
A PLC Primer

TABLE OF CONTENTS

What Is a PLC?	pg 3
Why Use PLCs?	pg 4
But What Exactly <i>Is</i> a PLC?	pg 5
A Little More About Inputs and Outputs	pg 7
And a Little More About the Control Program	pg 10
So How Does a PLC Keep All This Straight?	pg 13
To Sum It All Up	pg 14
Want To Learn More?	pg 15

WHAT IS A PLC?

A **programmable logic controller**, also called a *PLC* or *programmable controller*, is a computer-type device used to control equipment in an industrial facility. The kinds of equipment that PLCs can control are as varied as industrial facilities themselves. Conveyor systems, food processing machinery, auto assembly lines...you name it and there's probably a PLC out there controlling it.

In a traditional industrial control system, all control devices are wired directly to each other according to how the system is supposed to operate. In a PLC system, however, the PLC replaces the wiring between the devices. Thus, instead of being wired directly to each other, all equipment is wired to the PLC. Then, the control program inside the PLC provides the "wiring" connection between the devices. The **control program** is the computer program stored in the PLC's memory that tells the PLC what's supposed to be going on in the system. The use of a PLC to provide the wiring connections between system devices is called *softwiring*.

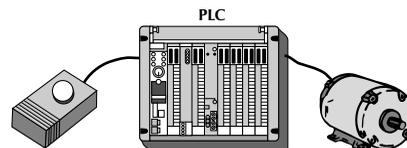
EXAMPLE

Let's say that a push button is supposed to control the operation of a motor. In a traditional control system, the push button would be wired directly to the motor. In a PLC system, however, both the push button and the motor would be wired to the PLC instead. Then, the PLC's control program would complete the electrical circuit between the two, allowing the button to control the motor.

In a traditional system, all control devices are wired directly to each other...



...In a PLC system, all control devices are wired to the PLC.



WHY USE PLCs?

The softwiring advantage provided by programmable controllers is tremendous. In fact, it is one of the most important features of PLCs. Softwiring makes changes in the control system easy and cheap. If you want a device in a PLC system to behave differently or to control a different process element, all you have to do is change the control program. In a traditional system, making this type of change would involve physically changing the wiring between the devices, a costly and time-consuming endeavor.

EXAMPLE

Let's say that two push buttons, PB1 and PB2, are connected to a PLC. Two pilot lights, PL1 and PL2, are also connected to the PLC. The way these devices are connected now pressing push button PB1 turns on pilot light PL1 and pressing push button PB2 turns on pilot light PL2. Let's say that you want to change this around so that PB1 controls PL2 and PB2 controls PL1. In a traditional system, you would have to rewire the circuit so that the wiring from the first push button goes to the second pilot light and vice versa. However, because these devices are connected to a PLC, making this change is as simple as making a small change in the control program.

In addition to the programming flexibility we just mentioned, PLCs offer other advantages over traditional control systems. These advantages include:

- high reliability
- small space requirements
- computing capabilities
- reduced costs
- ability to withstand harsh environments
- expandability

BUT WHAT EXACTLY IS A PLC?

A PLC basically consists of two elements:

- the central processing unit
- the input/output system

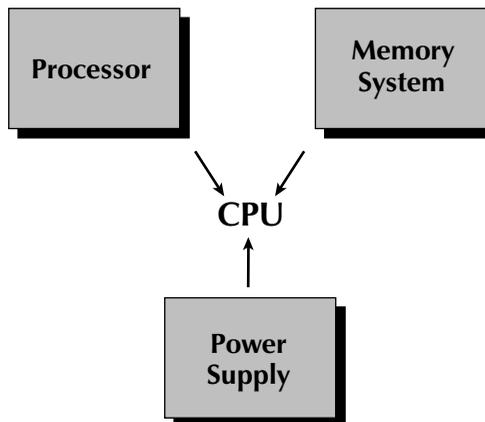
————— The Central Processing Unit —————

The **central processing unit** (CPU) is the part of a programmable controller that retrieves, decodes, stores, and processes information. It also executes the control program stored in the PLC's memory. In essence, the CPU is the “brains” of a programmable controller. It functions much the same way the CPU of a regular computer does, except that it uses special instructions and coding to perform its functions. The CPU has three parts:

- the processor
- the memory system
- the power supply

The **processor** is the section of the CPU that codes, decodes, and computes data. The **memory system** is the section of the CPU that stores both the control program and data from the equipment connected to the PLC. The **power supply** is the section that provides the PLC with the voltage and current it needs to operate.

*The CPU has three main parts...
...the processor...
...the memory system...
...and the power supply.*



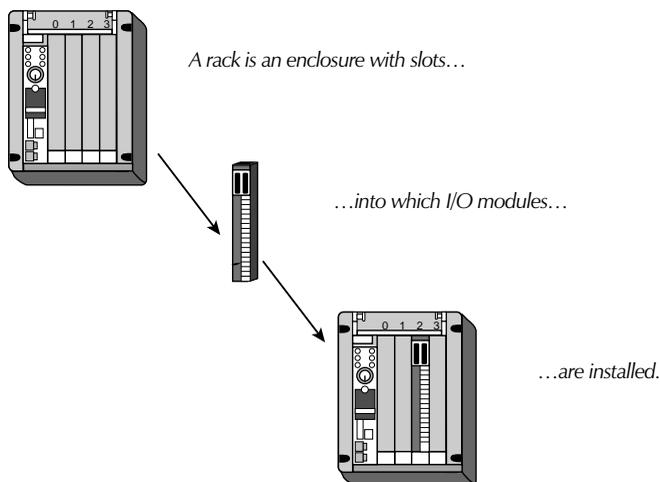
The Input/Output System

The **input/output (I/O) system** is the section of a PLC to which all of the field devices are connected. If the CPU can be thought of as the brains of a PLC, then the I/O system can be thought of as the arms and legs. The I/O system is what actually physically carries out the control commands from the program stored in the PLC's memory.

The I/O system consists of two main parts:

- the rack
- I/O modules

The **rack** is an enclosure with slots in it that is connected to the CPU. **I/O modules** are devices with connection terminals to which the field devices are wired. Together, the rack and the I/O modules form the interface between the field devices and the PLC. When set up properly, each I/O module is both securely wired to its corresponding field devices and securely installed in a slot in the rack. This creates the physical connection between the field equipment and the PLC. In some small PLCs, the rack and the I/O modules come pre-packaged as one unit.



A LITTLE MORE ABOUT INPUTS AND OUTPUTS

All of the field devices connected to a PLC can be classified in one of two categories:

- inputs
- outputs

Inputs are devices that supply a signal/data to a PLC. Typical examples of inputs are push buttons, switches, and measurement devices. Basically, an input device tells the PLC, “Hey, something’s happening out here...you need to check this out to see how it affects the control program.”

Outputs are devices that await a signal/data from the PLC to perform their control functions. Lights, horns, motors, and valves are all good examples of output devices. These devices stay put, minding their own business, until the PLC says, “You need to turn on now” or “You’d better open up your valve a little more,” etc.

EXAMPLE

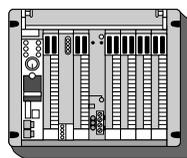
An overhead light fixture and its corresponding wall switch are good examples of everyday inputs and outputs. The wall switch is an input—it provides a signal for the light to turn on. The overhead light is an output—it waits until the switch sends a signal before it turns on.

Let’s pretend that you have a souped-up overhead light/switch circuit that contains a PLC. In this situation, both the switch and the light will be wired to the PLC instead of to each other. Thus, when you turn on the switch, the switch will send its “turn on” signal to the PLC instead of to the light. The PLC will then relay this signal to the light, which will then turn on.

An input device sends a signal to a PLC...



PLC



...An output device receives a signal from a PLC.

There are two basic types of input and output devices:

- discrete
- analog

Discrete devices are inputs and outputs that have only two states: on and off. As a result, they send/receive simple signals to/from a PLC. These signals consist of only 1s and 0s. A 1 means that the device is on and a 0 means that the device is off.

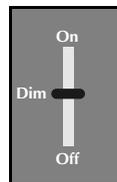
Analog devices are inputs and outputs that can have an infinite number of states. These devices can not only be on and off, but they can also be barely on, almost totally on, not quite off, etc. These devices send/receive complex signals to/from a PLC. Their communications consist of a variety of signals, not just 1s and 0s.

EXAMPLE

The overhead light and switch we just discussed are both examples of discrete devices. The switch can only be either totally on or totally off at any given time. The same is true for the light.

A thermometer and a control valve are examples of the other type of I/O devices—*analog*. A thermometer is an analog input device because it provides data that can have an infinite number of states. Temperature isn't just hot or cold. It can have a variety of states, including warm, cool, moderate, etc. A control valve is an analog output for the same reason. It can be totally on or totally off, but it can also have an infinite number of settings between these two states.

A discrete device can only be on or off...



...An analog device can be either on, off, or anywhere in between.

Because different input and output devices send different kinds of signals, they sometimes have a hard time communicating with the PLC. While PLCs are powerful devices, they can't always speak the "language" of every device connected to them. That's where the I/O modules we talked about earlier come in. The modules act as "translators" between the field devices and the PLC. They ensure that the PLC and the field devices all get the information they need in a language that they can understand.

...AND A LITTLE MORE ABOUT THE CONTROL PROGRAM

We talked a little bit earlier about the control program. The control program is a software program in the PLC's memory. It's what puts the *control* in a programmable controller.

The user or the system designer is usually the one who develops the control program. The control program is made up of things called **instructions**. Instructions are, in essence, little computer codes that make the inputs and outputs do what you want in order to get the result you need.

There are all different kinds of instructions and they can make a PLC do just about anything (add and subtract data, time and count events, compare information, etc.). All you have to do is program the instructions in the proper order and make sure that they are telling the right devices what to do and voila!...you have a PLC-controlled system. And remember, changing the system is a snap. If you want the system to act differently, just change the instructions in the control program.

Different PLCs offer different kinds of instructions. That's part of what makes each type of PLC unique. However, all PLCs use two basic types of instructions:

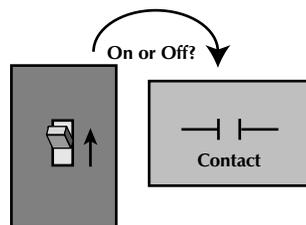
- contacts
- coils

Contacts are instructions that refer to the input conditions to the control program—that is, to the information supplied by the input field devices. Each contact in the control program monitors a certain field device. The contact waits for the input to do something in particular (e.g., turn on, turn off, etc.—this all depends on what type of contact it is). Then, the contact tells the PLC's control program, "The input device just did what it's supposed to do. You'd better check to see if this is supposed to affect any of the output devices."

Coils are instructions that refer to the outputs of the control program—that is, to what each particular output device is supposed to do in the system. Like a contact, each coil also monitors a certain field device. However, unlike a contact, which monitors the field device and then tells the PLC what to do, a coil monitors the PLC control program and then tells the field device what to do. It tells the output device, “Hey, the PLC just told me that the switch turned on. That means that you’re supposed to turn on now. So let’s go!”

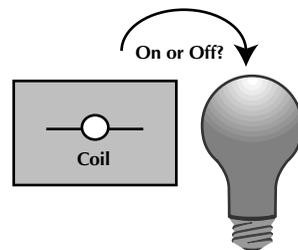
EXAMPLE

Let’s talk again about that souped-up switching circuit, in which a wall switch and an overhead light are connected to a PLC. Let’s say that turning on the switch is supposed to turn on the light. In this situation, the PLC’s control program would contain a contact that examines the input device—the wall switch—for an on condition and a coil that references the light. When the switch turns on, the contact will “energize,” meaning that it will tell the PLC that the condition it’s been looking for has happened. The PLC will relay this information to the coil instruction by energizing it. This will let the coil know that it needs to tell its referenced output—the light—to turn on.

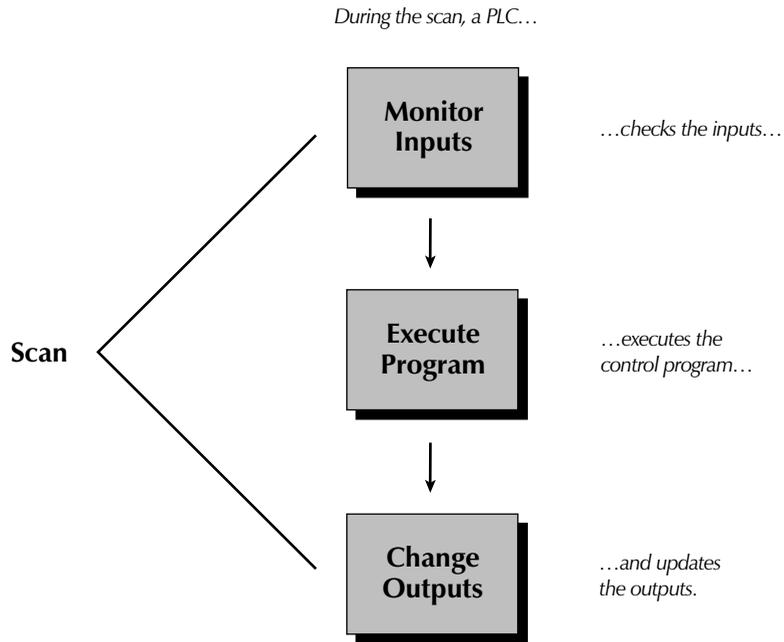


A contact is a computer code that monitors the status of an input...

...A coil is a computer code that monitors the status of an output.



In PLC talk, this three-step process of monitoring the inputs, executing the PLC control program, and changing the status of the outputs accordingly is called the **scan**.



So How DOES A PLC KEEP ALL THIS STRAIGHT?

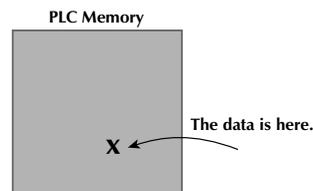
A PLC's memory system is very complex, allowing it to store information not only about the control program but about the status of all the inputs and outputs as well. To keep track of all this information, it uses a system called *addressing*. An **address** is a label or number that indicates where a certain piece of information is located in a PLC's memory. Just like your home address tells where you live in your city, a device or piece of data's address tells where information about it resides in the PLC's memory. That way, if a PLC wants to find out information about a field device, it knows to look in its corresponding address location.

Some addresses contain information about the status of particular field devices. Other addresses store data that's the result of control program computations. Still others contain reference data entered by the system programmer. Nonetheless, no matter what type of data it is, a PLC uses its addressing scheme to keep track of it all. That way, it'll have the right data when it needs it.

Just like your address tells where you can be found in your city...



...A device's address tells where it can be found in the PLC's memory.



To SUM IT ALL Up

PLCs can seem a little daunting at first, but there's no need to panic. Just remember that all PLCs follow the basic rules of operation we've just discussed. All PLCs have a CPU and an input/output system. They also all use a control program, instructions, and addressing to make the equipment in the control system do what it's supposed to do.

And no matter how many bells and whistles you add to it, every PLC does the same three things: (1) examines its input devices, (2) executes its control program, and (3) updates its output devices accordingly. So in reality, understanding PLCs is as simple as 1-2-3!

WANT TO LEARN MORE?

Industrial Text and Video offers a wide selection of training and reference materials in both PLCs and Electrical and Motor Controls to help you expand your knowledge. All of our packages contain videos, reference books, and computer software to help you grasp the essential details of each topic.

ABT-ITV900**Basic Introduction to PLCs Video Training Series**

Designed for maintenance personnel, this PLC course covers the basic operation of PLCs, as well as all the critical maintenance functions, such as troubleshooting the PLC and I/O system and preventive maintenance.

ABT-ITV500**Advanced PLC Video Training Series**

A comprehensive PLC program that's applicable to all makes and models of PLCs. With ten videos, computer software, and three reference books, this series has everything needed to become a PLC expert.

ABT-ITV1761**MicroLogix™ 1000 Controller Total Training Series**

This nine-tape program is designed to help electrical maintenance personnel—even those without any prior PLC knowledge—understand, program, and troubleshoot the MicroLogix™ 1000 controller.

ABT-ITV700**Electrical and Motor Controls Video Training Series**

Grasp the most critical and fundamental skills in the five most important electrical maintenance areas—power distribution, PLC technology, sensor and components, ladder diagrams, and motor controls. The packages are available individually or as a complete series.

ABT-ITV206BOOK**Programmable Controllers: Theory and Implementation, 2nd Edition**

Written by industry experts, this 1035-page reference covers important, up-to-date, real world programmable controller topics and applications. It's a generic PLC reference tool that will help you with all of the different makes and models of PLCs in your facility.

ABT-ITV206WKBK**Programmable Controllers: Workbook and Study Guide**

With over 800 questions and answers, this companion workbook is the perfect quick-reference tool.

For more information, contact Industrial Text & Video:

By Phone: 800-752-8398 By Fax: 770-240-2209

By Mail: 1950 Spectrum Cir., Tower A—1st Floor, Marietta GA 30067

By E-mail: sales@industrialtext.com

