

# Table of Contents

Introduction .....	2
PLCs .....	4
Number Systems .....	8
Terminology .....	12
Basic Requirements.....	18
S7-200 Micro PLCs.....	20
Programming a PLC.....	33
Discrete Inputs/Outputs .....	41
Analog Inputs and Outputs.....	48
Timers.....	51
Counters .....	58
High-Speed Instructions .....	61
Specialized Expansion Modules .....	65
Review Answers .....	72
Final Exam .....	73
quickSTEP Online Courses .....	76

# Introduction

Welcome to another course in the STEP series, **Siemens Technical Education Program**, designed to prepare our distributors to sell Siemens Energy & Automation products more effectively. This course covers **Basics of PLCs** and related products.

Upon completion of **Basics of PLCs** you should be able to:

- Identify the major components of a PLC and describe their functions
- Convert numbers from decimal to binary, BCD, and hexadecimal
- Identify typical discrete and analog inputs and outputs
- Identify key differences of the various S7-200 models
- Identify the types of expansion modules available for S7-200 PLCs
- Describe the types of programming available for S7-200 PLCs
- Describe the operation of commonly used program functions such as timers and counters
- Identify the proper manual to refer to for programming or installation of an S7-200 PLC

This knowledge will help you better understand customer applications. In addition, you will be better able to describe products to customers and determine important differences between products.

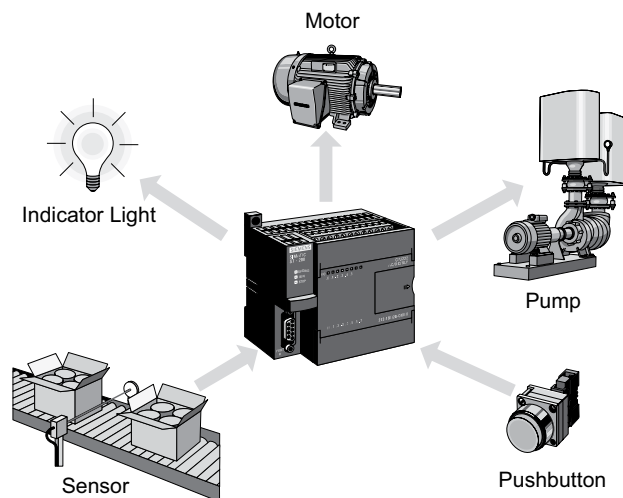
You should complete **Basics of Electricity** before attempting **Basics of PLCs**. An understanding of many of the concepts covered in **Basics of Electricity** is required for **Basics of PLCs**. In addition, you may wish to complete **Basics of Control Components**. Devices covered in **Basics of Control Components** are used with programmable logic controllers.

If you are an employee of a Siemens Energy & Automation authorized distributor, fill out the final exam tear-out card and mail in the card. We will mail you a certificate of completion if you score a passing grade. Good luck with your efforts.

Siemens is a trademark of Siemens AG. Product names mentioned may be trademarks or registered trademarks of their respective companies. Specifications subject to change without notice.

# PLCs

A **programmable logic controller (PLC)**, also referred to as a **programmable controller**, is the name given to a type of computer commonly used in commercial and industrial control applications. PLCs differ from office computers in the types of tasks that they perform and the hardware and software they require to perform these tasks. While the specific applications vary widely, all PLCs monitor inputs and other variable values, make decisions based on a stored program, and control outputs to automate a process or machine. This course is meant to supply you with basic information on the functions and configurations of PLCs with emphasis on the **S7-200 PLC** family.



## Basic PLC Operation

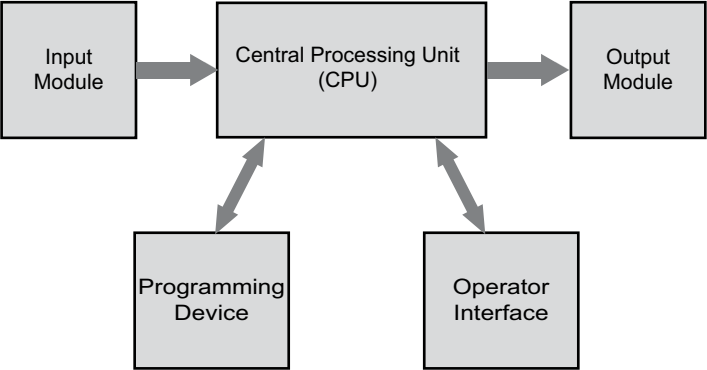
The basic elements of a PLC include **input modules** or **points**, a **central processing unit (CPU)**, **output modules** or **points**, and a **programming device**. The type of input modules or points used by a PLC depends upon the types of input devices used. Some input modules or points respond to digital inputs, also called discrete inputs, which are either on or off. Other modules or inputs respond to analog signals. These analog signals represent machine or process conditions as a range of voltage or current values. The primary function of a PLC's input circuitry is to convert the signals provided by these various switches and sensors into logic signals that can be used by the CPU.

The CPU evaluates the status of inputs, outputs, and other variables as it executes a stored program. The CPU then sends signals to update the status of outputs.

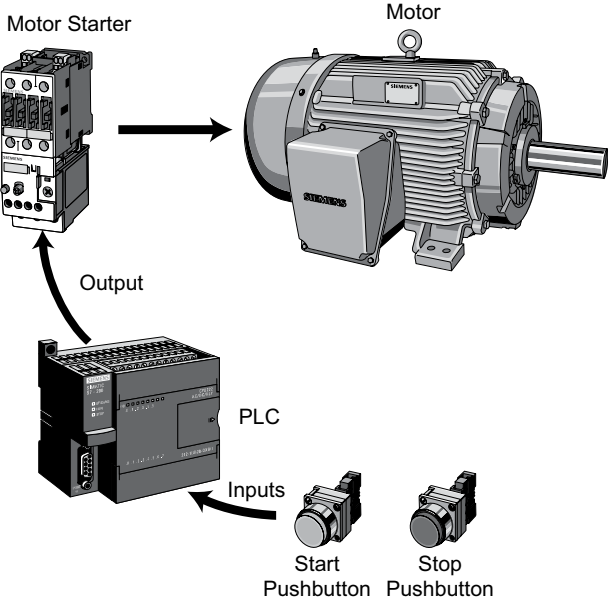
Output modules convert control signals from the CPU into digital or analog values that can be used to control various output devices.

The programming device is used to enter or change the PLC's program or to monitor or change stored values. Once entered, the program and associated variables are stored in the CPU.

In addition to these basic elements, a PLC system may also incorporate an operator interface device to simplify monitoring of the machine or process.

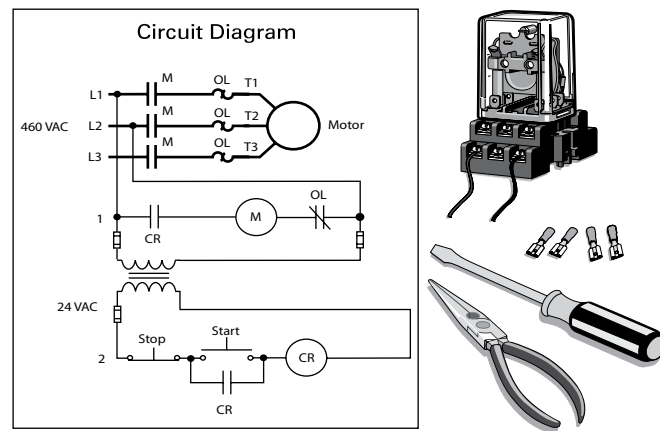


In the simple example shown below, pushbuttons (sensors) connected to PLC inputs are used to start and stop a motor connected to a PLC output through a motor starter (actuator). No programming device or operator interface are shown in this simple example.



## Hard-Wired Control

Prior to PLCs, many control tasks were performed by contactors, control relays, and other electromechanical devices. This is often referred to as **hard-wired control**. Circuit diagrams had to be designed, electrical components specified and installed, and wiring lists created. Electricians would then wire the components necessary to perform a specific task. If an error was made, the wires had to be reconnected correctly. A change in function or system expansion required extensive component changes and rewiring.



## Advantages of PLCs

PLCs not only are capable of performing the same tasks as hard-wired control, but are also capable of many more complex applications. In addition, the PLC program and electronic communication lines replace much of the interconnecting wires required by hard-wired control. Therefore, hard-wiring, though still required to connect field devices, is less intensive. This also makes correcting errors and modifying the application easier.

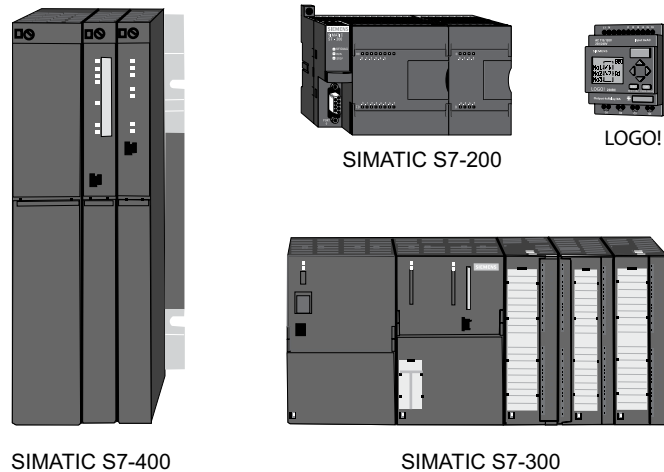
Some of the additional advantages of PLCs are as follows:

- Smaller physical size than hard-wire solutions.
- Easier and faster to make changes.
- PLCs have integrated diagnostics and override functions.
- Diagnostics are centrally available.
- Applications can be immediately documented.
- Applications can be duplicated faster and less expensively.

## Siemens Modular PLCs

Siemens **SIMATIC PLCs** are the foundation upon which our **Totally Integrated Automation (TIA)** concept is based. Because the needs of end users and machine builders vary widely, SIMATIC PLCs are available as conventional modular controllers, embedded automation products, or as PC-based controllers.

**Modular SIMATIC controllers** are optimized for control tasks and can be adapted to meet application requirements using plug-in modules for input/output (I/O), special functions, and communications. Examples of products in this category include: LOGO! and S7-200 micro automation products, S7-300 and S7-400 modular system PLCs, C7 combination controller and panel, and ET 200 distributed I/O system with local intelligence.



## Other SIMATIC Controllers

**SIMATIC embedded automation products** are available in a microbox, panel PC, or multi-functional PC-based system. All products utilize rugged, fan-free, diskless hardware platforms with an operating system optimized for each platform. Examples of products in this category include: Microbox 420-RTX, Microbox 420-T, Panel PC 477-HMI/RTX, and WinAC MP.

**SIMATIC PC-based controllers** are available as software that can run on standard PC systems or in a plug-in card (slot PLC) for increased reliability. This category includes WinAC software and WinAC slot PLC.

## SIMATIC Software

**SIMATIC software** is the universal configuring and programming environment for SIMATIC controllers, human machine interface systems, and process control systems. SIMATIC software with **STEP 7** and numerous engineering tools supports all phases of product deployment, from hardware configuration of the system and parameterization of modules to service of the installed system. A variety of programming options are available. This includes basic programming languages (Instruction List, Ladder Diagram, and Function Block Diagram), high-level languages (Structured Text and Sequential Function Chart), and engineering tools (S7 Structured Control Language, S7-Graph, S7-PLCSIM, S7-HiGraph, and Continuous Function Chart).

# Number Systems

Because a PLC is a computer, it stores information in the form of on or off conditions (1 or 0), referred to as **bits**. Sometimes bits are used individually and sometimes they are used to represent numerical values. Understanding how these bits can be used to represent numerical values requires an understanding of the **binary number system**.

## Decimal System

In order to understand the binary number system, it is first useful to recall some of the basics of the **decimal number system**. All number systems have the same three characteristics: digits, base, weight. For example, the decimal system has the following characteristics:

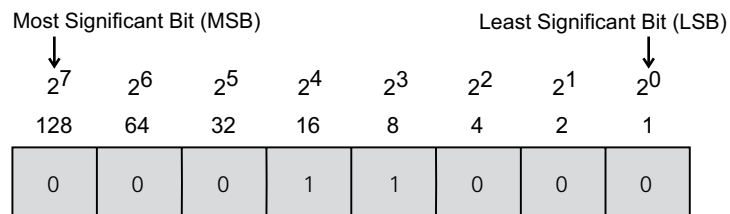
Ten digits      0, 1, 2, 3, 4, 5, 6, 7, 8, 9  
Base            10  
Weights        Powers of base 10 (1, 10, 100, 1000, ...)

## Binary System

The **binary system** has the following characteristics:

Two digits:    0, 1  
Base            2  
Weights        Powers of base 2 (1, 2, 4, 8, 16, ...)

The binary system has a base of 2 and uses only two characters, 1 and 0. Each bit is associated with a power of 2 based on its position in the number. The further to the left, the higher the power of 2. The number in the far left-hand column is referred to as the **most significant bit** or **MSB** and the number in the far right-hand column is referred to as the **least significant bit** or **LSB**. A 1 is placed in a position if that power of 2 is used in the number. Otherwise, a 0 is placed in a position.



00011000 in binary = 24 in Decimal



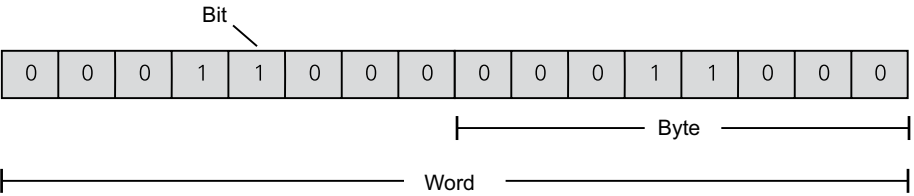
The process of converting a binary number to an equal decimal value is as simple as adding the equivalent decimal value for each position in the binary number where a 1 is shown. Positions with a 0 do not add to the number value.

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1
0	0	1	0	1	0	0	1

Decimal Value =  $32 + 8 + 1 = 41$

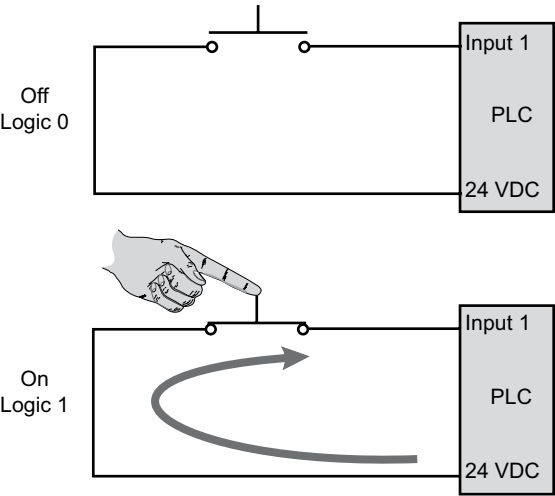
**Bits, Bytes, and Words**

Each position in a binary number is called a **bit**. The number of bits used to represent numbers varies with the device. However, instructions and data are usually grouped in bytes and eight bits make up one **byte**. Two bytes, or 16 bits, make up one **word**.



**Logic 0, Logic 1**

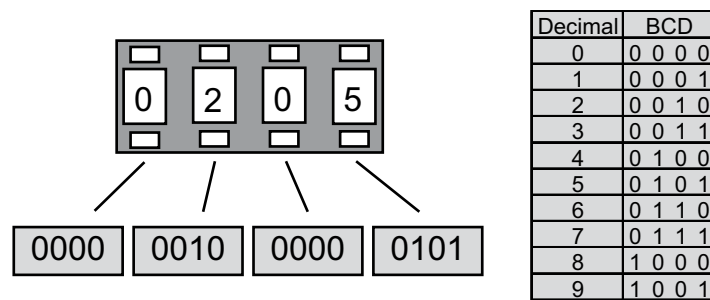
While PLCs are capable of sensing and generating analog values, programmable controllers internally use signals that are on or off. These on and off conditions correspond to the binary values 1 and 0. For example, a binary 0, also called logic 0, can be used to indicate that a switch is off, and a binary 1 (logic 1) can be used to indicate that a switch is on.



## BCD

While it is necessary for PLCs to use binary values, humans often need to see values represented in decimal. As a result, some input and output devices provide a decimal display where each decimal digit corresponds to four PLC binary inputs or outputs. The most common system used by input and output devices of this type is referred to as **binary-coded decimal (BCD)**.

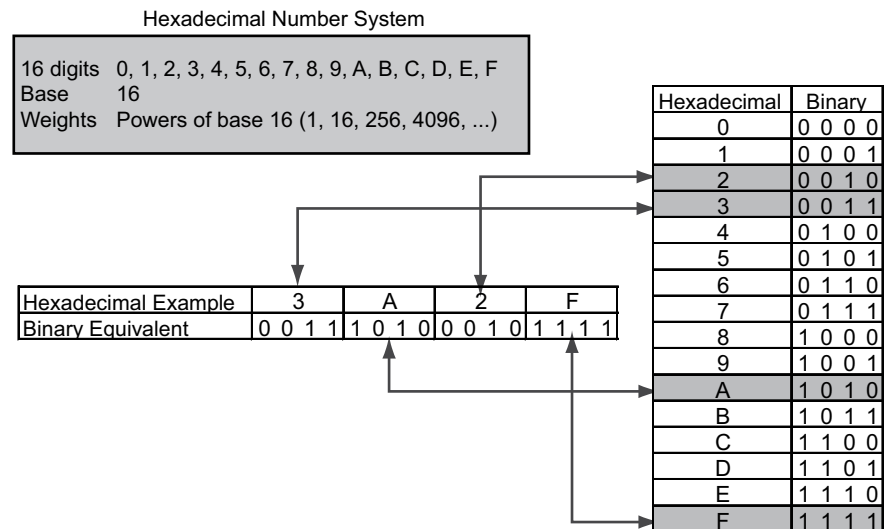
One example of a BCD device is a type of four-digit thumbwheel switch. Each thumbwheel digit controls four PLC inputs. This means that for a four-digit thumbwheel, 16 inputs are required. Because each thumbwheel digit only needs to represent decimal values from 0 through 9, only ten corresponding binary values are required for each digit.



## Hexadecimal

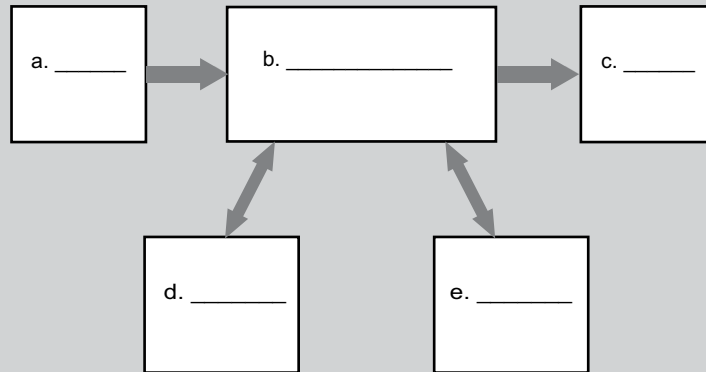
**Hexadecimal** is another system used in PLCs. The ten digits of the decimal system are used for the first ten characters of the hexadecimal system. The first six letters of the alphabet are used for the remaining six characters.

The hexadecimal system is used in PLCs because it allows the status of a large number of binary bits to be represented in a small space such as on a computer screen or programming device display. Each hexadecimal character represents the exact status of four binary bits.



## Review 1

1. Identify each of the following blocks in a basic PLC system:



2. The base of the binary number system is \_\_\_\_ .
3. The base of the hexadecimal number system is \_\_\_\_.
4. Convert the decimal number 10 to each of the following number types:

Binary \_\_\_\_\_

BCD \_\_\_\_\_

Hexadecimal \_\_\_\_\_

# Terminology

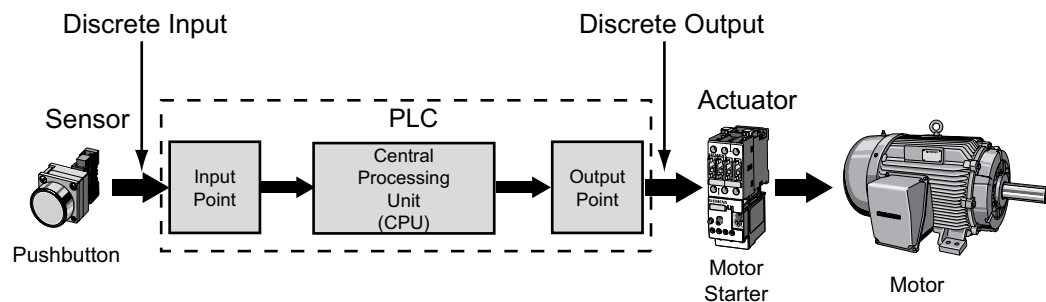
Developing an understanding of PLCs requires learning some basic terminology. This section provides an overview of commonly used PLC terms, beginning with the terms sensor and actuator.

## Sensors

**Sensors** are devices that convert a physical condition into an electrical signal for use by a controller, such as a PLC. Sensors are connected to the input of a PLC. A pushbutton is one example of a sensor that is often connected to a PLC input. An electrical signal indicating the condition (open or closed) of the pushbutton contacts is sent from the pushbutton to the PLC.

## Actuators

**Actuators** are devices that convert an electrical signal from a controller, such as a PLC, into a physical condition. Actuators are connected to the PLC output. A motor starter is one example of an actuator that is often connected to a PLC output. Depending on the status of the PLC output, the motor starter either provides power to the motor or prevents power from flowing to the motor.



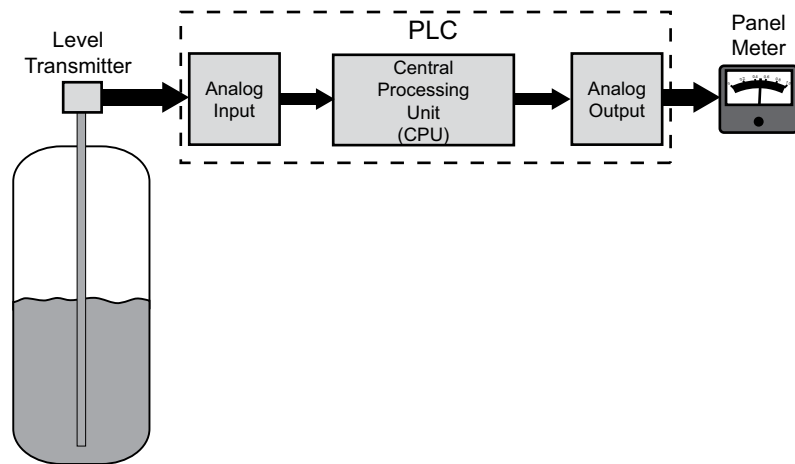
**Discrete Inputs and Outputs** **Discrete inputs and outputs**, also referred to as **digital inputs** and **outputs**, are either on or off. Pushbuttons, toggle switches, limit switches, proximity switches, and relay contacts are examples of devices often connected to PLC discrete inputs. Solenoids, relay and contactor coils, and indicator lamps are examples of devices often connected to PLC discrete outputs.

In the on condition, a discrete input or output is represented internal to the PLC as a logic 1. In the off condition, a discrete input or output is represented as a logic 0.

## Analog Inputs and Outputs

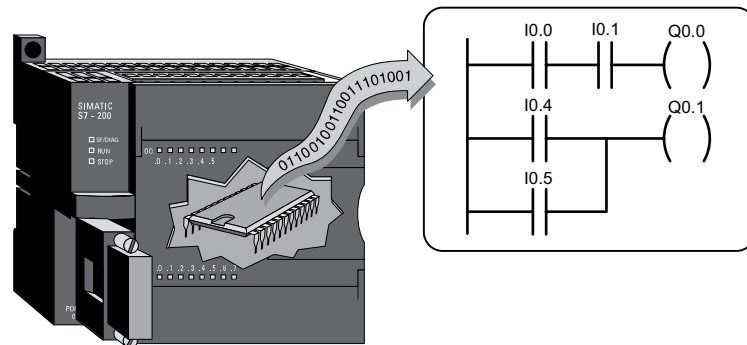
**Analog inputs and outputs** are continuous, variable signals. Typical analog signals vary from 0 to 20 milliamps, 4 to 20 milliamps, or 0 to 10 volts.

In the following example, a level transmitter monitors the level of liquid in a storage tank and sends an analog signal to a PLC input. An analog output from the PLC sends an analog signal to a panel meter calibrated to show the level of liquid in the tank. Two other analog outputs, not shown here, are connected to current-to-pneumatic transducers that control air-operated flow-control valves. This allows the PLC to automatically control the flow of liquid into and out of the storage tank.



## CPU

The **central processor unit (CPU)** is a microprocessor system that contains the system memory and is the PLC's decision-making unit. The CPU monitors inputs, outputs, and other variables and makes decisions based on instructions held in its program memory.

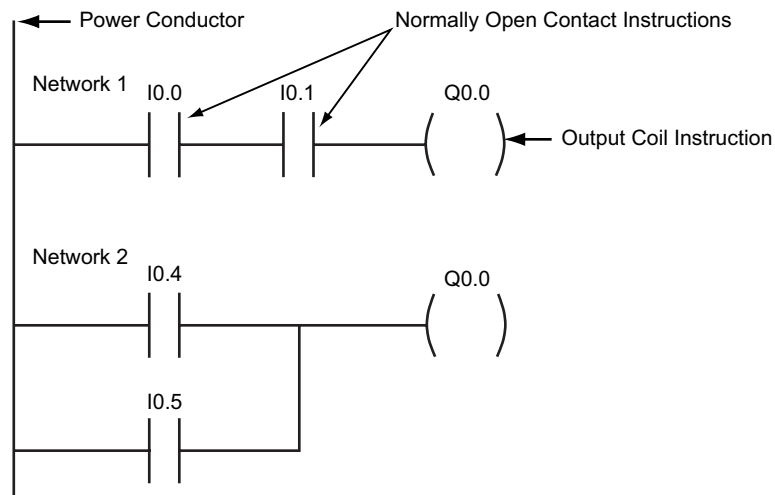


## Ladder Logic Programming

A program consists of instructions that accomplish specific tasks. The degree of complexity of a PLC program depends upon the complexity of the application, the number and type of input and output devices, and the types of instructions used.

**Ladder logic (LAD)** is one programming language used with PLCs. Ladder logic incorporates programming functions that are graphically displayed to resemble symbols used in hard-wired control diagrams.

The left vertical line of a ladder logic diagram represents the power or energized conductor. The output coil instruction represents the neutral or return path of the circuit. The right vertical line, which represents the return path on a hard-wired control line diagram, is omitted. Ladder logic diagrams are read from left-to-right and top-to-bottom. Rungs are sometimes referred to as networks. A network may have several control elements, but only one output coil.



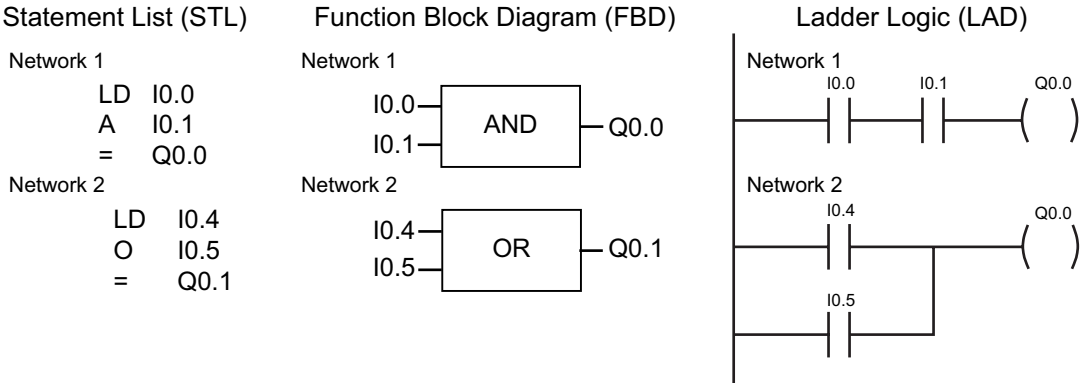
## Statement List and Function Block Diagrams

While ladder logic programs are still common, there are many other ways to program PLCs. Two other common examples are statement list and function block diagrams.

**Statement list (STL)** instructions include an operation and an operand. The operation to be performed is shown on the left. The operand, the item to be operated on, is shown on the right.

**Function block diagrams (FBD)** include rectangular functions with inputs shown on the left side of the rectangle and outputs shown on the right side.

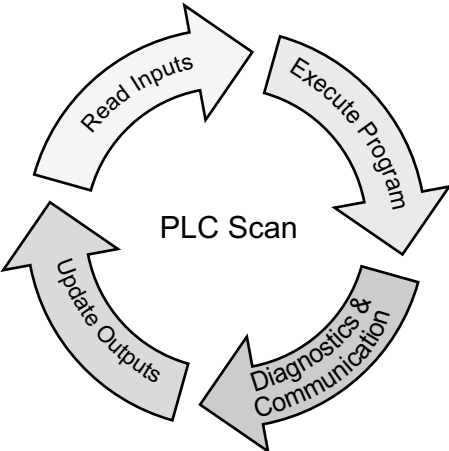
In the following example, the program segments perform the same function.



In addition to LAD, STL, and FBD, multiple other types of programming languages are used for PLCs. Each type of programming has its advantages and disadvantages. Factors such as application complexity, types of programming available for a specific PLC model, and user standards and preferences determine which type of programming is used for an application.

**PLC Scan**

The PLC program is executed as part of a repetitive process referred to as a **scan**. A PLC scan starts with the CPU reading the status of inputs. Next, the application program is executed. Then, the CPU performs internal diagnostics and communication tasks. Finally, the CPU updates the status of outputs. This process repeats as long as the CPU in the run mode. The time required to complete a scan depends on the size of the program, the number of I/Os, and the amount of communication required.



## Memory Types and Size

**Kilo**, abbreviated **k**, normally refers to 1000 units. When talking about computer or PLC memory, however, 1k means 1024. This is because of the binary number system ( $2^{10}=1024$ ). 1k can refer to 1024 bits, bytes, or words, depending the context.

**Random Access Memory (RAM)** is memory that allows data to be written to and read from any address (location). RAM is used as a temporary storage area. RAM is volatile, meaning that the data stored in RAM will be lost if power is lost. A battery backup is required to avoid losing data in the event of a power loss.

**Read Only Memory (ROM)** is a type of memory used where it is necessary to protect data or programs from accidental erasure. The original data stored in ROM can be read, but not changed. In addition, ROM memory is **nonvolatile**. This means that information will not be lost as the result of a loss of electrical power. ROM is normally used to store the programs that define the capabilities of the PLC.

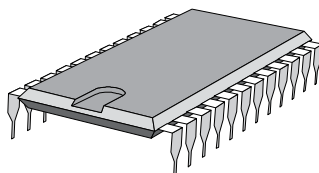
**Erasable Programmable Read Only Memory (EPROM)** provides a level of security against unauthorized or unwanted changes in a program. EPROMs are designed so that data stored in them can be read, but not easily altered. Changing EPROM data requires a special effort. UVEPROMs (ultraviolet erasable programmable read only memory) can only be erased with an ultraviolet light. EEPROM (electronically erasable programmable read only memory), can only be erased electronically.

## Software, Hardware, and Firmware

**Software** is the name given to computer instructions, regardless of the programming language. Essentially, software includes the instructions or programs that direct hardware.

**Hardware** is the name given to all the physical components of a system. The PLC, the programming device, and the connecting cable are examples of hardware.

**Firmware** is user or application specific software burned into EPROM and delivered as part of the hardware. Firmware gives the PLC its basic functionality.



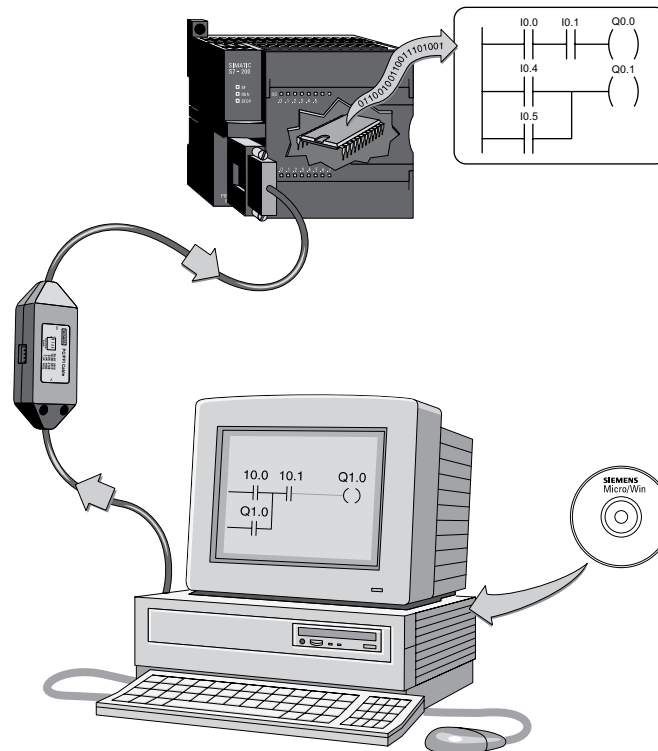


## Putting it Together

The user memory of a PLC, such as the S7-200 PLC shown in the following illustration, includes space for the user program as well as addressable memory locations for storage of data. The amount of program and data space available depends on the CPU model.

User program space stores instructions that are executed repetitively as part of the PLC scan. The user program is developed using a programming device, such as a personal computer (PC) with programming software, then loaded into the user program memory of the PLC.

A variety of addressable memory locations are used for storage of data that is available to the user program. Among other things, this includes memory locations for variable data, discrete inputs and outputs, analog inputs and outputs, timers, counters, high-speed counters, etc.

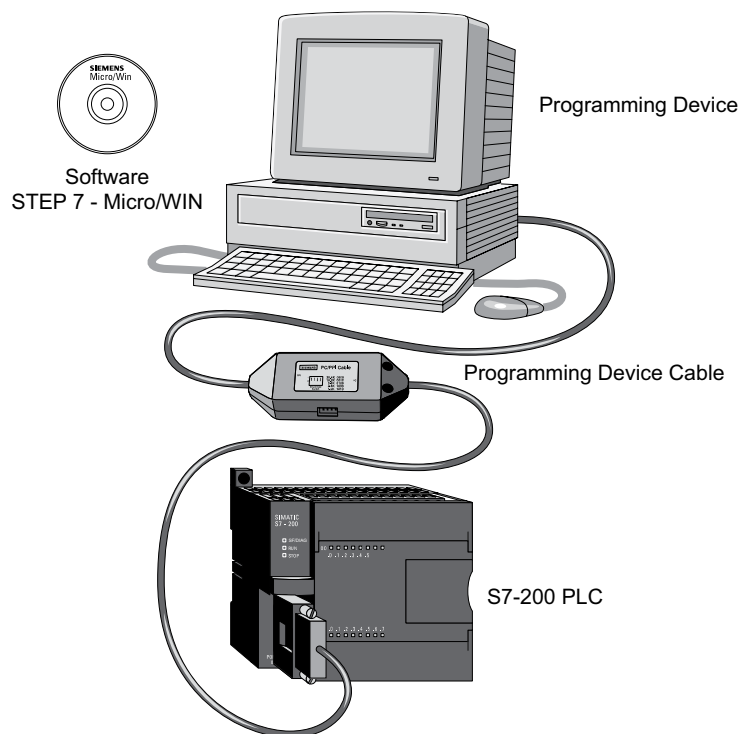


# Basic Requirements

Throughout this course we will be using the S7-200 PLC for specific examples of PLC concepts. The S7-200 PLC is used for this purpose because of its ease of use and wide-spread application.

The items shown in the following illustration are needed to create or change an S7-200 PLC program. The program is created using **STEP 7-Micro/WIN** programming software, which runs on a Windows-based personal computer (Win2000, Windows XP, and higher operating system).

A special cable is needed when a personal computer is used as a programming device. Two versions of this cable are available. One version, called an **RS-232/PPI Multi-Master Cable**, connects a personal computer's RS-232 interface to the PLC's RS-485 connector. The other version, called a **USB/PPI Multi-Master Cable**, connects a personal computer's USB interface to the PLC's RS-485 connector.



## Review 2

1. Pushbuttons, limit switches, and relay contacts are examples of devices that may be connected to PLC \_\_\_\_\_ inputs.
2. Solenoids, relay and contactor coils, and indicator lamps are examples devices that may be connected to PLC \_\_\_\_\_ outputs.
3. The \_\_\_\_\_ contains the system memory and makes decisions based on instructions stored in program memory.
4. \_\_\_\_\_ is a PLC programming language that incorporates programming functions that are graphically displayed to resemble symbols used in hard-wired control diagrams.
5. \_\_\_\_\_ and \_\_\_\_\_ are also common examples of ways to program a PLC.
6. A PLC program is executed as part of a repetitive process referred to as a \_\_\_\_\_.
7. When talking about computer or PLC memory, 1k refers to \_\_\_\_\_ bits, bytes, or words.
8. Software that is burned into EPROM is called \_\_\_\_\_.
9. An RS-232/PPI Multi-Master cable or a USB/PPI-Multi-Master cable may be used to connect a personal computer to an S7-200 PLC's \_\_\_\_\_ connector.

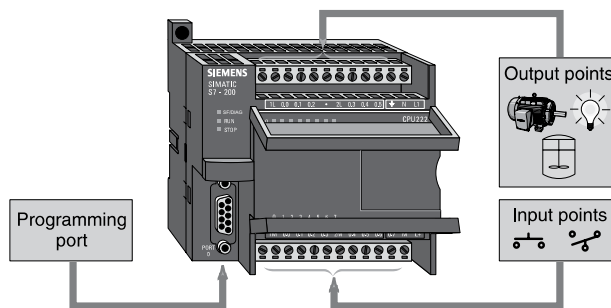
# S7-200 Micro PLCs

The S7-200 Micro PLC is the smallest member of the SIMATIC S7 family of programmable controllers.

Each S7-200 central processing unit (CPU) model also includes input and output points in the same housing as the CPU.

Inputs and outputs (I/O) are the system control points. Inputs monitor field devices, such as switches and analog sensors. Outputs control other devices, such as motors and control valves.

The programming port is the connection to the programming device and also provides a means for connecting the PLC to other devices, such as display panels.



## S7-200 Models

There are six S7-200 CPU types (**CPU 221**, **CPU 222**, **CPU 224**, **CPU 224XP**, **CPU 224XPsi**, and **CPU 226**) and two power supply configurations for each type.

Model Description	Power Supply	Input Types	Output Types	Comm Ports
221 DC/DC/DC	20.4-28.8 VDC	6 x 24 VDC	4 x 24 VDC	1
221 AC/DC/Relay	85-264 VAC, 47-63 Hz	6 x 24 VDC	4 x Relay	1
222 DC/DC/DC	20.4-28.8 VDC	8 x 24 VDC	6 x 24 VDC	1
222 AC/DC/Relay	85-264 VAC, 47-63 Hz	8 x 24 VDC	6 x Relay	1
224 DC/DC/DC	20.4-28.8 VDC	14 x 24 VDC	10 x 24 VDC	1
224 AC/DC/Relay	85-264 VAC, 47-63 Hz	14 x 24 VDC	10 x Relay	1
224XP DC/DC/DC	20.4-28.8 VDC	14 x 24 VDC, 2 x Analog	10 x 24 VDC, 1 x Analog	2
224XP AC/DC/Relay	85-264 VAC, 47-63 Hz	14 x 24 VDC, 2 x Analog	10 x Relay, 1 x Analog	2
224XPsi DC/DC/DC	20.4-28.8 VDC	14 x 24 VDC, 2 x Analog	10 x 24 VDC (current sinking), 1 x Analog	2
226 DC/DC/DC	20.4-28.8 VDC	24 x 24 VDC	16 x 24DC	2
226 AC/DC/Relay	85-264 VAC, 47-63 Hz	24 x 24 VDC	16 x Relay	2

In the model description, the first term following the CPU type indicates the power supply type, the second term indicates the input type, and the third term indicates the output type. For example, a 222 AC/DC/Relay model is powered from an AC source, has DC input points, and relay contact output points.

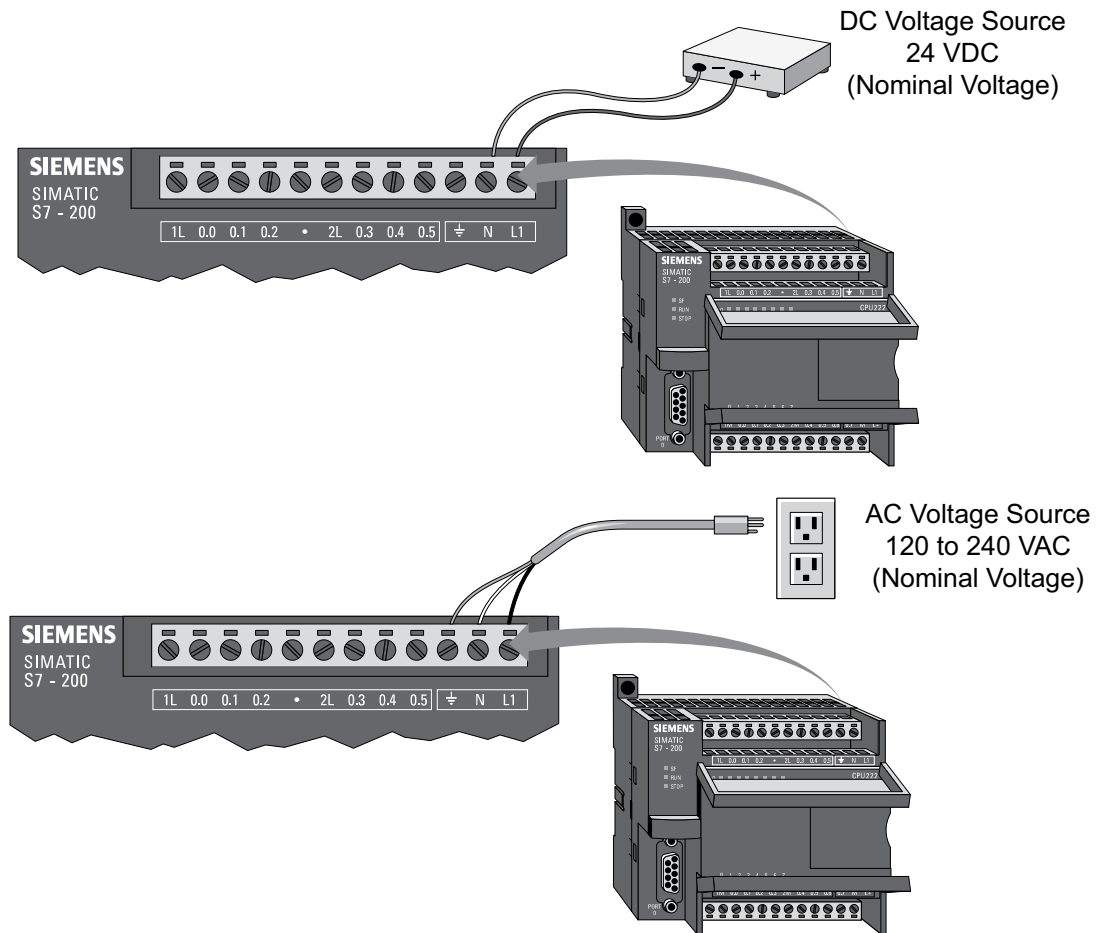
## S7-200 Features

The S7-200 family includes a range of CPUs which provide a variety of features to aid in designing a cost-effective automation solution. The accompanying table provides a summary of the major features, many of which are covered in this course. Note that the CPU 224XPsi has 10 current sinking digital outputs, but its other features are the same as for the CPU 224XP.

Feature	CPU 221	CPU 222	CPU 224	CPU 224XP CPU 224XPsi	CPU 226
<b>Memory</b>					
Program (with run mode edit)	4096 Bytes	4096 Bytes	8192 Bytes	12288 Bytes	16384 Bytes
Program (w/o run mode edit)	4096 Bytes	4096 Bytes	12288 Bytes	16384 Bytes	24576 Bytes
User Data	2048 Bytes	2048 Bytes	8192 Bytes	10240 Bytes	10240 Bytes
Optional Memory Cartridges	64k or 256k Bytes	64k or 256k Bytes	64k or 256k Bytes	64k or 256k Bytes	64k or 256k Bytes
Memory Backup (super cap)	50 Hours typical	50 Hours typical	100 Hours typical	100 Hours typical	100 Hours typical
Memory Backup (opt. battery)	200 Days typical	200 Days typical	200 Days typical	200 Days typical	200 Days typical
<b>I/O</b>					
Digital I/O without Exp. Modules	6 In/4 Out	8 In/6 Out	14 In/10 Out	14 In/10 Out	24 In/16 Out
Analog I/O without Exp. Modules	None	None	None	2 In/1 Out	None
Max Expansion Modules	None	2	7	7	7
<b>Instructions</b>					
Internal Relays	256	256	256	256	256
Counters	256	256	256	256	256
Timers	256	256	256	256	256
32-Bit, Floating-Point Math (+-*/)	Yes	Yes	Yes	Yes	Yes
<b>Enhanced Features</b>					
High-Speed Counters	4 (30 KHz)	4 (30 KHz)	6 (30 KHz)	4 (30 KHz). 2 (200 KHz)	6 (30 KHz)
Pulse Outputs (DC)	2 (20 KHz)	2 (20 KHz)	2 (20 KHz)	2 (20 KHz)	2 (20 KHz)
Timed Interrupts	2 (1ms - 255ms)	2 (1ms - 255ms)	2 (1ms - 255ms)	2 (1ms - 255ms)	2 (1ms - 255ms)
Edge Interrupts	4	4	4	4	4
Real-Time Clock	Optional	Optional	Built-In	Built-In	Built-In
Password Protection	Yes	Yes	Yes	Yes	Yes
<b>Communications</b>					
Number of Ports	1 (RS-485)	1 (RS-485)	1 (RS-485)	2 (RS-485)	2 (RS-485)
Protocols Supported Port 0	PPI, MPI Slave, Freeport	PPI, MPI Slave, Freeport	PPI, MPI Slave, Freeport	PPI, MPI Slave, Freeport	PPI, MPI Slave, Freeport
Optional Communications	Not Expandable	PROFIBUS DP Slave, AS-Interface Master, Ethernet, Internet, Modem	PROFIBUS DP Slave, AS-Interface Master, Ethernet, Internet, Modem	PROFIBUS DP Slave, AS-Interface Master, Ethernet, Internet, Modem	PROFIBUS DP Slave, AS-Interface Master, Ethernet, Internet, Modem

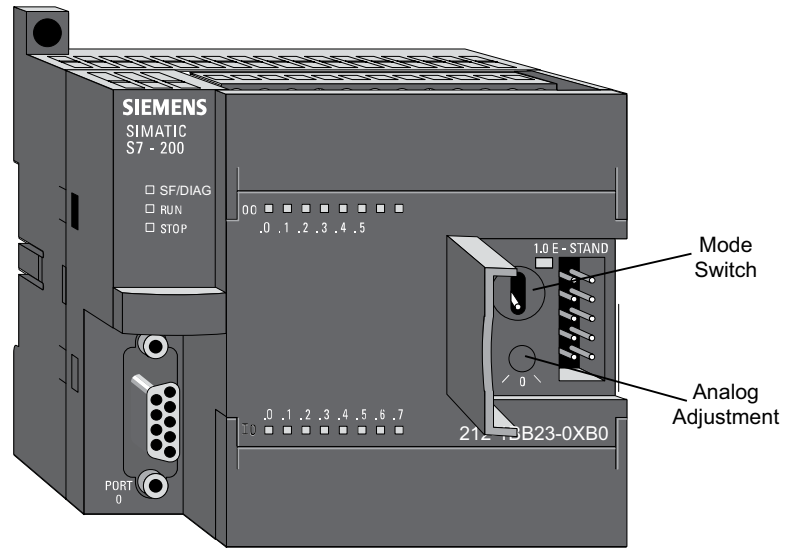
## Power Sources

Depending on the CPU model, an S7-200 CPU is powered from either a 24 VDC or a 120 to 240 VAC power supply. For example, an CPU 221 DC/DC/DC model is powered from a 24 VDC power supply and a CPU 222 AC/DC/Relay model is powered from a 120 or 240 VAC power supply.



## Mode Switch and Analog Adjustment

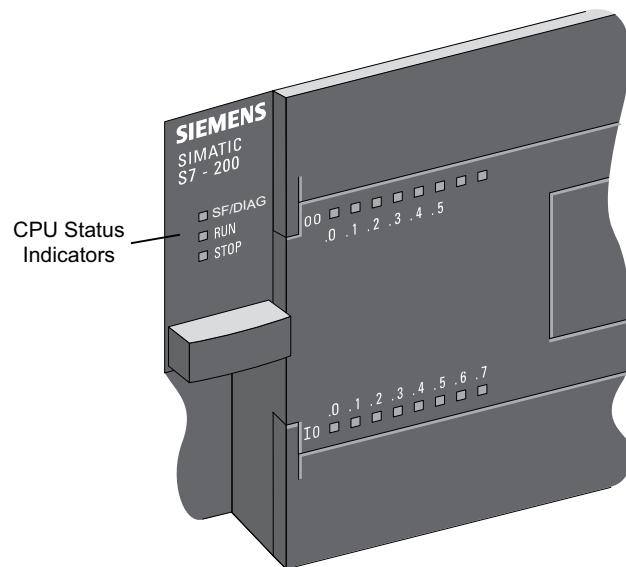
Each S7-200 CPU has a **mode switch** with three positions, **RUN**, **STOP**, and **TERM**. When the mode switch is in the RUN position, the CPU is in the RUN mode and executing the program, unless a fault has occurred. When the mode switch is in the STOP position, the CPU is in the STOP mode and not executing the user program. When the mode switch is in the TERM position, the programming device can select the operating mode.



An **analog adjustment** is available to increase or decrease values stored in special memory. This can allow a variable in the user program to change as the analog adjustment is changed. CPU 21 and CPU 22 models have one analog adjustment. CPU 224, CPU 224XP, CPU 224 XPsi, and CPU 226 have two analog adjustments.

## CPU Status Indicators

The **CPU status indicators** display the current CPU mode. When the CPU is in the RUN mode, the green **RUN indicator** is lit. When the CPU is in the STOP mode, the yellow **STOP indicator** is lit. The **System Fault/Diagnostic (SF/DIAG) indicator** turns red for a system fault and yellow to indicate certain diagnostic conditions.

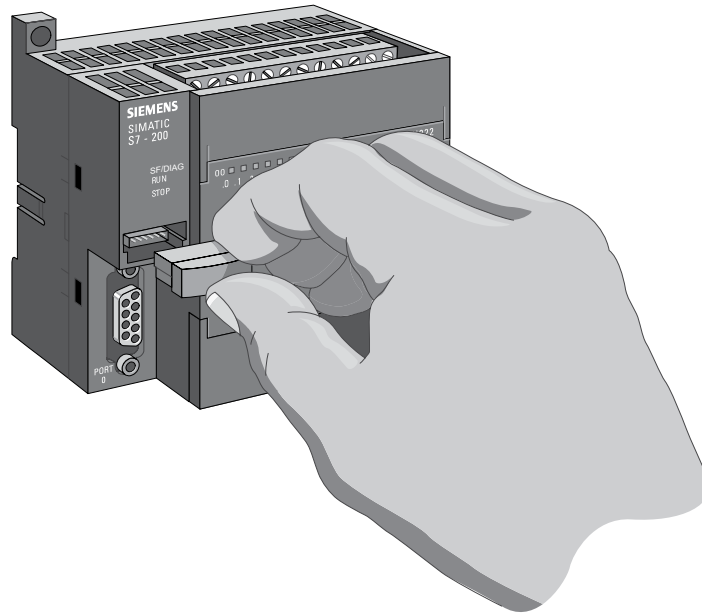


The **I/O status indicators** represent the on or off status of corresponding inputs and outputs. For example, when the CPU senses an input is on, the corresponding green indicator is lit.

## Optional Cartridges

S7-200 CPUs support an optional **memory cartridge** that provides portable EEPROM storage for the user program. The cartridge can be used to copy a program from one S7-200 PLC to a like S7-200 PLC. Two memory cartridge sizes are available, 64k and 256k bytes.

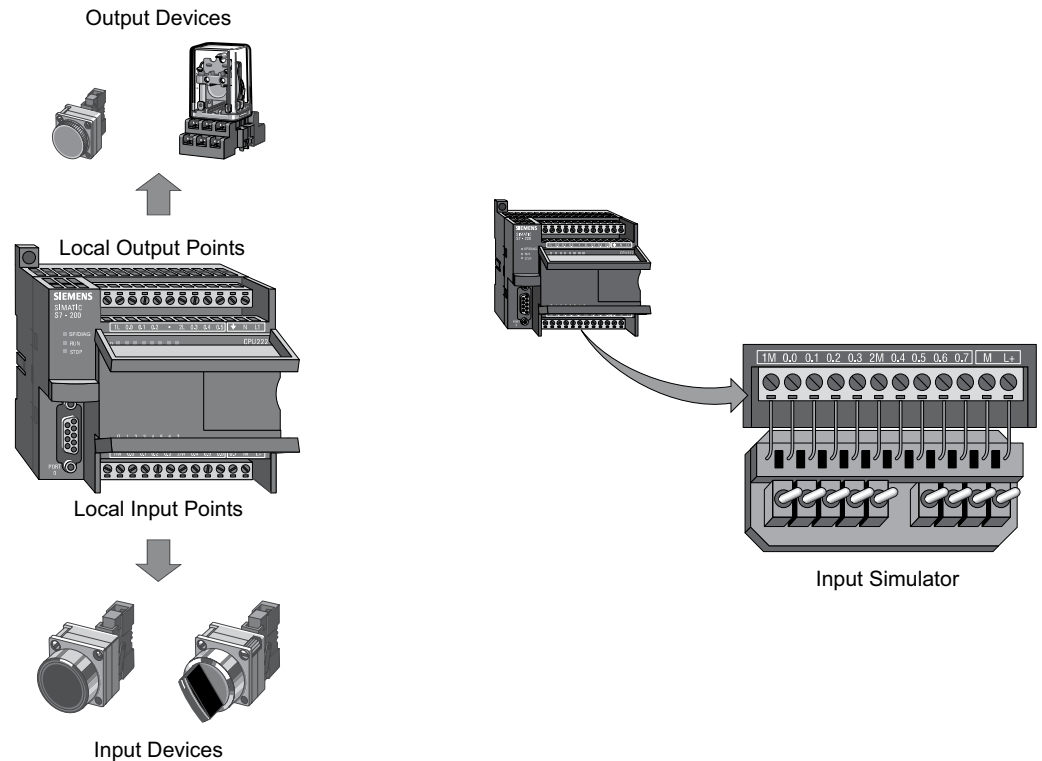
Two other cartridges are also available. A **real-time clock with battery** is available for use on the CPU 221 and CPU 222. (CPU 224, CPU 224XP, CPU 224XPsi, and CPU 226 have a real-time clock built in.) The battery provides up to 200 days of data retention time in the event of a power loss. Another cartridge is available with a **back-up battery only**.





## Inputs and Outputs

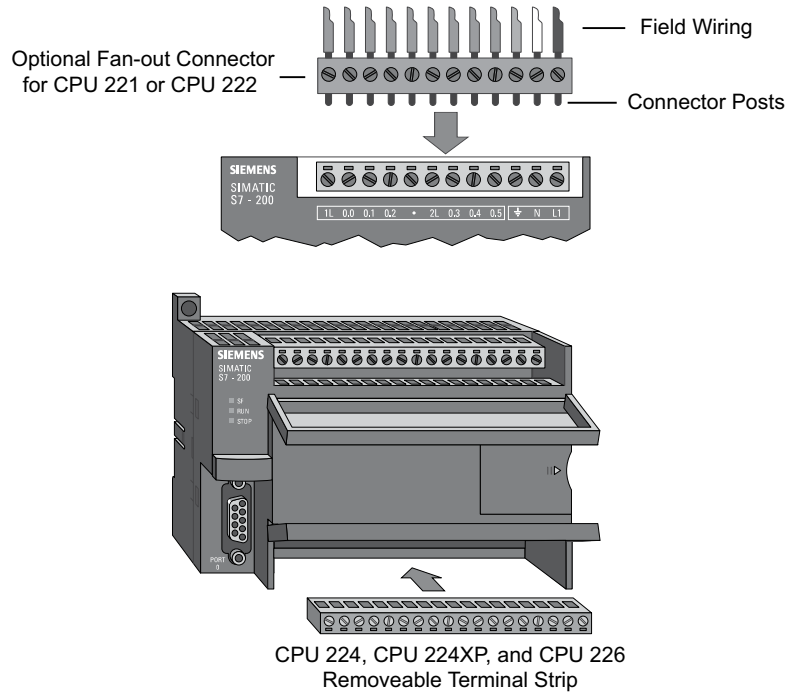
Input devices, such as switches, pushbuttons, and other sensors are connected to the terminal strip under the bottom cover of the PLC.



A convenient method of testing a program is to wire toggle switches to the inputs. **Input simulators** with pre-wired toggle switches are available for use with S7-200 PLCs. Switches are wired between the 24 VDC power supply (L+) and the inputs. For example, the switch on the far left is wired between the first input (0.0) and L+. When the switch is closed, 24 VDC is applied to the input. When the switch is open, 0 VDC is applied to the input.

Output devices, such as relays, are connected to the terminal strip under the top cover of the PLC. When testing a program, it is not necessary to connect output devices. The LED status indicators signal if an output is active.

An optional **fan-out connector** allows field wiring connections to remain fixed when removing or replacing a CPU 221 or CPU 222. The appropriate connector slides into either the input, output, or expansion module terminals.

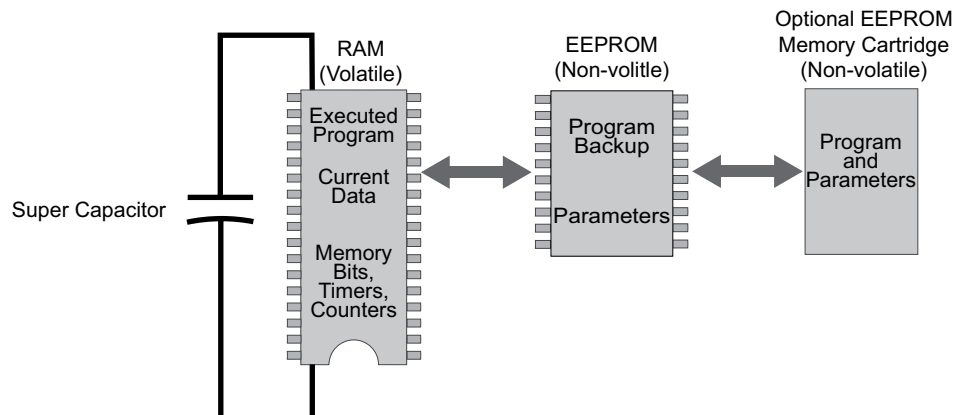


CPU 224, CPU 224XP, CPU 224Xpsi, and CPU 226 do not have an optional fan-out connector. Instead, their terminal strips are removable.

## Super Capacitor

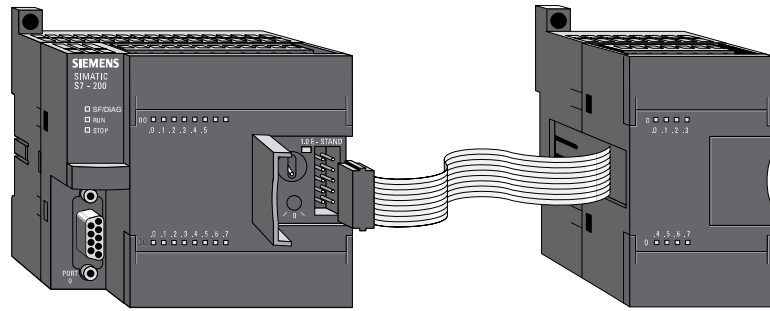
A **super capacitor**, so named because of its ability to maintain a charge for a long period of time, protects data stored in RAM in the event of a power loss.

The RAM memory is typically backed up for 50 hours on the CPU 221 and CPU 222 and for 100 hours on the CPU 224, CPU 224 XP, CPU 224 Xpsi, and CPU 226.

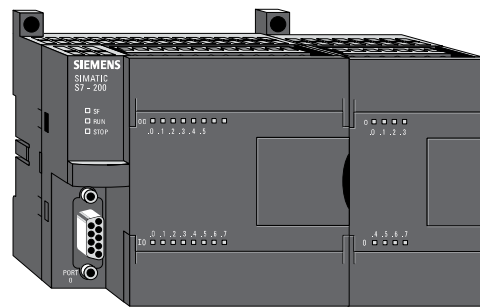
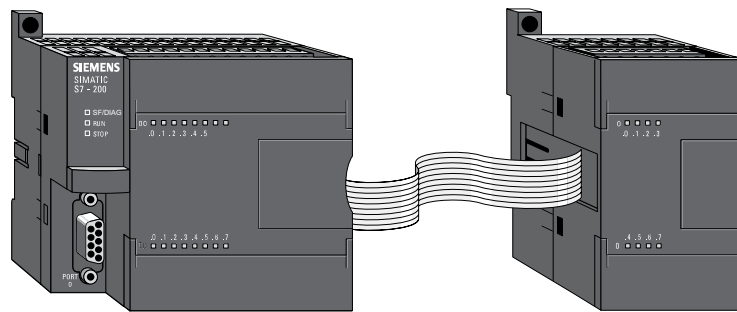


## Expansion Modules

S7-200 PLCs are expandable by adding expansion modules. **Expansion modules** with inputs and/or outputs are connected to the base unit using a ribbon connector.

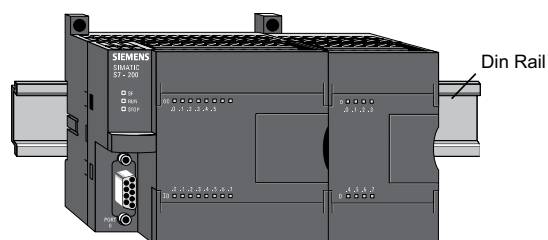


The ribbon connector is protected by a cover on the base unit. Side-by-side mounting completely encloses and protects the ribbon connector.



## Mounting

S7-200 PLCs can be mounted in one of two ways. A DIN clip allows installation on a standard **DIN rail**. The DIN clip snaps open to allow installation and snaps closed to secure the unit on the rail. The S7-200 can also be panel mounted using installation holes located behind the access covers.



## Available Expansion

Most S7-200 expansion modules are designed to provide additional I/O. However, several expansion modules are available to support communication options, positioning, and weighing (SIWAREX MS).

CPU 221 comes with 6 discrete inputs and 4 discrete outputs and does not accept expansion modules.

CPU 222 comes with 8 discrete inputs and 6 discrete outputs and accepts up to 2 expansion modules.

CPU 224, CPU 224XP, and CPU 224XPsi come with 14 discrete inputs and 10 discrete outputs and accept up to 7 expansion modules. Note: The digital outputs for the CPU 224XPsi are current sinking.

CPU 226 comes with 24 discrete inputs and 16 discrete outputs and accepts up to 7 expansion modules.

CPU221								6 Inputs, 4 Outputs No Expansion Modules (EM)
CPU222	EM	EM						8 Inputs, 6 Outputs Up to 2 Expansion Modules
CPU224	EM	EM	EM	EM	EM	EM	EM	14 Inputs, 10 Outputs Up to 7 Expansion Modules
CPU224XP CPU224XPsi	EM	EM	EM	EM	EM	EM	EM	14 Inputs, 10 Outputs 2 Analog In, 1 Analog Out Up to 7 Expansion Modules
CPU226	EM	EM	EM	EM	EM	EM	EM	24 Inputs, 16 Outputs Up to 7 Expansion Modules

Expansion Modules					
Discrete Input	8 x 24 VDC	8 x 120/230 VAC	16 x 24 VDC		
Discrete Output	4 x 24 VDC	8 x 24 VDC	8 x 120/230 VAC	4 x Relay	8 x Relay
Discrete Combination	4 x 24 VDC In/4 x 24 VDC out		8 x 24 VDC In/8 x 24 VDC Out		
	16 x 24 VDC In/16 x 24 VDC Out		32 x 24 VDC In/32 x 24 VDC O		
	4 x 24 VDC In/4 x Relay		8 x 24 VDC In/8 x Relay		
	16 x 24 VDC In/16 x Relay		32 x 24 VDC In/32 x Relay		
Analog Input	4 x Analog	4 x Thermocouple	2 x RTD		
	8 x Analog	8 x Thermocouple	4 x RTD		
Analog Output	2 x Analog	4 x Analog			
Analog Combination	4 x Analog In/1x Analog Out				
Communication Modules	Modem	AS-Interface	PROFIBUS-DP	Ethernet	Ethernet IT
	GSM/GPRS Modem				
Other Modules	Position	SIWAREX MS			

## I/O Numbering

S7-200 inputs and outputs are labeled at the wiring terminations and next to the status indicators. These alphanumeric symbols identify the I/O address to which a device is connected. This address is used by the CPU to determine which input is present and which output needs to be turned on or off.

I designates a discrete input and Q designates a discrete output. The first number identifies the byte, the second number identifies the bit.

Image register space for digital I/O is always reserved in increments of eight bits (one byte). If a module does not provide a physical point for each bit of each reserved byte, these unused bits cannot be assigned to subsequent modules in the I/O chain.

Each analog I/O point is associated with a 16-bit word in the S7-200 PLC and is identified by AI (for analog input) or AQ (for analog output) followed by a W (representing a word of memory) and a starting byte number. Analog I/O words start on even-numbered bytes (such as 0, 2, or 4).

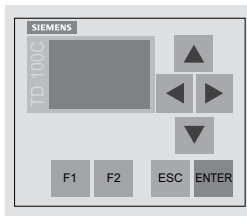
Analog I/O points are always allocated in increments of two points. If a module does not provide physical I/O for each of these points, these I/O points are lost and are not available for assignment to subsequent modules in the I/O chain.

The following example shows the addressing for one sample application.

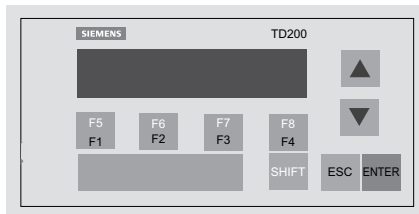
CPU 224XP		Module 0		Module 1	Module 2		Module 3	Module 4	
14 Discrete In	10 Discrete Out								
2 Analog In	1 Analog Out	4 Discrete In	4 Discrete Out	8 Discrete In	4 Analog In	1 Analog Out	8 Discrete Out	4 Analog In	1 Analog Out
I0.0	Q0.0	I2.0	Q2.0	I3.0	AIW 4	AQW 4	Q3.0	AIW 12	AQW 8
I0.1	Q0.1	I2.1	Q2.1	I3.1	AIW 6	AQW 6	Q3.1	AIW 14	AQW 10
I0.2	Q0.2	I2.2	Q2.2	I3.2	AIW 8		Q3.2	AIW 16	
I0.3	Q0.3	I2.3	Q2.3	I3.3	AIW 10		Q3.3	AIW 18	
I0.4	Q0.4	I2.4	Q2.4	I3.4			Q3.4		
I0.5	Q0.5	I2.5	Q2.5	I3.5			Q3.5		
I0.6	Q0.6	I2.6	Q2.6	I3.6			Q3.6		
I0.7	Q0.7	I2.7	Q2.7	I3.7			Q3.7		
I1.0	Q1.0	Expansion I/O							
I1.1	Q1.1	Addresses shown with a black background are not available and cannot be used in the program.							
I1.2	Q1.2								
I1.3	Q1.3								
I1.4	Q1.4								
I1.5	Q1.5								
I1.6	Q1.6								
I1.7	Q1.7								
AIW 0	AQW 0								
AIW 2	AQW 2								
Local I/O									

## SIMATIC Micro Panels

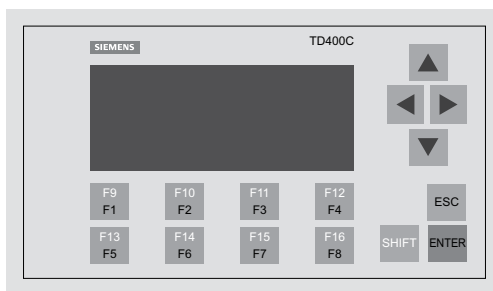
Siemens offers a variety of SIMATIC Micro Panels designed for use with S7-200 PLCs. These panels provide easy to implement solutions for a variety of display needs.



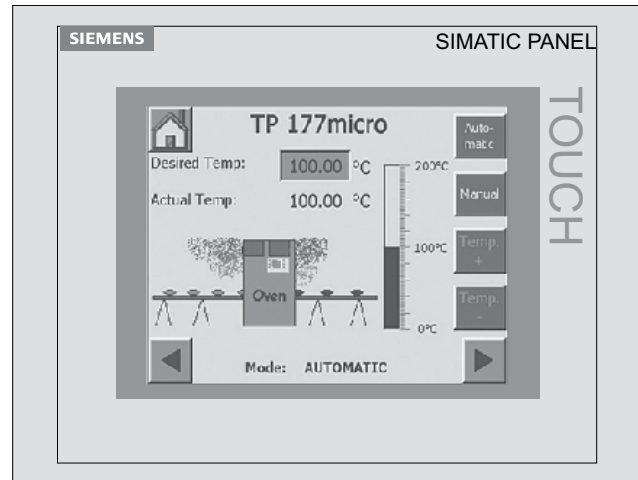
TD 100C



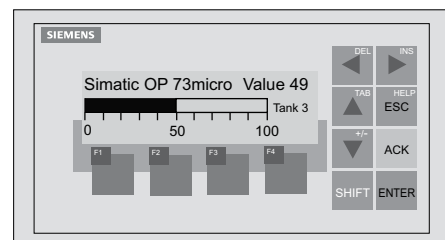
TD 200 and TD 200C



TD 400C



TP 177micro



OP 73micro

Text display **TD 100C** provides a 4-line display with up to 16 characters per line.

Text displays **TD 200** and **TD 200C** provide a back-lit, high-contrast liquid crystal 2-line display for up to 80 text messages with integrated variables. TD 200 provides 8 user-configurable function keys in a fixed arrangement. TD 200C provides up to 20 user-configurable keys in a user-defined layout.

Text Display **TD 400C** provides a back-lit, high-contrast liquid crystal 4-line display for up to 80 text messages with integrated variables.

Graphics operator panel **OP 73micro** provides a full graphics 3-inch display for bitmaps, bars, and text with different font sizes.

Touch Panel **TP 177micro** provides a 6-inch touch screen for vector graphics. The graphics on the screen can be set up for viewing with the panel mounted horizontally or vertically.

## Reference Manual

The **SIMATIC S7-200 Programmable Controller System Manual** provides complete information on installing and programming the S7-200 PLCs. This manual can be downloaded as a PDF file from the **Technical Info** link on the Siemens S7-S00 web site: [http://www.automation.siemens.com/\\_en/s7-200/index.htm](http://www.automation.siemens.com/_en/s7-200/index.htm)

<b>SIEMENS</b>	
	Preface, Contents
	Product Overview 1
	Getting Started 2
SIMATIC	Installing the S7-200 3
	PLC Concepts 4
S7-200	Programming Concepts, Conventions and Features 5
Programmable Controller	S7-200 Instruction Set 6
System Manual	Communicating over a Network 7
	Hardware Troubleshooting Guide and Software Debugging Tools 8
	Open Loop Motion Control with the S7-200 9
	Creating a Program for the Modem Module 10
	Using the USS Protocol Library to Control a MicroMaster Drive 11
	Using the Modbus Protocol Library 12
	Using Recipes 13
	Using Data Logs 14
	PID Auto-Tune and the PID Tuning Control Panel 15
	Appendices
	Index

### Review 3

1. The six models of S7-200 are \_\_\_\_\_ , \_\_\_\_\_ , \_\_\_\_\_ , \_\_\_\_\_ , \_\_\_\_\_ , and \_\_\_\_\_ .
2. Which of the following is not available for an CPU 221?
  - a. Mode Switch
  - b. Expansion Module
  - c. Programming Port
  - d. Status Indicators
3. A CPU 222 can have a maximum of \_\_\_ expansion modules and a CPU 224 can have a maximum of \_\_\_ expansion modules.
4. A CPU 222 DC/DC/DC has \_\_\_ DC inputs and \_\_\_ DC outputs without expansion modules.
5. A CPU 224 DC/DC/DC has \_\_\_ DC inputs and \_\_\_ DC outputs without expansion modules.
6. The fourth output of an S7-200 would be labeled \_\_\_\_\_ .
7. S7-200 can be panel mounted or installed on a \_\_\_\_\_ rail.



# Programming a PLC

## STEP 7-Micro/WIN32

**STEP 7-Micro/WIN** is the software used with the S7-200 PLC to create a user program. STEP 7-Micro/WIN programs consist of a number of instructions that must be arranged in a logical order to obtain the desired PLC operation.

STEP 7-MicroWIN programming software can be run off line or online. **Off-line programming** allows the user to edit the program and perform a number of maintenance tasks. The PLC does not need to be connected to the programming device in this mode.



**Online programming** requires the PLC to be connected to the programming device. In this mode, program changes are downloaded to the PLC. In addition, status of the input/output elements can be monitored. The CPU can be started, stopped, or reset.

S7-200 PLCs have two instruction sets, **SIMATIC** and **IEC 1131-3**. The SIMATIC instruction set was developed by Siemens prior to the adoption of the IEC 1131-3 standard. The IEC 1131-3 instruction set was adopted by the International Electrotechnical Commission (IEC) to provide a common approach for PLC programming. The IEC 1131-3 instruction set is often preferred by users who work with PLCs from multiple suppliers.

STEP 7-Micro/WIN has three editors for program development, one for each of the types of programming available, ladder logic (LAD), statement list (STL), and function block diagram (FBD). The STL editor is often preferred by experienced programmers because of the similarity of STL programs to assembly language computer programs. However, the STL editor can only be used with the SIMATIC instruction set. Both the LAD and FBD editors can be used with either instruction set. Throughout this course, although other instruction types will occasionally be shown, the emphasis will be on SIMATIC LAD instructions.

## Basic Ladder Logic Symbols

PLC ladder logic consists of a commonly used set of symbols that represent instructions. Understanding these basic symbols is essential to understanding PLC operation.

## Contacts

One of the most confusing aspects of PLC programming for first-time users is the relationship between the device that controls a status bit and the programming function that uses a status bit. Two of the most common programming functions are the **normally open (NO) contact** and the **normally closed (NC) contact**. Symbolically, power flows through these contacts when they are closed. The normally open contact (NO) is closed when the input or output status bit controlling the contact is 1. The normally closed contact (NC) is closed when the input or output status bit controlling the contact is 0.



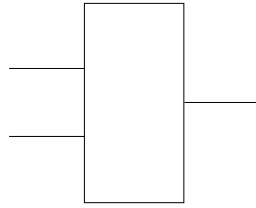
## Coils

**Coils** represent relays that are energized when power flows to them. When a coil is energized, it causes a corresponding output to turn on by changing the state of the status bit controlling that output to 1. That same output status bit may be used to control normally open and normally closed contacts elsewhere in the program.



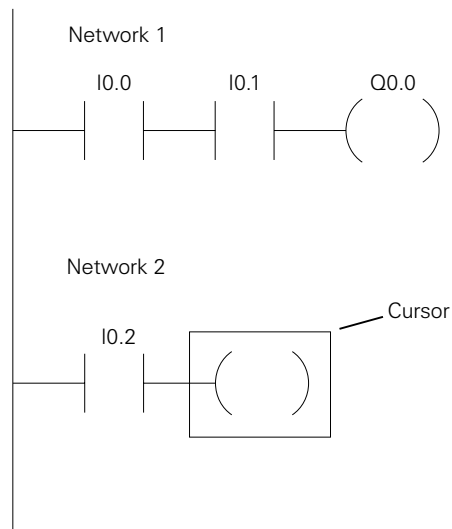
## Boxes

Boxes represent various instructions or functions that are executed when power flows to the box. Typical box functions include timers, counters, and math operations.



## Entering Elements

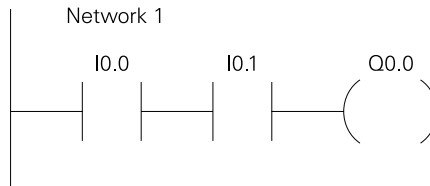
**Control elements** are entered in the ladder diagram by positioning the cursor and selecting the element from a list. In the following example the cursor has been placed in the position to the right of I0.2. A coil was selected from a pull-down list and inserted in this position.



## AND Operation

Each rung or network on a ladder represents a logic operation. The following programming example demonstrates an **AND operation**. Two contact closures and one output coil are placed on network 1. They are assigned addresses I0.0, I0.1, and Q0.0. Note that in the statement list a new logic operation always begins with a load instruction (LD). In this example I0.0 (input 1) and (A in the statement list) I0.1 (input 2) must be true in order for output Q0.0 (output 1) to be true. This same logic is also shown in a function block diagram.

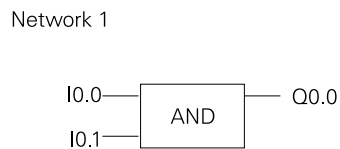
### Ladder Diagram Representation



### Statement List Representation

```
Network 1  
LD    I0.0  
A     I0.1  
=     Q0.0
```

### Function Block Diagram Representation



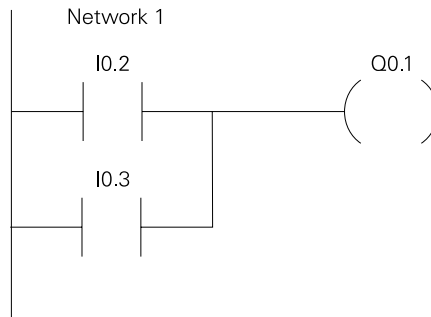
The following truth table represents the state of the output for each combination of input states.

I0.0	I0.1	Q0.0
0	0	0
0	1	0
1	0	0
1	1	1

## OR Operation

In this example, an **OR operation** is used in network 1. In the following example, if either input I0.2 (input 3) or (O in the statement list) input I0.3 (input 4), or both are true, then output Q0.1 (output 2) is true.

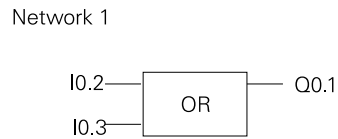
### Ladder Diagram Representation



### Statement List Representation

```
Network 1  
LD    I0.2  
O     I0.3  
=     Q0.1
```

### Function Block Diagram Representation



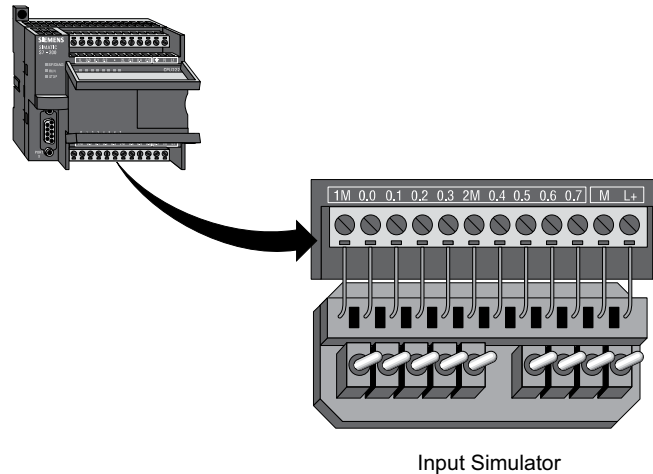
The following truth table represents the state of the output for each combination of input states.

I0.2	I0.3	Q0.1
0	0	0
0	1	1
1	0	1
1	1	1

## Testing a Program

Once a program has been written, it needs to be tested and debugged. One way this can be done is to simulate the field inputs with an input simulator, such as the one made for the S7-200 PLC.

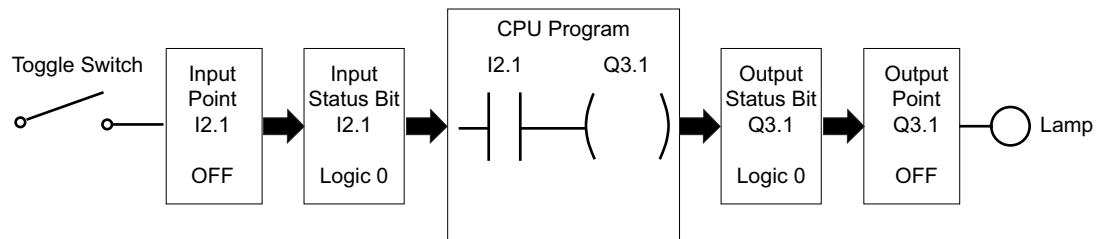
The program is first downloaded from the programming device to the CPU. The selector switch is placed in the RUN position. The simulator switches are operated and the resulting indication is observed on the output status indicator lamps.



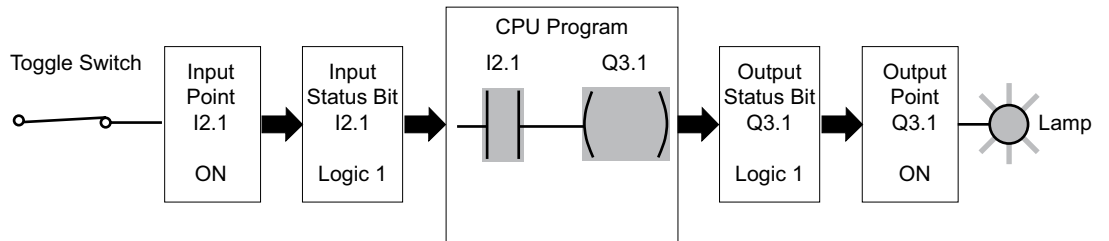
## Contact and Coil Status

After a program has been loaded and is running in the PLC, the actual status of ladder elements can be monitored using STEP 7 Micro/WIN software.

For example, in the following illustration, the toggle switch controls the status bit for I2.1. As long as the toggle switch is open, the I2.1 status bit is a logic 0. The I2.1 status bit controls the I2.1 normally open contact. Because the I2.1 status bit is a logic 0, the normally open contact function is open and no power is passed to the Q3.1 coil function. As a result, the Q3.1 status bit remains a logic 0 and output point Q3.1 is off.



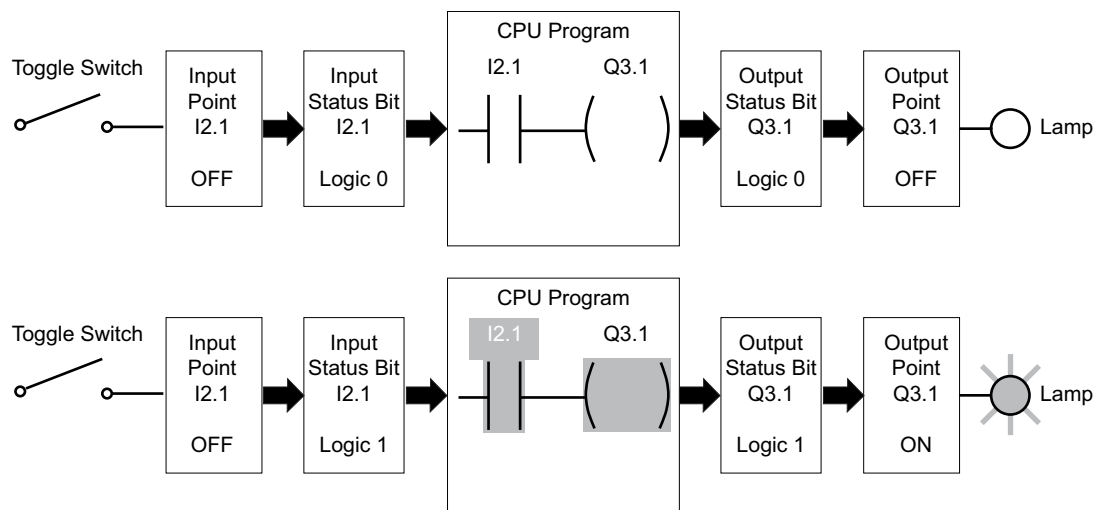
When the toggle switch closes, input point I2.1 turns on and I2.1 status bit changes to a logic 1. This causes normally open contact I2.1 to close and turn on Q3.1 coil. Note that a closed contact and a coil that is on are shown highlighted in the program. When Q3.1 coil turns on, the Q3.1 status bit goes to a logic 1 and output point Q3.1 turns on. This causes the lamp to light.



## Forcing









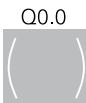
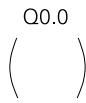


**Forcing** is another useful tool in the startup and maintenance of a PLC system. Forcing overrides one or more input or output status bits, causing them to stay in either a logic 0 or logic 1 status.

For example, in the following illustration, the toggle switch is open. Under normal circumstances, the toggle switch would have to be closed to turn on the lamp. However, if the I2.1 status bit is forced to a logic 1, the lamp will turn on, as long as the program is functioning correctly and there are no hardware or wiring problems. Similarly, the Q3.1 status bit could be forced to a logic 1 to turn on the lamp.



Forcing is useful not only to test and debug programs and hardware during startup, but also to troubleshoot systems with problems.

The following table shows the appearance of ladder diagram elements with associated status bits in the on, off, forced on, and forced off conditions.

	Status Bit On Contacts Closed	Status Bit Off Contacts Open	Status Bit Forced On Contacts Closed	Status Bit Forced Off Contacts Open
<b>Normally Open Contacts</b>				
<b>Normally Closed Contacts</b>				
<b>Output Coils</b>				

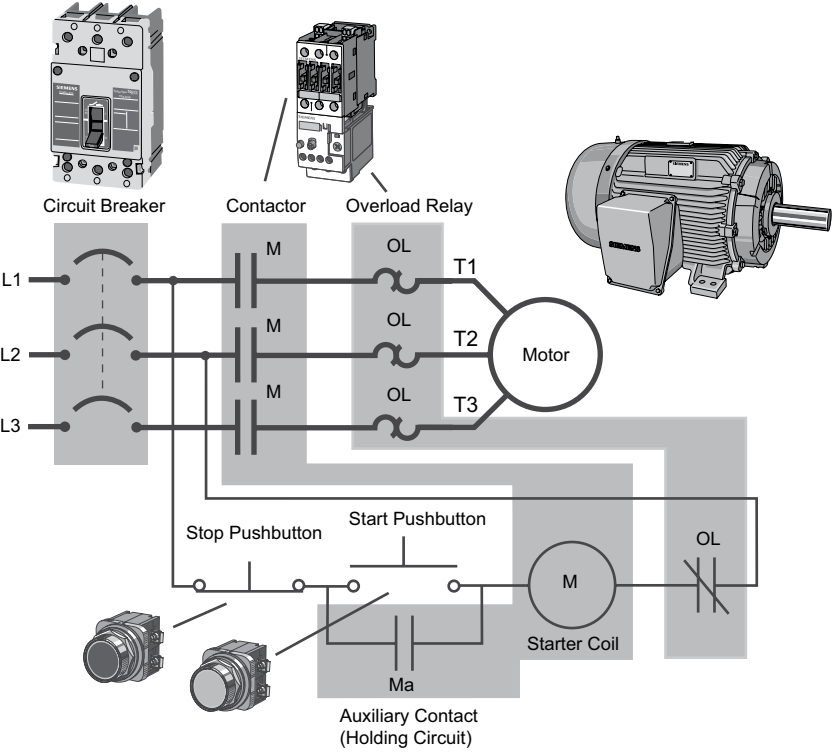


# Discrete Inputs/Outputs

## Motor Starter Example

While the lamp application previously discussed is useful to explain basic PLC operation, a more practical, and only slightly more complex, application is start-stop control of an AC motor. Before examining the PLC application, first consider a hard-wired approach.

The following line diagram illustrates how a normally open and a normally closed pushbutton might be connected to control a three-phase AC motor. In this example, a motor starter coil (M) is wired in series with a normally open, momentary Start pushbutton, a normally closed, momentary Stop pushbutton, and normally closed overload relay (OL) contacts.

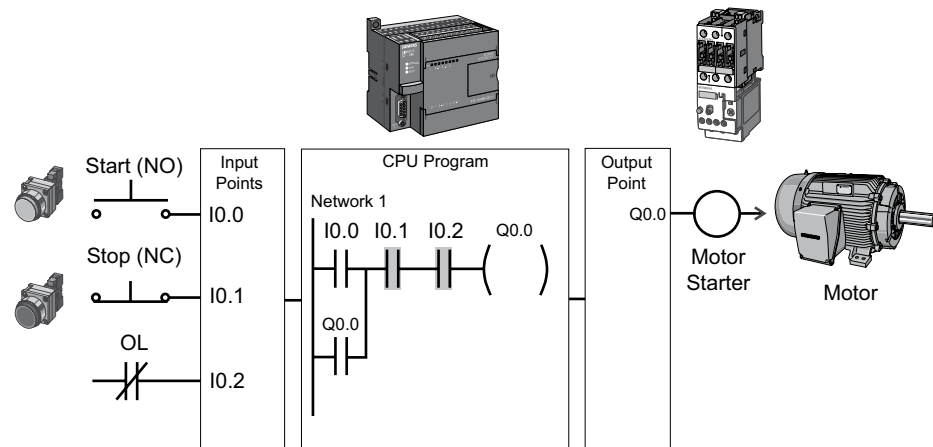


Momentarily pressing the Start pushbutton completes the path for current flow and energizes the motor starter (M). This closes the associated M and Ma (auxiliary contact located in the motor starter) contacts. When the Start button is released, current continues to flow through the Stop button and the Ma contact, and the M coil remains energized.

The motor will run until the normally closed Stop button is pressed, unless the overload relay (OL) contacts open. When the Stop button is pressed, the path for current flow is interrupted, opening the associated M and Ma contacts, and the motor stops.

## PLC Motor Control

This motor control application can also be accomplished with a PLC. In the following example, a normally open Start pushbutton is wired to the first input (I0.0), a normally closed Stop pushbutton is wired to the second input (I0.1), and normally closed overload relay contacts (part of the motor starter) are connected to the third input (I0.2). These inputs are used to control normally open contacts in a line of ladder logic programmed into the PLC.

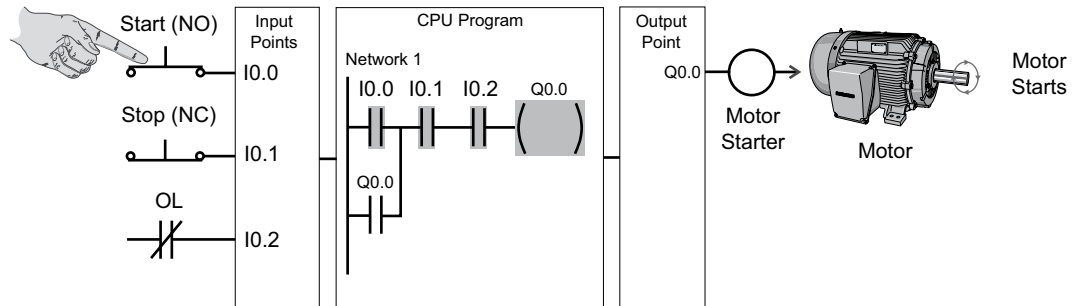


Initially, I0.1 status bit is a logic 1 because the normally closed (NC) Stop Pushbutton is closed. I0.2 status bit is a logic 1 because the normally closed (NC) overload relay (OL) contacts are closed. I0.0 status bit is a logic 0, however, because the normally open Start pushbutton has not been pressed.

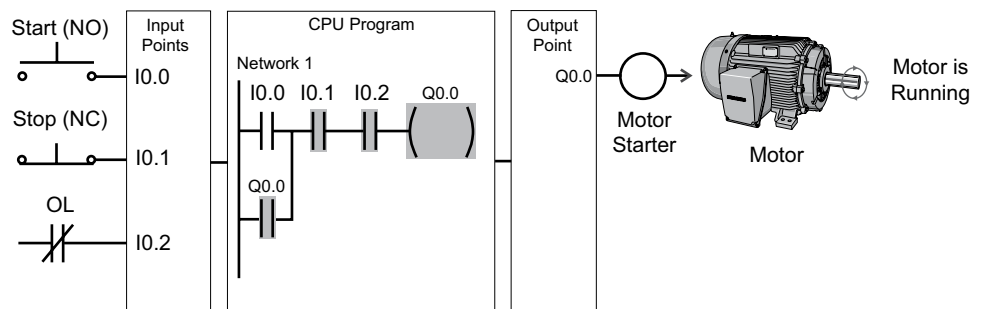
Normally open output Q0.0 contact is also programmed on Network 1 as a sealing contact. With this simple network, energizing output coil Q0.0 is required to turn on the motor.

## Program Operation

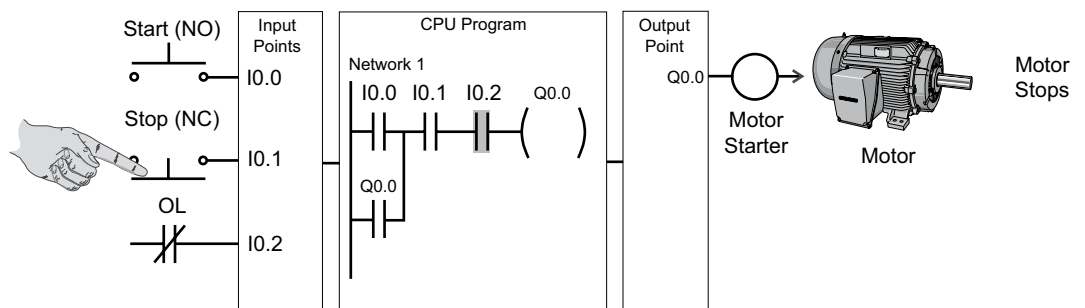
When the Start pushbutton is pressed, the CPU receives a logic 1 from input I0.0. This causes the I0.0 contact to close. All three inputs are now a logic 1. The CPU sends a logic 1 to output Q0.0. The motor starter is energized and the motor starts.



The output status bit for Q0.0 is now a 1. On the next scan, when normally open contact Q0.0 is solved, the contact will close and output Q0.0 will stay on even if the Start pushbutton is released.



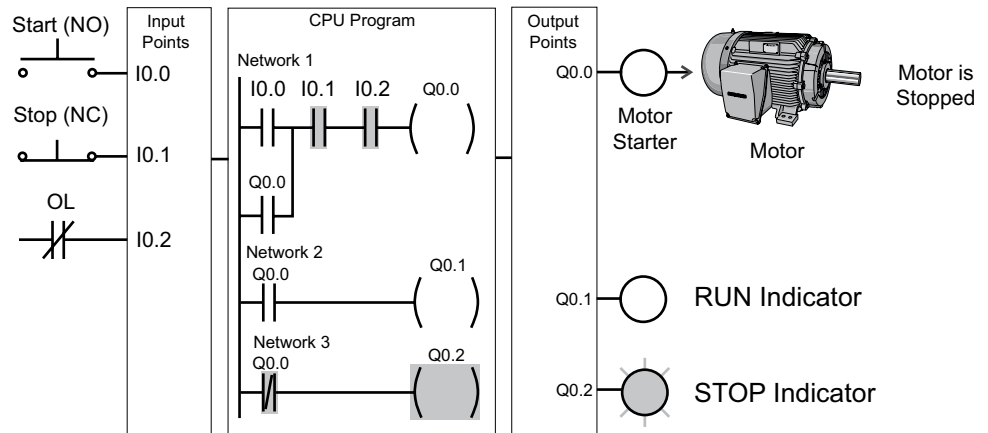
When the Stop pushbutton is pressed, input I0.1 turns off, the I0.1 contact opens, output coil Q0.0 de-energizes, and the motor turns off.



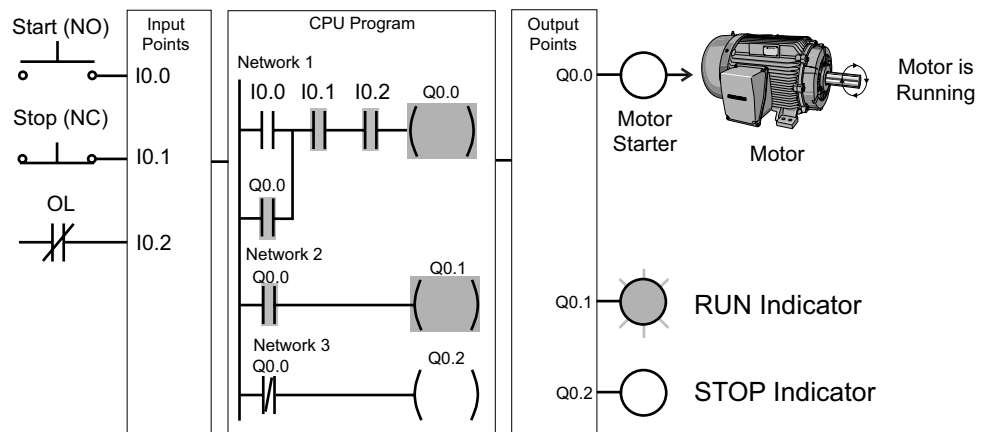
## Adding Run and Stop Indicator Lights

The application can be easily expanded to include indicator lights for run and stop conditions. In this example, a RUN indicator light is connected to output Q0.1 and a STOP indicator light is connected to output Q0.2.

The ladder logic for this application includes normally open Q0.0 contact connected on Network 2 to output coil Q0.1 and normally closed Q0.0 contact connected on Network 3 to output coil Q0.2. When Q0.0 is off, the normally open Q0.0 contact on Network 2 is open and the RUN indicator off. At the same time, the normally closed Q0.0 contact is closed and the STOP indicator is on.



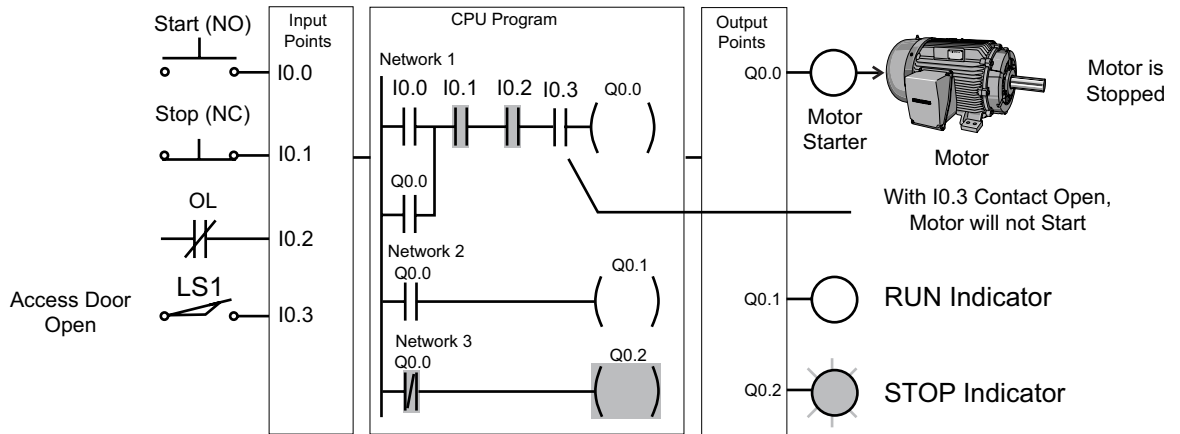
When the Start button is pressed, the PLC starts the motor. Output Q0.0 is now on. Normally open Q0.0 contact on Network 2 is now closed and the RUN indicator is on. At the same time, the normally closed Q0.0 contact on Network 3 is open and the STOP indicator light connected to output Q0.2 is off.



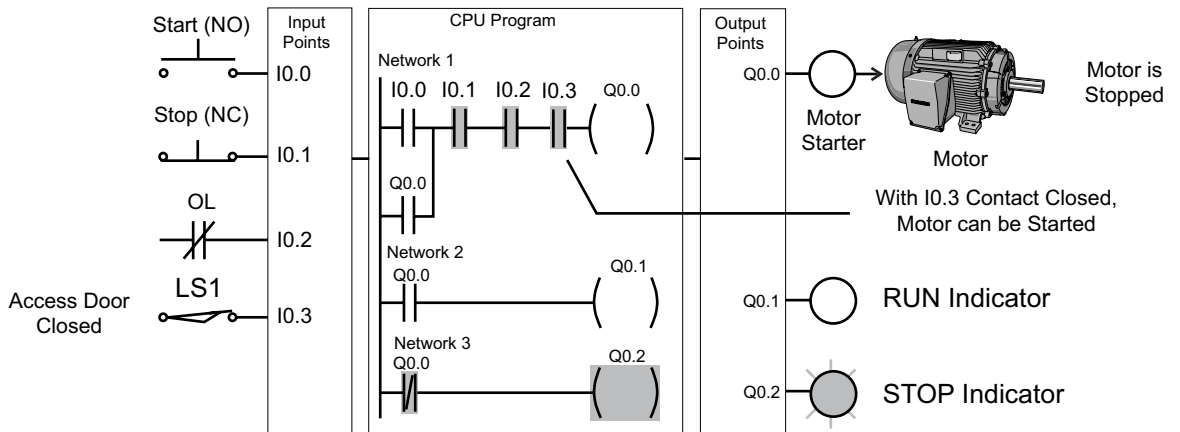
## Adding a Limit Switch

The application can be further expanded by adding a limit switch. The limit switch could be used in this application for a variety of functions. For example, the limit switch could be used to stop the motor or prevent the motor from being started.

In this example, the limit switch is associated with an access door to the motor or its associated equipment. The limit switch is connected to input I0.3 and controls a normally open contact in the program. If the access door is open, limit switch LS1 is open and normally open contact I0.3 is also open. This prevents the motor from starting.



When the access door is closed, limit switch LS1 is closed and normally open contact I0.3 is also closed. This allows the motor to start when the Start pushbutton is pressed.

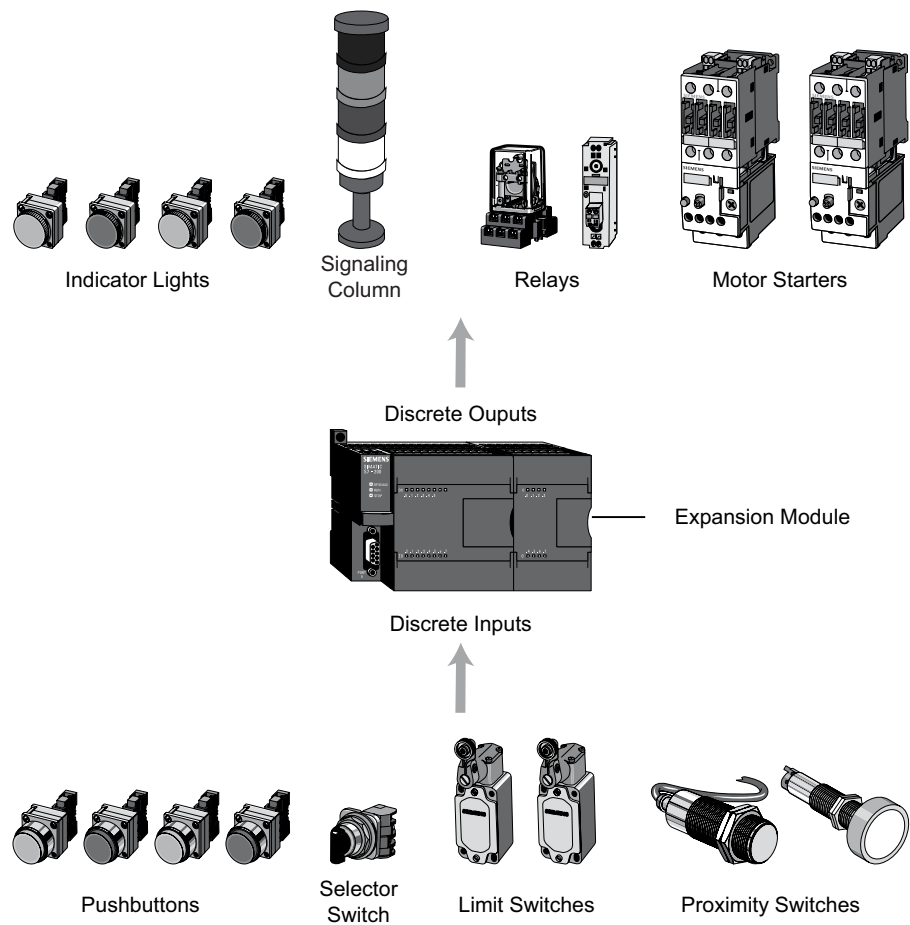


## Further Expansion

The PLC program can be further expanded to accommodate a wide variety of commercial and industrial applications.

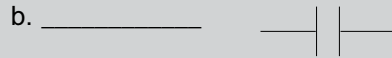
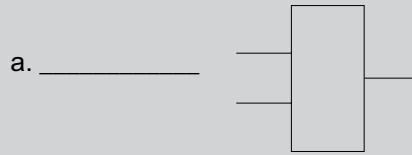
Start/Stop pushbuttons, selector switches, indicator lights, and signaling columns can be added. Motor starters can be added for control of additional motors. Over-travel limit switches can be added along with proximity switches for sensing object position. Various types of relays can be added to expand the variety of devices being controlled.

As needed, expansion modules can be added to further increase the I/O capability. The applications are only limited by the number of I/Os and amount of memory available for the PLC.



## Review 4

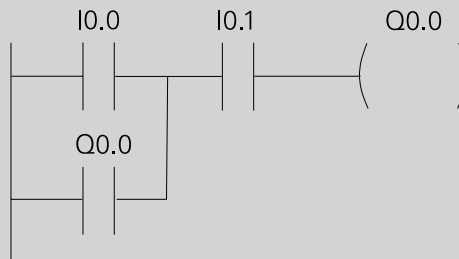
1. Identify the following symbols:



2. Complete the following tables:

AND Function			OR Function		
Input 1	Input 2	Output	Input 1	Input 2	Output
0	0	a. _____	0	0	e. _____
0	1	b. _____	0	1	f. _____
1	0	c. _____	1	0	g. _____
1	1	d. _____	1	1	h. _____

3. In the following network, coil Q0.0 will be on when contact \_\_\_\_\_ is closed and either contact \_\_\_\_\_ or contact \_\_\_\_\_ or both are closed.



# Analog Inputs and Outputs

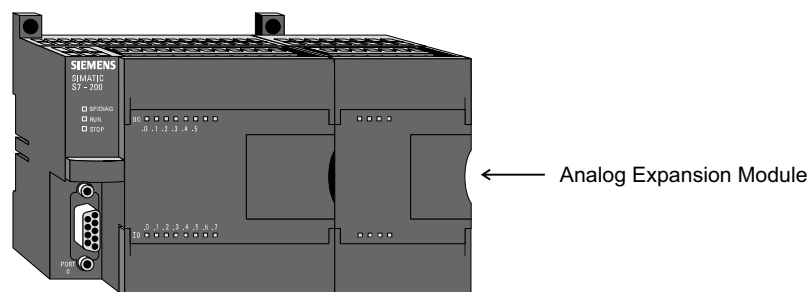
Many PLCs also work with analog I/O devices. Analog devices use signals that are continuously variable within a specified range, such as 0 to 10 VDC or 4 to 20 mA.

Analog signals are used to represent variable values, such as speed, rate of flow, temperature, weight, level, etc. In order to process an input of this type, a PLC must convert the analog signal to a digital value. S7-200 PLCs convert each analog voltage or current value into a 12-bit digital value.

Digital values from analog inputs are stored in addressable memory for use by the user program. Similarly, the user program can place digital values in addressable memory locations for conversion to analog values for the designated analog outputs.

The only S7-200 CPU model with analog I/O points on board is CPU 224XP, which has 2 analog inputs and 1 analog output. However, analog I/O points can be added using expansion modules for any CPU other than CPU 221. CPU 222 allows for 2 expansion modules and the remaining CPUs allow for 7 expansion modules.

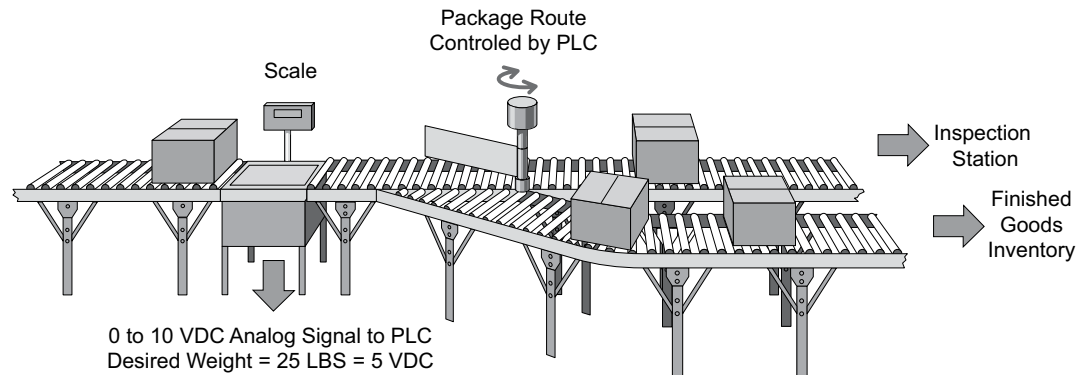
Expansion modules are available with 4 or 8 analog inputs, 2 or 4 analog outputs, or 4 analog inputs and 1 analog output. In addition, expansion modules are available for use with thermocouples or RTD type sensors which sense the temperature at a specific point in a machine or process.





## Analog Input Example

Analog inputs can be used for a variety of purposes. In the following example, a scale is connected to a load cell. A load cell is a device that generates an electrical output proportional to the force applied.



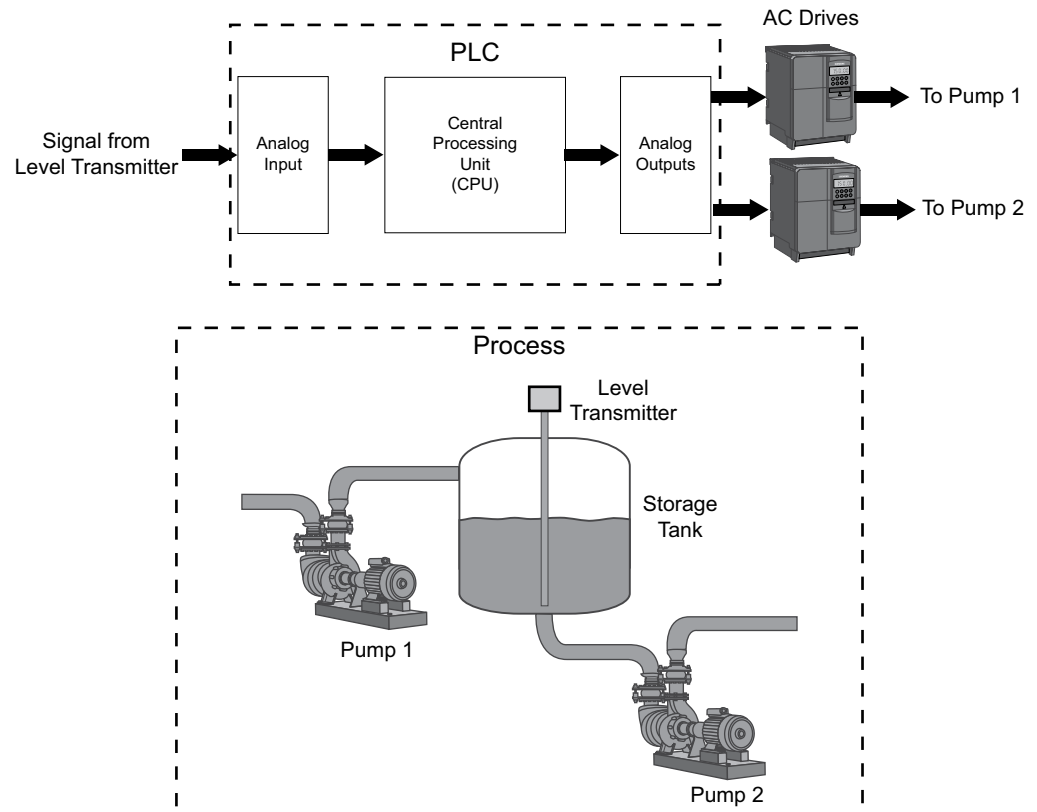
The load cell in this example converts a value of weight from 0 to 50 pounds into a 0 - 10 VDC analog value. The 0 - 10 VDC load cell signal is connected to an S7-200 PLC's analog input.

The analog value applied to the PLC can be used in various ways. For instance, the actual weight can be compared to a desired weight for a package. Then, as the package is moved on a conveyor, the S7-200 PLC can control a gate to direct packages of varying weight.

## Analog Output Example

Analog outputs from a PLC are often supplied directly or through signal converters or transmitters to control valves, instruments, electronic drives or other control devices which respond to analog signals.

For example, analog outputs from the PLC could be used to control the flow of fluid in a process by controlling AC drives. Rather than simply turning the AC drives on or off, which could be accomplished by discrete outputs, analog signals can be used to control the output of the AC drives. This would allow the speed of the pumps to be varied dynamically in response to changes in process requirements.

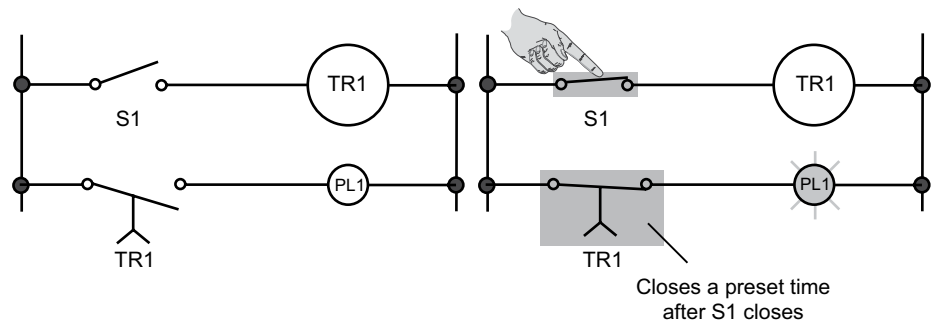


# Timers

In a PLC, **timers** are programming functions that keep track of time and allow PLC programs to provide varied responses depending on the elapsed time.

## Hard-wired Timing Circuit

Timers in a PLC program can be compared to hard-wired timing circuits, such as the one represented in the accompanying control line diagram. In this example, normally open (NO) switch (S1) is used with timer (TR1). When S1 closes, TR1 begins timing. When the timer's preset time elapses, TR1 closes its associated normally open TR1 contact and pilot light PL1 turns on. When S1 opens, TR1 de-energizes immediately, the TR1 contact opens, and PL1 turns off.

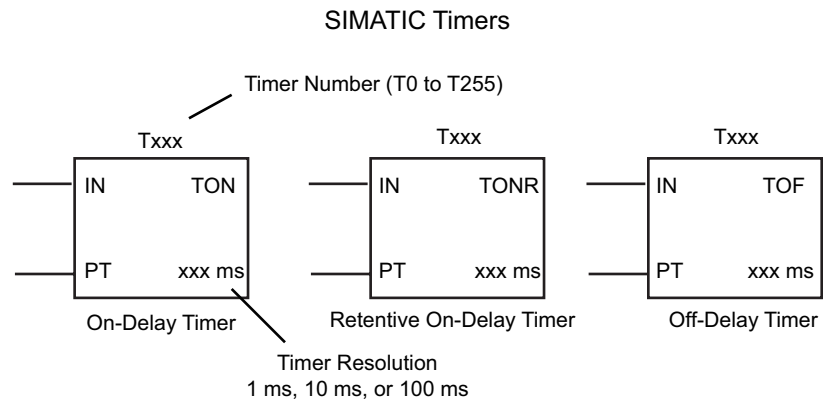


This type of timer is referred to as an on-delay timer. The term "on-delay" indicates that the timing begins when the timer receives a signal to turn on. In this example, that happens when S1 closes.

## S7-200 SIMATIC Timers

The S7-200 SIMATIC LAD instruction set includes three types of timers: On-Delay Timer (TON), Retentive On-Delay Timer (TONR), and Off-Delay Timer (TOF). Timers are represented in an S7-PLC ladder logic program by boxes.

S7-200 timers have a resolution of 1 millisecond, 10 milliseconds, or 100 milliseconds. This resolution appears in the lower right corner of the timer box. As shown in the following illustration, the resolution and type of timer that can be used depends on the timer number. The maximum value of time shown is for a single timer. By adding program elements, greater time intervals can be timed.



Timer Type	Timer Number	Resolution	Maximum Value
TONR (retentive)	T0, T64	1 ms	32.767 seconds
	T1 to T4, T65 to T68	10 ms	327.67 seconds
	T5 to T31, T69 to T95	100 ms	3276.7 seconds
TON, TOF (non-retentive)	T32, T96	1 ms	32.767 seconds
	T33 to T36, T97 to T100	10 ms	327.67 seconds
	T37 to T63, T101 to T255	100 ms	3276.7 seconds

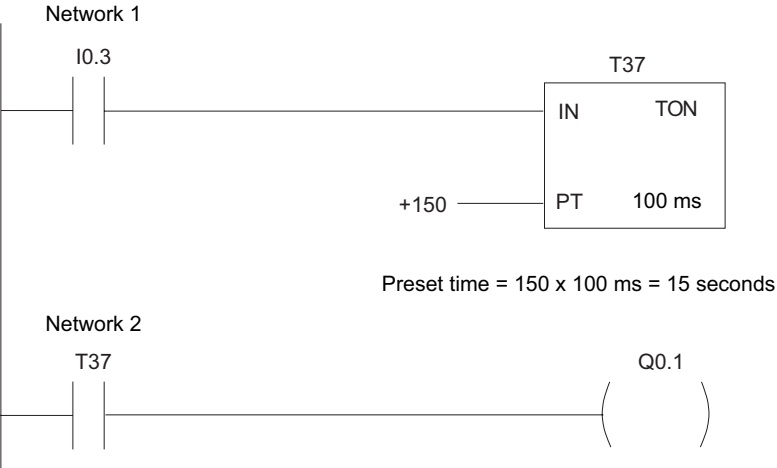
### SIMATIC On-Delay Timer (TON)

The previous example illustrated how a hardware on-delay timer works. The corresponding software function in the S7-200 SIMATIC LAD instruction set is the **On-Delay Timer (TON)**.

After the On-Delay Timer (TON) receives an enable (logic 1) at its input (IN), a predetermined amount of time (preset time - PT) passes before the timer bit (T-bit) turns on. The T-bit is a logic function internal to the timer and is not shown on the symbol. The timer resets the accumulated time to zero when the enabling input goes to a logic 0.

In the following timer example, when input I0.3 turns on, the I0.3 contact closes, and timer T37 begins timing. T37 has a time base of 100 ms (0.1 seconds). The preset time (PT) value has been set to 150. Because the resolution of the timer is set to 100 ms, a preset value of 150 is equal to 15 seconds (150 x 100 ms). Therefore, 15 seconds after the I0.3 contact closes, timer bit T37 becomes a logic 1, the T37 contact closes, and output coil Q0.1 and its associated output point turn on.

If the switch opens before 15 seconds has elapsed, the accumulate time resets to 0. Because this type of timer does not retain its accumulated time when its input (IN) goes to logic 0, it is said to be non-retentive.

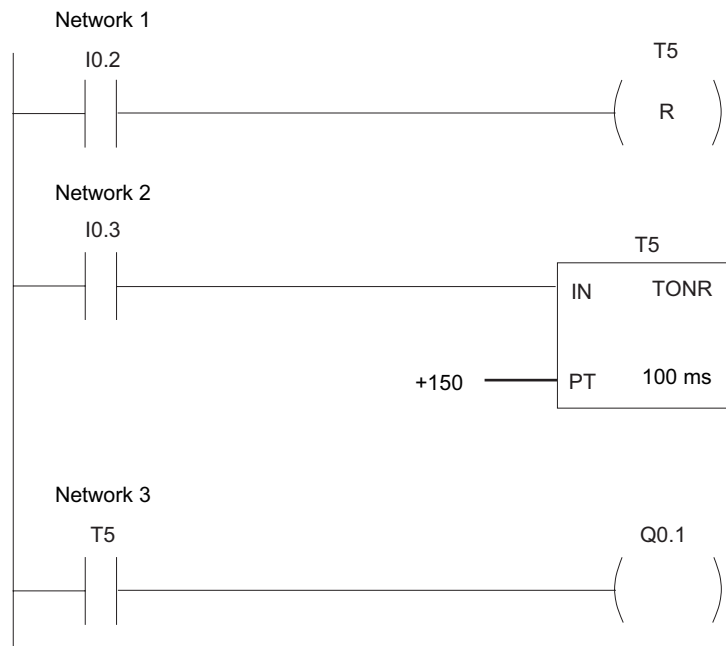


## SIMATIC Retentive On-Delay Timer (TONR)

The **SIMATIC Retentive On-Delay Timer (TONR)** functions in a similar manner to the On-Delay Timer (TON). Just like the On-Delay timer (TON), the Retentive On-Delay Timer (TONR) times when the enabling input (IN) is on. However, the Retentive On-Delay Timer (TONR) does not reset when the input (IN) turns off. Instead, the timer must be reset with a Reset (R) instruction.

The following example shows a Retentive On-Delay timer (TONR) with a resolution of 100 ms and preset value of 150 (15 seconds). When input I0.3 turns on, I0.3 contact closes, and timer T5 begins timing. If, for example, after 10 seconds input I0.3 turns off, the timer stops. When input I0.3 turns on again, the timer begins timing at 10 seconds. Timer bit T5 turns on 5 seconds after input I0.3 closes for the second time. When timer bit T5 turns on, contact T5 closes, and output Q0.1 turns on.

The Reset (R) function, shown in network 1, is necessary to reset the accumulated time of the Retentive On-delay Timer (TONR) to zero. In this example, the Reset (R) function turns on and resets the timer when contact I0.2 closes. This causes the T5 contact to open and output Q0.1 to turn off.

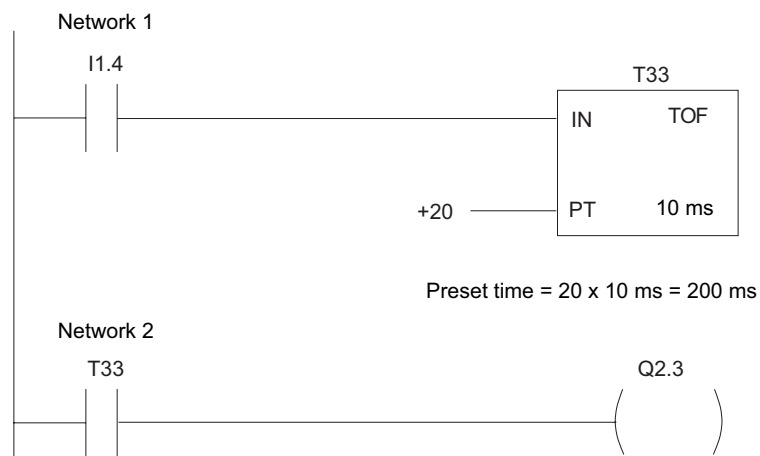


## SIMATIC Off-Delay Timer (TOF)

The **SIMATIC Off-Delay Timer (TOF)** begins timing when input (IN) turns off. In the following example, when contact I1.4 closes, the current value of timer T33 is set to 0, timer bit T33 turns on immediately, closing the T33 contact, and turning on output Q2.3.

When contact I1.4 opens, the timer times until the preset time elapses, 200 ms in this example. Then, timer bit T33 turns off, contact T33 opens, and output Q2.3 turns off.

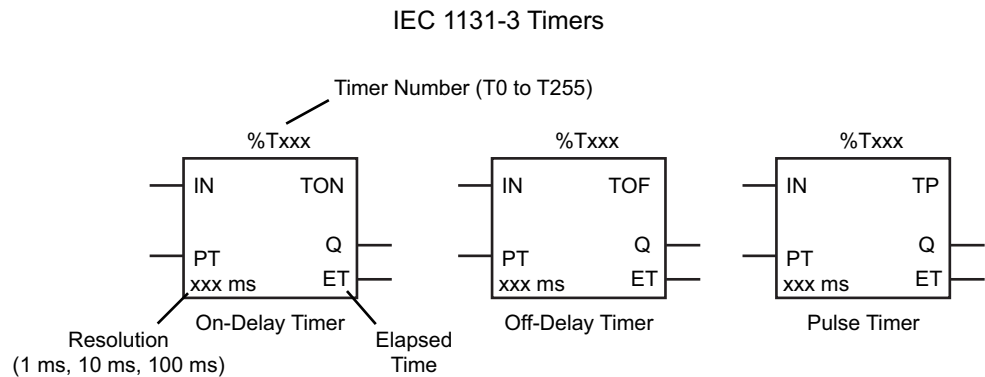
If the I1.4 contact had closed again before the 200 ms preset time had elapsed, the timer's current value would again be set to 0, timer bit T33 would remain on, contact T33 would remain closed, and output Q2.3 would remain on.



## IEC 1131-3 Timers

It is not the intent of this course to cover all S7-200 instructions, but timer instructions provide an opportunity to understand some of the differences between the SIMATIC and IEC 1131-3 instruction sets.

The timers previously discussed were SIMATIC timers. The **IEC 1131-3 instruction set** also includes three timers, **On-Delay Timer (TON)**, **Off-Delay Timer (TOF)**, and **Pulse Timer (TP)**. The same three resolutions (1 ms, 10 ms, and 100 ms) are available as for the SIMATIC timers, and the resolution is determined by the timer number as shown in the following illustration.



Timer Number	Resolution	Maximum Value
T32, T96	1 ms	32.767 seconds
T33 to T36, T97 to T100	10 ms	327.67 seconds
T37 to T63, T101 to T255	100 ms	3276.7 seconds

The On-Delay Timer (TON) begins timing when its enable input (IN) turns on. When the elapsed time (ET) equals the preset time (PT), the timer stops timing, and the output (Q) turns on. The timer is reset, when IN turns off.

The Off-Delay Timer (TOF) output (Q) turns on immediately when IN turns on. When IN turns off, the timer begins timing. When ET equals PT, Q turns off. The elapsed time is maintained until the next time IN turns on. If IN turns on before ET equals PT, Q remains on.

The Pulse Timer (TP) generates pulses of a preset duration. When IN turns on, Q turns on, and the timer begins timing. When ET equals PT, Q turns off. The elapsed time is maintained until IN turns off.



## Review 5

1. CPU \_\_\_\_\_ has two analog inputs and one analog output on-board.
2. Three types of SIMATIC timers available in the S7-200 instruction set are \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_.
3. The maximum time value for a 100 millisecond time base timer is \_\_\_\_\_ seconds.
4. Which SIMATIC Timer requires a Reset instruction?
5. Three types of IEC 1131-3 timers available in the S7-200 instruction set are \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_.

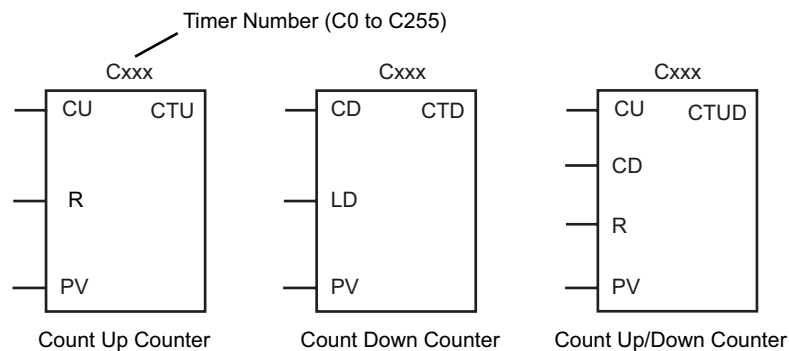
# Counters

Just like mechanical counters, PLC counter instructions keep track of events. As it counts, a counter instruction compares an accumulated count value to a preset value to determine when the desired count has been reached. Counters can be used to start an operation when a count is reached or to prevent an operation from occurring until a count has been reached.

## S7-200 SIMATIC Counters

The S7-200 SIMATIC LAD instruction set includes three types of counters: Count Up Counter (CTU), Count Down Counter (CTD), and Count Up/Down Counter (CTUD).

SIMATIC Counters



The **Count Up Counter (CTU)** counts up by one each time the count up (CU) input transitions from off to on. When the accumulated count equals the preset value (PV) the counter bit (not shown) turns on. The counter continues to count until the accumulated count equals the maximum value (32767). When the reset input (R) turns on or when a Reset instruction is executed, the accumulated count resets to zero and the counter bit turns off.

The **Count Down Counter (CTD)** counts down by one each time the count down (CD) input transitions from off to on. When the count reaches zero, the counter bit turns on. When the load (LD) input turns on, the counter resets the current value to equal the preset value (PV), and the counter bit turns off.

The **Count Up/Down Counter (CTUD)** counts up by one each time the count up (CU) input transitions from off to on and counts down by one each time the count down (CD) input transitions from off to on. When the accumulated count equals the preset value (PV), the counter bit turns on. When the reset input (R) turns on or when a Reset instruction is executed, the accumulated count resets to zero and the counter bit turns off.

If the count reaches the maximum positive value (32,767), the next count up input sets the accumulated count to the maximum negative value (-32,767). Similarly, if the count reaches the maximum negative value, the next down count sets the accumulated count to the maximum positive value.

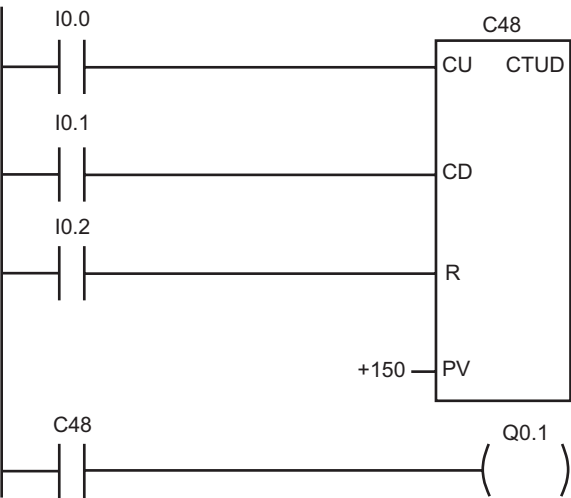
**Count Up/Down Counter (CTUD) Example**

Counters are common instructions used for counting a wide variety of events such as parts manufactured or packed, items processed, machine operations, etc. For example, a counter might be used to keep track of the items in an inventory storage area.

In the following example, Count Up/down Counter (CTUD) C48 is reset to zero when contact I0.2 closes. This could be event could be triggered automatically or manually to indicate that the associated storage location is empty.

When contact I0.0 closes, the counter counts up by 1. This could be triggered by a proximity switch sensing that an item has been placed in the storage location.

When contact I0.1 closes, the counter counts down by 1. This could be triggered by a proximity switch sensing that an item has been removed from the storage location.



In this example, the storage location has 150 spaces. When the accumulated count reaches 150, the counter bit turns on, contact C48 closes, and output Q0.1 turns on. This could trigger other logic in the program to divert new items to another location until such time as an item is removed from this location.

## IEC 1131-3 Counters

The counters previously discussed were SIMATIC counters. The IEC 1131-3 instruction set also includes three counters. These counters are similar to the SIMATIC counters, but there are a few differences.

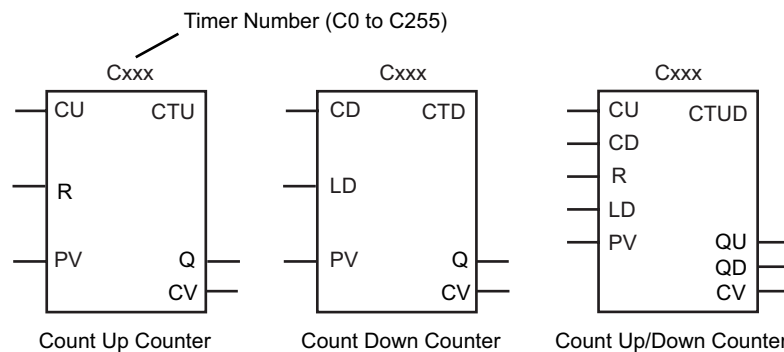
Each IEC 1131-3 counter has an output (Q) and cumulative value (CV) in the counter box.

**Count Up Counter (CTU)** stops counting when CV equals the preset value (PV), and turns on output Q.

Except for the CV and Q values, the IEC 1131-3 **Count Down Counter (CTD)** functions like the SIMATIC version. When CV equals zero, it stops counting, and output Q turns on.

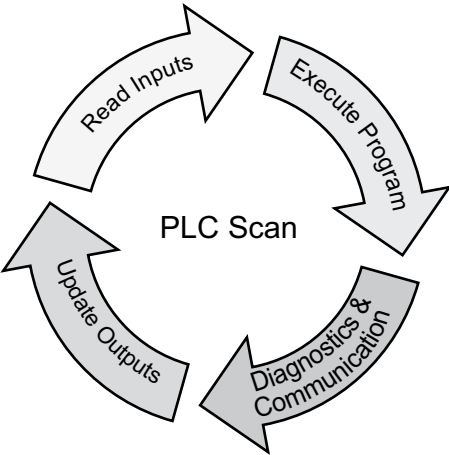
**Count Up/Down Counter (CTUD)** stops counting up when CV equals PV and turns on output QU. CTUD stops counting down when CV equals zero and turns on output QD.

IEC 1131-3 Counters



# High-Speed Instructions

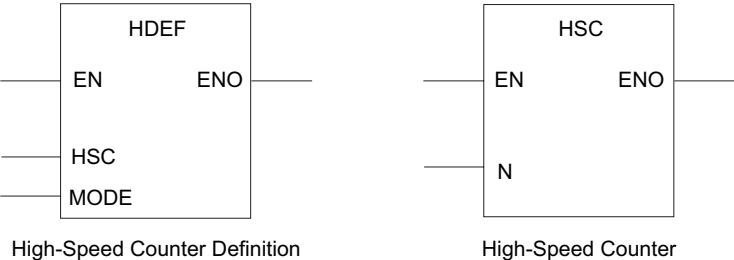
As discussed earlier, PLCs have a scan time. The scan time depends on the size of the program, the number of I/Os, and the amount of communication required. However, events may occur in an application that require a response from the PLC sooner than the scan cycle would normally permit. For these applications high-speed instructions, such as those associated with high-speed counters, can be used.



## High-Speed Counters

A **high-speed counter** is represented by two boxes in ladder logic. One box is the **High-Speed Counter Definition (HDEF)** instruction and the other box is the **High-Speed Counter (HSC)** instruction.

CPU 221 and CPU 222 support four high-speed counters (HSC0, HSC3, HSC4, HSC5). CPU 224, CPU 224XP, CPU 224XPsi, and CPU 226 support six high-speed counters (HSC0, HSC1, HSC2, HSC3, HSC4, HSC5).



## Definition Boxes and High-Speed Counters

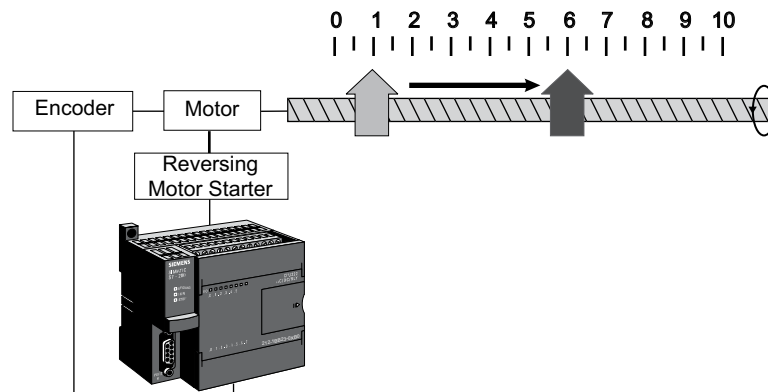
The High-Speed Counter Definition (HDEF) instruction assigns the operating mode to a specific high-speed counter (HSCx). The mode selection defines the clock, direction, start, and reset functions of the high-speed counter. High-speed counters can be defined by the definition box to operate in any of the 12 available modes. Not all counters can operate in all of the available modes, however. Refer to the S7-200 System Manual for definitions available for each counter.

The High-Speed Counter (HSC) instruction configures and controls a specific high-speed counter based upon the state of the special HSC bits. The N parameter specifies the high-speed counter number. Each counter has dedicated inputs for clocks, direction control, reset, and start, where these functions are supported.

## Positioning Example

**Positioning** is one example of an application that may require use of a high-speed counter. In the following illustration, two PLC discrete outputs (one for forward and one for reverse) control a reversing motor starter, which, in turn, controls a motor.

The motor shaft is connected to an encoder and to a positioning screw. A platform mounted on the positioning screw moves away from position 0 as the motor turns in the forward direction and towards position 0 as the motor turns in the reverse direction. Pulses from the encoder are connected to PLC inputs associated with a high speed counter.



In this example, the high-speed counter is programmed to move the platform from position 1 to position 6 and later to return to position 1. These positions could be associated with manufacturing operations performed on a part mounted on the platform.

Assume, for example, that the encoder generates 600 pulses per revolution, and it takes 1000 motor revolutions to move the platform from one position to another, moving the platform from position 1 to position 6 (5 positions) takes 5000 motor revolutions or 30,000 encoder pulses. In most practical applications, the frequency of these pulses is too high for them to be counted with inputs that are not associated with a high-speed counter.

## **Interrupts**

S7-200 PLCs incorporate instructions for use with **interrupts**. Interrupts are used to initiate a specific, short PLC program segment, called an interrupt routine, when an internal or external event occurs. After the interrupt routine has been executed, control is returned to the main program.

Three types of interrupts are supported by S7-200 PLCs, **communication port interrupts**, **I/O interrupts**, and **time-based interrupts**. Communication port interrupts are used to control a communication port operated in Freeport mode. I/O interrupts are used to respond quickly to high-speed I/O transitions, such as those associated with high-speed counters or pulse train outputs. Time-based interrupts allow the user program to execute an interrupt routine on a cyclic basis.

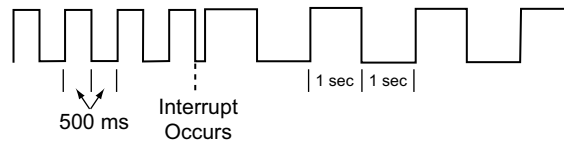
Each of these types of interrupts has an associated priority that determines which interrupt is processed first in the event that two or more interrupts are requested at the same time. Communication port interrupts have the highest priority and time-based interrupts have the lowest priority.

## **Pulse Train Output (PTO)**

S7-200 PLCs have two **PTO/PWM generators** that create either a high-speed pulse train or a pulse width modulated waveform. One generator is assigned to output point Q0.0 and the other to output point Q0.1. When a generator is activated, it controls its respective output.

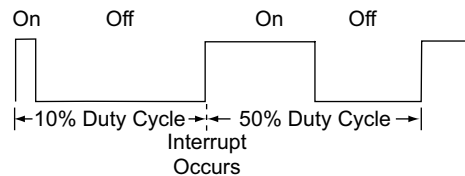
**Pulse Train Output (PTO)** is used to provide a series of pulses to an output device, such as a stepper motor driver. The PTO provides a square wave output for a specified number of pulses and a specified cycle time. The number of pulses can be from 1 to 4,294,967,295 pulses. The Pulse Train Output has a 50% duty cycle. This means the pulse is off for the same amount of time that it is on.

The number of pulses and the cycle time can be changed with an interrupt. In the accompanying example, each pulse is initially on for 500 ms and off for 500 ms. After four pulses, an interrupt occurs which changes the cycle time to 2 seconds, 1 second on and 1 second off.



## Pulse Width Modulation (PWM)

The **Pulse Width Modulation (PWM) function** provides a fixed cycle time with a variable duty cycle. When the pulse width is equal to the cycle time, the duty cycle is 100% and the output is turned on continuously. In the following example, the output initially has a 10% duty cycle (on 10%, off 90%). After an interrupt, the output switches to a 50% duty cycle (on 50%, off 50%).



The PWM function can be used to provide a programmable or adjustable control of machine timing. This allows machine operation to be varied to compensate for product variations or mechanical wear.

## And Much More

The instructions listed in this section are only examples of the types of instructions available for S7-200 PLCs. The full instruction set includes a much broader range of capabilities. Refer to the S7-200 System Manual for additional information.

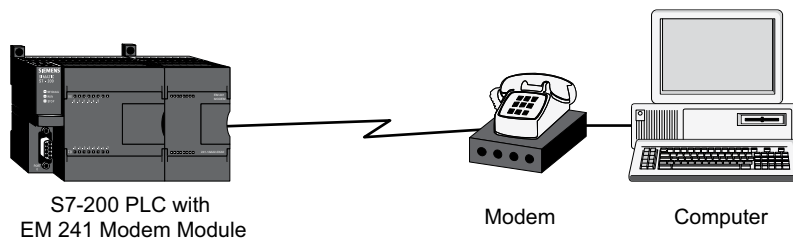


# Specialized Expansion Modules

In addition to the expansion modules previously discussed that provide additional discrete or analog I/O, several expansion modules are available to provide communication interfaces or specialized I/O functions.

## EM 241 Modem Module

One of these modules is the **EM 241 Modem module**. This module supports communication between a computer with STEP 7 Micro/WIN and an S7-200 PLC.



The EM 241 provides an international telephone line interface and supports sending numeric and text paging messages, as well as SMS (Short Message Service) messages to cellular phones. This is useful for remote diagnostics and maintenance, machine control, alarm systems, and general communication functions.

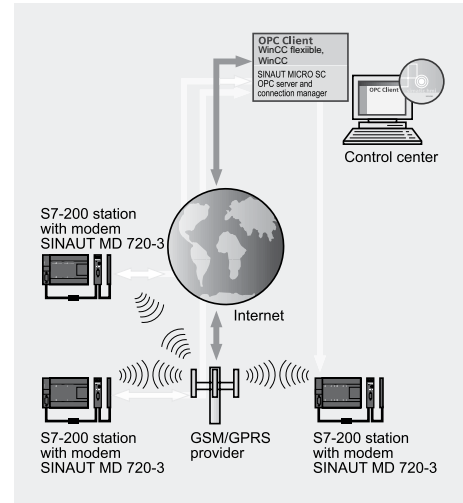
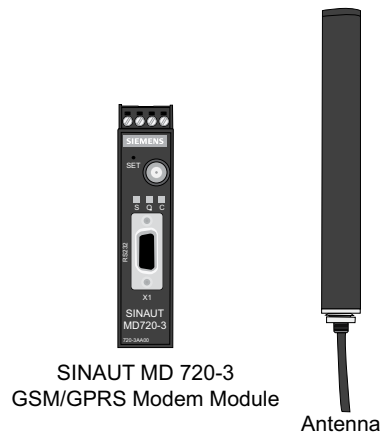
In addition to CPU-to-CPU communication via a telephone line, the EM 241 also supports Modbus RTU protocol, a protocol that has been widely used for many years.

## SINAUT MD 720-3 GSM/GPRS Modem Module

**SINAUT Telecontrol (Siemens Network Automation)** permits networking of individual controls and control systems over a WAN (Wide Area Network).

One approach for providing this capability is **SINAUT Micro**. This is a simple and flexible way to link stationary or mobile stations to a master control center. SINAUT Micro is appropriate where smaller amounts of data have to be transmitted to permit monitoring and control of remote stations using wireless techniques with the General Packet Radio Service (GPRS) of the Global System for Mobile Communication (GSM) mobile radio network.

The **SINAUT MD720-3 GSM/GPRS Modem module** and associated **ANT794-4MR antenna** are the hardware elements used to connect an S7-200 PLC into a SINAUT Micro system. **SINAUT Micro SC software** is also required.

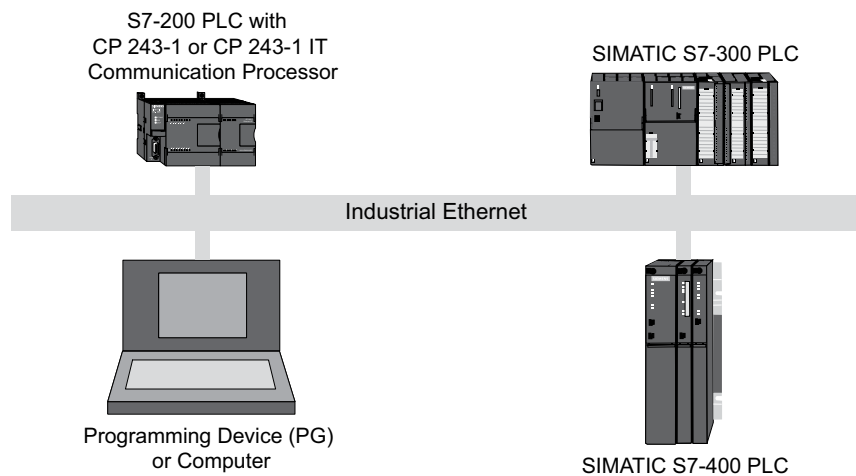


## CP 243-1, CP 243-1 IT Communication Processors

Industrial Ethernet provides a proven means of networking computers and a variety of intelligent devices. **CP 243-1 and CP 243-1 IT communication processors** are used to connect an S7-200 PLC to an **Industrial Ethernet network**.

CP 243-1 and CP 243-1 IT communication processors can be used to connect an S7-200 PLC via Industrial Ethernet to a computer running STEP 7 Micro/WIN. This allows the S7-200 PLC to be configured, programmed, and diagnosed remotely. In addition, an S7-200 PLC connected to an Industrial Ethernet network can communicate with S7-200, S7-300, and S7-400 PLCs and a variety of other devices.

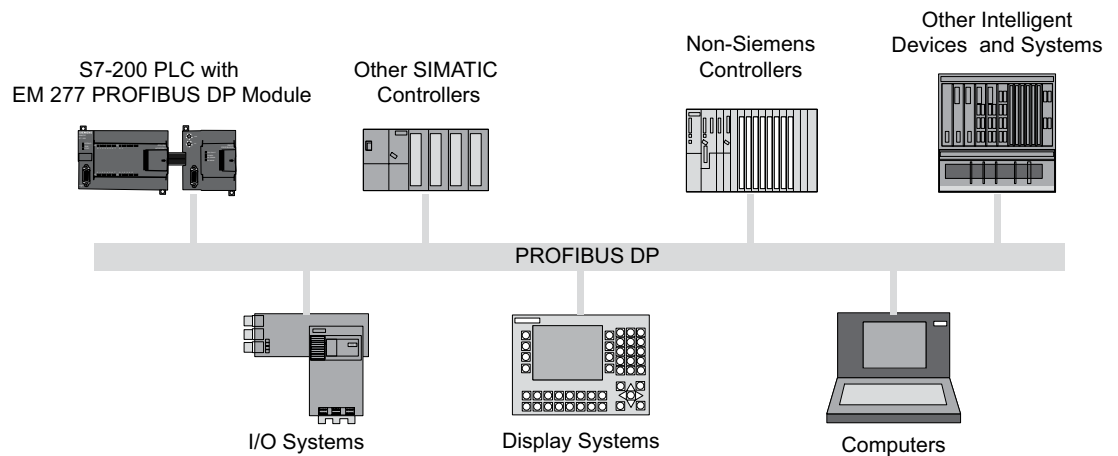
The IT functions of the CP 243-1 IT Internet module simplify the process of setting up a control system that can email diagnostic information or transfer files using Internet protocols.



## EM 277 PROFIBUS-DP Module

**PROFIBUS DP** is an open, international fieldbus communication standard that allows a broad range of intelligent devices from various manufacturers to communicate rapidly and efficiently. This reduces wiring costs as well as start-up and maintenance expenses.

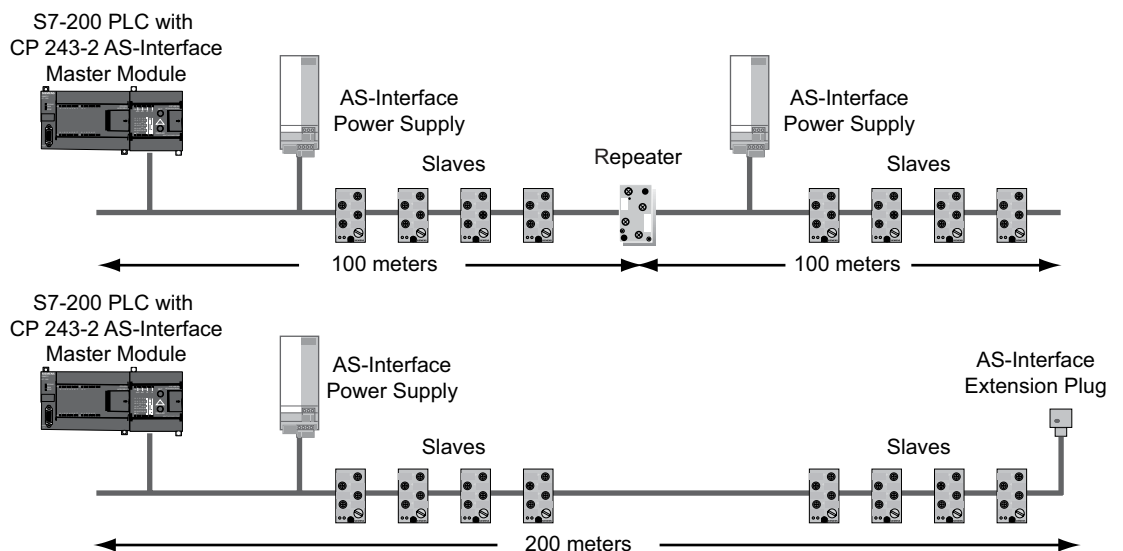
**EM 277 PROFIBUS-DP module** allows connection of the S7-200 CPU (CPU 222 and above) to a PROFIBUS-DP network as a slave.



## CP 243-2 AS-Interface Master Module

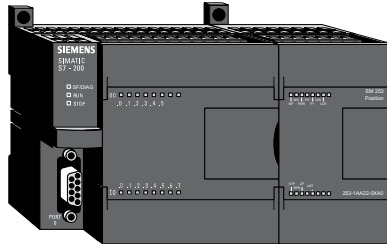
**Actuator Sensor Interface (AS-Interface or AS-i)** is a system for networking field devices such as sensors and actuators with control and operator interface devices. AS-i replaces the extensive parallel wiring often used to connect sensors and actuators to controllers with a simple 2-core cable. The cable is designed so that devices can only be connected correctly.

**CP 243-2 AS-Interface Master module** allows connection of the S7-200 CPU (CPU 222 and above) to a AS-I network as a master.



## EM 253 Position Module

Position control describes a range of applications that involve movement with varying degrees of precision. The **EM 253 Position module** is a simple but powerful positioning module that enables the user to control position systems from micro-steppers to intelligent servo drives (with integrated closed-loop control).



S7-200 PLC with EM 253 Position Module

## EM 253 Features

Features of the module include:

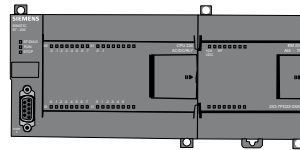
- Provides high-speed control with a range from 20 to 200,000 pulse per second
- Supports both S curve or linear acceleration and deceleration
- Provides a configurable measuring system that allows you to enter data as engineering units (such as inches or centimeters) or as a number of pulses
- Provides configurable backlash compensation
- Supports absolute, relative, and manual methods of position control
- Provides continuous operation
- Provides up to 25 motion profiles with up to 4 speed changes per profile
- Provides four different reference-point seek modes with a choice of the starting seek direction and final approach direction for each sequence
- Provides removable field wiring connectors for easy installation and removal

## Expansion Modules for Temperature Measurement

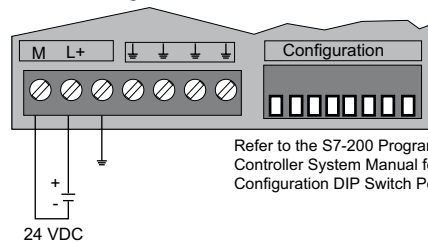
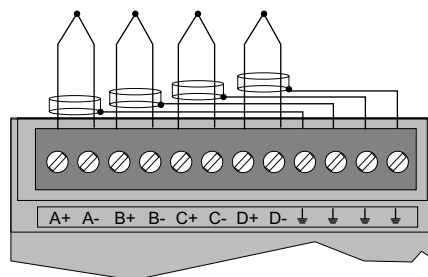
Two S7-200 PLC expansion modules are available for accurate temperature measurement, **EM 231 Thermocouple module** and **EM 231 RTD module**.

EM 231 Thermocouple module provides analog inputs for **thermocouples**. A thermocouple is a temperature sensor made from two dissimilar metals joined at a point called a junction. A thermocouple produces a small voltage that is dependent upon temperature. Various types of thermocouples are available for use in different temperature ranges. Two versions of EM 231 Thermocouple modules are available, one for four thermocouples and one for eight thermocouples. Each version is compatible with J, K, T, E, R, S, or N thermocouples, but the thermocouples used with a specific module must be of the same type.

EM 231 RTD module provides analog inputs for **resistance temperature detectors (RTDs)**. An RTD is a temperature sensor made from a metal, such as platinum, nickel, or copper, that varies in resistance in a predictable manner as temperature varies. Two versions of the EM 231 RTD module are available, one with two analog inputs and one with four analog inputs. Either version can be used with a variety of RTD types, but the RTDs used with a specific module must be of the same type.

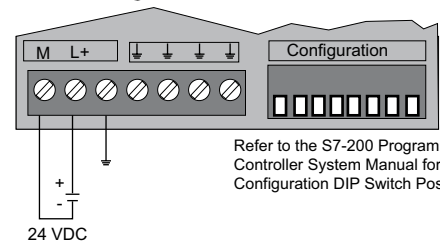
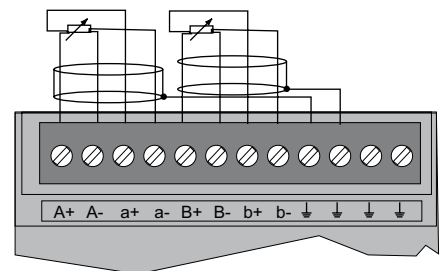


S7-200 PLC with EM 231 Expansion Module



EM 231 Thermocouple module, 4 Input Version

Refer to the S7-200 Programmable Controller System Manual for Configuration DIP Switch Positions



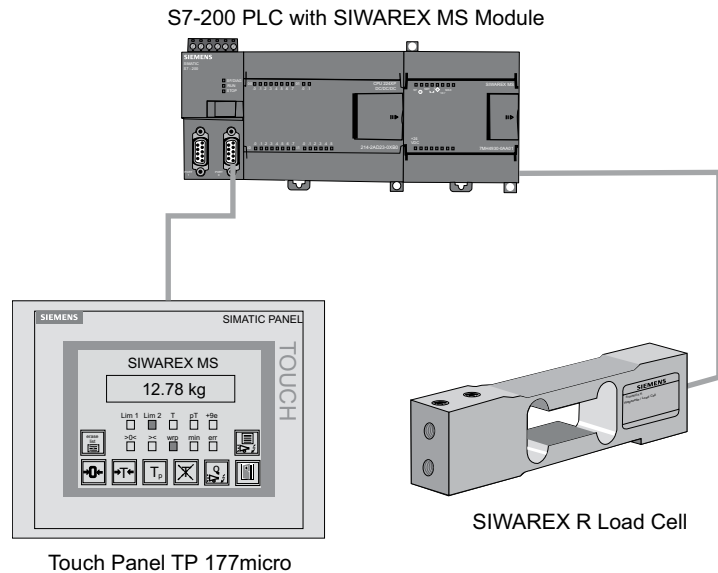
EM 231 RTD module, 2 Input Version

Refer to the S7-200 Programmable Controller System Manual for Configuration DIP Switch Positions

## SIWAREX MS Weighing Module

**SIWAREX MS Weighing module** provides a simple, easy to install approach for weighing and force measurement applications. SIWAREX MS Weighing module is designed to measure the voltage produced by sensors commonly used to measure weight, force, or torque.

SIWAREX MS is easily integrated into an S7-200 PLC system as an expansion module. This makes information obtained from SIWAREX MS available to other components of the automation system. In addition, Siemens offers a wide variety of compatible sensors and other components.



## Review 6

1. Three types of SIMATIC counters available in the S7-200 instruction set are \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_.
2. CPU 221 and CPU 222 support \_\_\_\_ high speed counters. CPU 224, CPU 224XP, CPU 224XPsi, and CPU 226 support \_\_\_\_ high speed counters.
3. S7-200 PLCs have two \_\_\_\_\_ that create either a high-speed pulse train or a pulse-width modulated waveform.
4. \_\_\_\_\_ and \_\_\_\_\_ communication processors are used to connect an S7-200 PLC to an Industrial Ethernet network.
5. \_\_\_\_\_ module allows connection of an S7-200 CPU (CPU222 and above) to a PROFIBUS-DP network as a slave.
6. \_\_\_\_\_ module allows connection of an S7-200 CPU (CPU222 and above) to an AS-I network as a master.
7. Two versions of EM 231 Thermocouple module are available, one for \_\_\_\_ thermocouples and one for \_\_\_\_ thermocouples.
8. Two versions of EM 231 RTD module are available, one for \_\_\_\_ RTDs and one for \_\_\_\_ RTDs.

# Review Answers

- Review 1** 1) a: input module, b: CPU, c: output module, d: programming device, e: operator interface; 2) 2; 3) 16; 4) 1010, 0001 0000, A.
- Review 2** 1) discrete; 2) discrete; 3) CPU; 4) Ladder logic; 5) Statement list, function block diagrams; 6) scan; 7) 1024; 8) firmware; 9) RS-485.
- Review 3** 1) CPU 221, CPU 222, CPU 224, CPU 224XP, CPU 224XPSi, CPU 226; 2) b; 3) 2, 7; 4) 8, 6; 5) 14, 10; 6) Q0.3; 7) DIN.
- Review 4** 1) a: box, b: normally open contact, c: coil;  
2) AND Function - a: 0, b: 0, c: 0, d: 1,  
OR Function - e: 0, f: 1, g: 1, h: 1;  
3) I0.1, I0.0, Q0.0.
- Review 5** 1) 224XP; 2) On-Delay Timer (TON), Retentive On-Delay Timer (TONR), Off-Delay Timer (TOF) ; 3) 3276.7 seconds;  
4) Retentive On-Delay Timer (TONR); 5) On-Delay Timer (TON), Off-Delay Timer (TOF), Pulse Timer (TP).
- Review 6** 1) Count Up Counter (CTU), Count Down Counter (CTD), Count Up/Down Counter (CTUD); 2) 4, 6; 3) PTO/PWM generators;  
4) CP 243-1, CP 243-1 IT; 5) EM 277 PROFIBUS-DP  
6) CP 243-2 AS-Interface Master; 7) 4, 8; 8) 2, 4.



# Final Exam

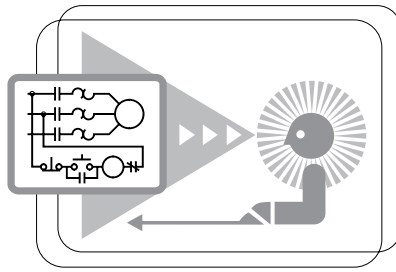
The final exam is intended to be a learning tool. The book may be used during the exam. A tear-out answer sheet is provided. After completing the test, mail the answer sheet in for grading. A grade of 70% or better is passing. If you pass the test, we will mail you a certificate of completion.

1. The component of a PLC that stores and executes the main program is the \_\_\_\_\_.
  - a. CPU
  - b. input/output module
  - c. programming device
  - d. operator interface
  
2. One byte is \_\_\_\_ consecutive bits.
  - a. 2
  - b. 8
  - c. 16
  - d. 32
  
3. A continuously varying voltage input is an example of a/an \_\_\_\_\_ input.
  - a. discrete
  - b. digital
  - c. analog
  - d. none of the above
  
4. An input that is either on or off is a/an \_\_\_\_\_ input.
  - a. analog
  - b. discrete
  - c. RTD
  - d. thermocouple
  
5. \_\_\_\_\_ is a programming language with symbols resembling elements used in hard-wired control line diagrams.
  - a. Ladder logic
  - b. Statement list
  - c. Function blocks
  - d. Flow chart
  
6. \_\_\_\_\_ is a type of memory that can be read from, but not changed.
  - a. RAM
  - b. ROM
  - c. R/W
  - d. Read-Write

7. A USB/PPI Multi-Master cable connects a personal computer's USB interface to a/an \_\_\_\_\_ connector on an S7-200 CPU.
- a. RS-485
  - b. RS-232
  - c. Ethernet
  - d. PROFIBUS-DP
8. The CPU 224 AC/DC/RELAY has \_\_\_\_\_.
- a. 8 DC inputs and 10 relay outputs
  - b. 8 AC inputs and 6 relay outputs
  - c. 14 DC inputs and 14 relay outputs
  - d. 14 DC inputs and 10 relay outputs
9. CPU 224 will accept up to \_\_\_\_ expansion modules.
- a. 3
  - b. 7
  - c. 10
  - d. 16
10. The S7-222 has the ability to store \_\_\_\_\_ bytes of user data.
- a. 1024
  - b. 8192
  - c. 2048
  - d. 10240
11. Which of the following is not part of a PLC scan?
- a. Read inputs
  - b. Execute program
  - c. Reset all timers
  - d. Update outputs
12. The address designation for output four of an S7-200 is \_\_\_\_\_.
- a. I0.4
  - b. I0.3
  - c. Q0.3
  - d. Q0.4
13. CPU 221 and CPU 222 support \_\_\_\_ high-speed counters.
- a. 2
  - b. 3
  - c. 4
  - d. 5
14. The maximum value of an S7-200 timer with a resolution of 1 millisecond is \_\_\_\_\_ seconds.
- a. 3.2767
  - b. 32.767
  - c. 327.67
  - d. 3276.7

15. An S7-200 timer with a time base of 100 milliseconds can time to a maximum value of \_\_\_\_\_ seconds.
- a. 3.2767
  - b. 32.767
  - c. 327.67
  - d. 3276.7
16. The time base of TON 32 is \_\_\_\_ millisecond(s).
- a. 0.1
  - b. 10
  - c. 1
  - d. 100
17. The maximum count of an S7-200 Count Up Counter (CTU) is \_\_\_\_\_.
- a. 32,767
  - b. 65,534
  - c. 98,301
  - d. 1,000,000
18. A/An \_\_\_\_\_ instruction is used to assign a mode to a high-speed counter.
- a. CTUD
  - b. Interrupt
  - c. HSC
  - d. HDEF
19. The \_\_\_\_\_ module allows connection of the S7-200 CPU to a PROFIBUS-DP network as a slave.
- a. EM 277
  - b. EM241
  - c. CP 243-1
  - d. CP 243-1 IT
20. \_\_\_\_\_ is used to temporarily override input or output status in order to test and debug the program or system.
- a. Transmitting
  - b. Forcing
  - c. Interrupting
  - d. Holding

# quickSTEP Online Courses



quickSTEP online courses are available at  
**<http://www.sea.siemens.com/step/default.html>**.

The quickSTEP training site is divided into three sections: Courses, Downloads, and a Glossary. Online courses include reviews, a final exam, the ability to print a certificate of completion, and the opportunity to register in the Sales & Distributor training database to maintain a record of your accomplishments.

From this site the complete text of all STEP courses can be downloaded in PDF format. These files contain the most recent changes and updates to the STEP courses.

A unique feature of the quickSTEP site is our pictorial glossary. The pictorial glossary can be accessed from anywhere within a quickSTEP course. This enables the student to look up an unfamiliar word without leaving the current work area.

