manner, the electrical integrity of the panel from the terminal blocks inward is ensured. If any continuity problems exist thereafter, they will be located in the field wiring. All remote I/O should be connected. The data highway or Ethernet networks should be temporarily connected. All displays should also be checked to make sure that the I/O points trigger the correct part of the screen or activate the correct alarm.

Some organizations prefer to perform a simulated operation checkout at this time. This is a highly useful approach and can be implemented if the simulators and the I/O point arrangement are organized to simulate the process outside the panel. Some I/O points may need to be jumpered for simulation. For instance, if run contacts are required to close a few seconds after the motor is called to run, they can often be temporarily connected to the output for the test. It is generally felt that each hour spent troubleshooting at this point will save two or three hours during system start-up. Occasionally, operations personnel are brought in
5.4 PLCs: Programmable Logic Controllers

System Checkout and Start-Up

After electrical interconnections are made and point-to-point wiring is completed (mechanical completion), the system is ready for start-up. The ability of the PLC to operate step by step through the start-up becomes very useful at this stage.

Experienced PLC personnel may provide temporary switches in the back of the panel in order to facilitate the start-up procedures. These switches can be key-locked, software-locked, or disconnected for normal operation. They are also very useful as future maintenance and troubleshooting tools to diagnose future problems.

Unanticipated circumstances are always a factor during start-up. Wiring errors, program errors, and mismatches between PLC and HMI databases are common problems at this stage. For this reason it is not uncommon to have electrical and programming personnel available at this time for implementation of any changes that might be necessary. Quite often, program changes can be accomplished over long-distance telephone lines using modems. These changes are not easily implemented without adequate documentation. After successful start-up, the plant is signed off.

After Start-Up

Once again, it is imperative for future successful plant operation that complete, current documentation be available. This documentation should include the items discussed previously. Especially useful for future changes or additions are the start-up notes and notes pertaining to future modifications.

Training of operations, maintenance, and engineering personnel should be timely and “hands-on.” It is useful to videotape these training sessions for future reference (e.g., for training of new personnel). Suggested programs for PLC training can be obtained from the PLC vendor. This is an important function provided locally at the job site, at a nearby metropolitan area, or at the PLC vendor factory.

Troubleshooting

When troubleshooting a PLC system it is good to remember that almost 80% of the time the problem is either outside the PLC entirely or in a broken I/O module. Program errors and wiring errors and mismatch of PLC/HMI databases can also cause these problems, but these are usually resolved during start-up. Some simple troubleshooting tips can be followed to decide if a problem is in PLC or elsewhere.

Discrete Inputs: Because DIs are usually high impedance, check the input voltages with a high impedance multimeter. Using a low impedance solenoid voltage checker on digital inputs can sometimes cause confusing results. For a discrete input, an indicator on the module shows the state of the input. This should match what is observed on the HMI screen or PLC programmer monitor. The commons for the DI card may be grouped or individual. Check the voltage between common for that point on the I/O card terminal block (or swing arm) and the point in question. If possible actuate the field device and verify that the PLC logic status and
PLCs are durable, delivering real-time control in a rugged and dependable package with no moving parts. They can be installed on either the factory floor or outdoors, withstanding temperature swings up to 60°C. PLCs have the critical ability to process sequential logic without experiencing faults in the operating system. Moreover, users continue to use PLCs because they know how to support them and understand the simple ladder logic language they use.10

The sheer number of PLC applications is enormous. According to a recent Control Engineering magazine poll, “The major applications for PLCs include machine control (87%), process control (58%), motion control (40%), batch control (26%), diagnostic (18%), and other (3%).” The results don’t add up to 100% because a single control system generally has multiple applications. Many sources documenting various PLC applications are listed in the Bibliography of this section.

**References**

5.4 PLCs: Programmable Logic Controllers


