

# Table of Contents

Introduction .....	2
PLCs .....	4
Number Systems .....	8
Terminology .....	14
Basic Requirements .....	23
S7-200 Micro PLCs .....	28
Connecting External Devices .....	39
Programming A PLC .....	41
Discrete Inputs/Outputs .....	49
Analog Inputs and Outputs .....	61
Timers .....	64
Counters .....	71
High-Speed Instructions .....	75
Network Communications .....	78
Review Answers .....	81
Final Exam .....	82

# Introduction

Welcome to another course in the STEP 2000 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
- Read a basic ladder logic diagram and statement list
- Identify operational differences between different S7-200 models
- Identify the proper manual to refer to for programming or installation of an S7-200 PLC
- Connect a simple discrete input and output to an S7-200
- Select the proper expansion module for analog inputs and outputs
- Describe the operation of timers and counters

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.

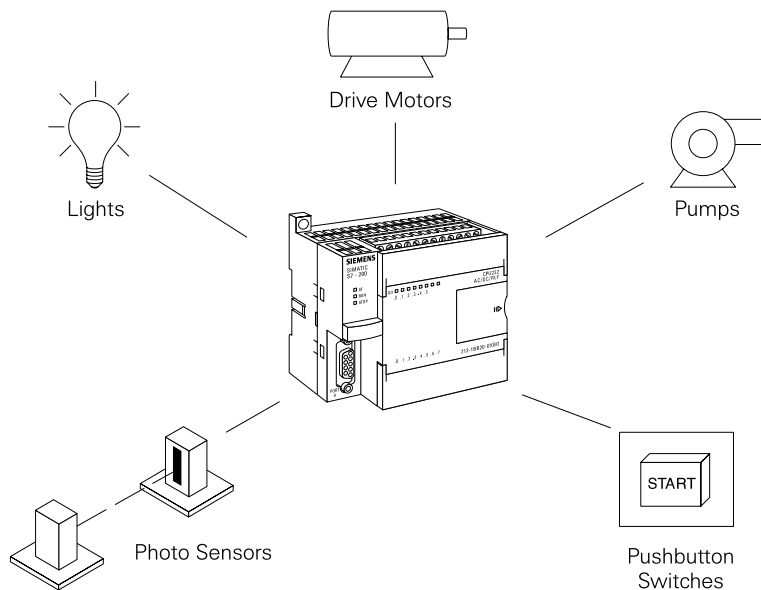
If you are an employee of a Siemens Energy & Automation authorized distributor, fill out the final exam tear-out card and mail in the card. We will mail you a certificate of completion if you score a passing grade. Good luck with your efforts.

SIMATIC, STEP 7, STEP 7-Micro, STEP 7-Micro/WIN, PG 702, and PG 740 are registered trademarks of Siemens Energy & Automation, Inc.

Other trademarks are the property of their respective owners.

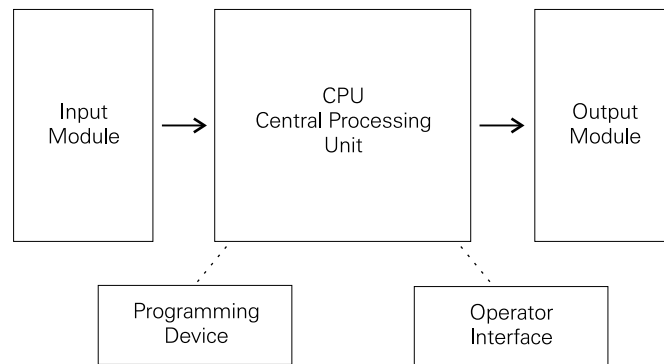
# PLCs

Programmable Logic Controllers (PLCs), also referred to as programmable controllers, are in the computer family. They are used in commercial and industrial applications. A PLC monitors inputs, makes decisions based on its program, and controls outputs to automate a process or machine. This course is meant to supply you with basic information on the functions and configurations of PLCs.

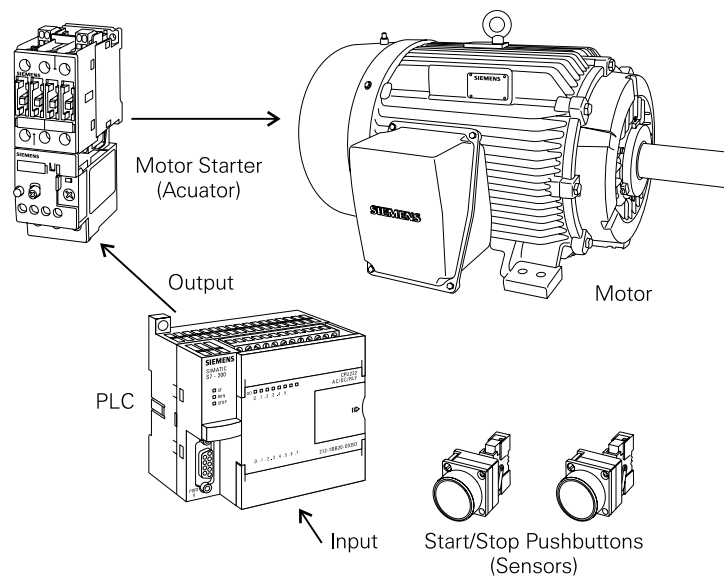


## Basic PLC Operation

PLCs consist of input modules or points, a Central Processing Unit (CPU), and output modules or points. An input accepts a variety of digital or analog signals from various field devices (sensors) and converts them into a logic signal that can be used by the CPU. The CPU makes decisions and executes control instructions based on program instructions in memory. Output modules convert control instructions from the CPU into a digital or analog signal that can be used to control various field devices (actuators). A programming device is used to input the desired instructions. These instructions determine what the PLC will do for a specific input. An operator interface device allows process information to be displayed and new control parameters to be entered.

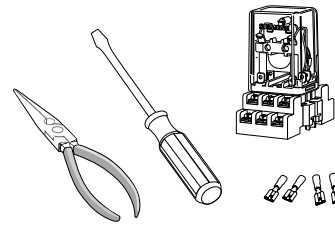
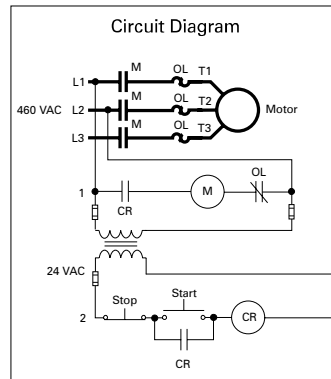


Pushbuttons (sensors), in this simple example, connected to PLC inputs, can be used to start and stop a motor connected to a PLC through a motor starter (actuator).



## Hard-Wired Control

Prior to PLCs, many of these control tasks were solved with contactor or relay controls. 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.



## PLCs

The same, as well as more complex tasks, can be done with a PLC. Wiring between devices and relay contacts is done in the PLC program. Hard-wiring, though still required to connect field devices, is less intensive. Modifying the application and correcting errors are easier to handle. It is easier to create and change a program in a PLC than it is to wire and rewire a circuit.

## Advantages

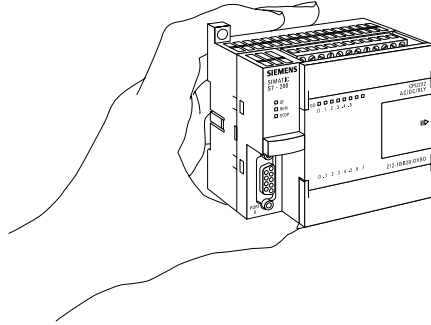
- Smaller physical size than hard-wire solutions
- Easier and faster to make changes
- PLCs have integrated diagnostics and override functions
- Diagnostics are centrally available
- Applications can be immediately documented
- Applications can be duplicated faster and less expensively

## Siemens PLCs

Siemens makes several PLC product lines in the SIMATIC® S7 family. They are: S7-200, S7-300, and S7-400.

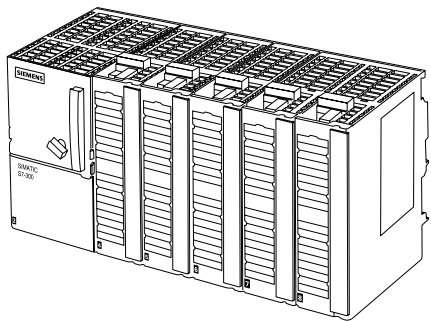
### S7-200

The S7-200 is referred to as a micro PLC because of its small size. The S7-200 has a brick design which means that the power supply and I/O are on-board. The S7-200 can be used on smaller, stand-alone applications such as elevators, car washes, or mixing machines. It can also be used on more complex industrial applications such as bottling and packaging machines.

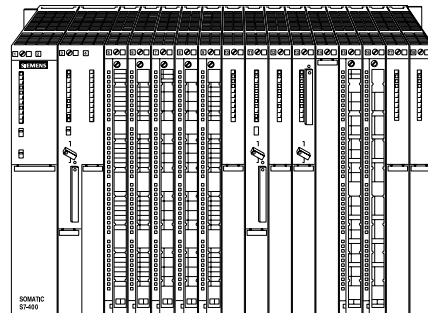


### S7-300 and S7-400

The S7-300 and S7-400 PLCs are used in more complex applications that support a greater number of I/O points. Both PLCs are modular and expandable. The power supply and I/O consist of separate modules connected to the CPU. Choosing either the S7-300 or S7-400 depends on the complexity of the task and possible future expansion. Your Siemens sales representative can provide you with additional information on any of the Siemens PLCs.



S7-300



S7-400

# Number Systems

Since a PLC is a computer, it stores information in the form of On or Off conditions (1 or 0), referred to as binary digits (bits). Sometimes binary digits are used individually and sometimes they are used to represent numerical values.

## Decimal System

Various number systems are used by PLCs. All number systems have the same three characteristics: digits, base, weight. The decimal system, which is commonly used in everyday life, has the following characteristics:

Ten digits      0, 1, 2, 3, 4, 5, 6, 7, 8, 9  
Base            10  
Weights        1, 10, 100, 1000, ...

## Binary System

The binary system is used by programmable controllers. The binary system has the following characteristics:

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

In the binary system 1s and 0s are arranged into columns. Each column is weighted. The first column has a binary weight of  $2^0$ . This is equivalent to a decimal 1. This is referred to as the least significant bit. The binary weight is doubled with each succeeding column. The next column, for example, has a weight of  $2^1$ , which is equivalent to a decimal 2. The decimal value is doubled in each successive column. The number in the far left hand column is referred to as the most significant bit. In this example, the most significant bit has a binary weight of  $2^7$ . This is equivalent to a decimal 128.

Most Significant Bit								Least Significant Bit
↓	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	↓ $2^0$
	128	64	32	16	8	4	2	1
	0	0	0	1	1	0	0	0

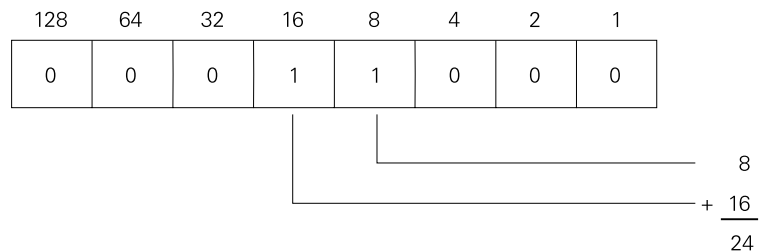


## Converting Binary to Decimal

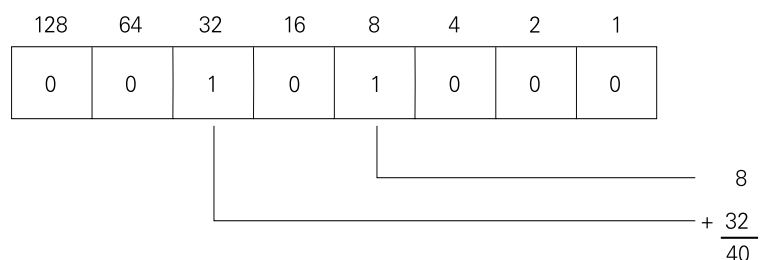
The following steps can be used to interpret a decimal number from a binary value.

- 1) Search from left to right (least significant to most significant bit) for 1s.
- 2) Write down the decimal representation of each column containing a 1.
- 3) Add the column values.

In the following example, the fourth and fifth columns from the right contain a 1. The decimal value of the fourth column from the right is 8, and the decimal value of the fifth column from the right is 16. The decimal equivalent of this binary number is 24. The sum of all the weighted columns that contain a 1 is the decimal number that the PLC has stored.

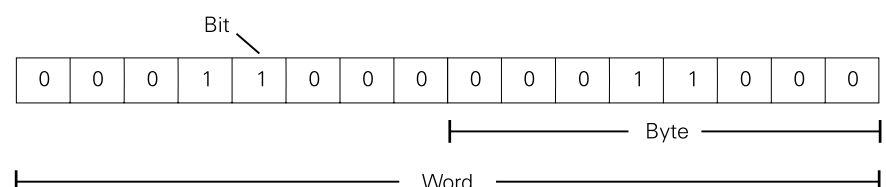


In the following example the fourth and sixth columns from the right contain a 1. The decimal value of the fourth column from the right is 8, and the decimal value of the sixth column from the right is 32. The decimal equivalent of this binary number is 40.



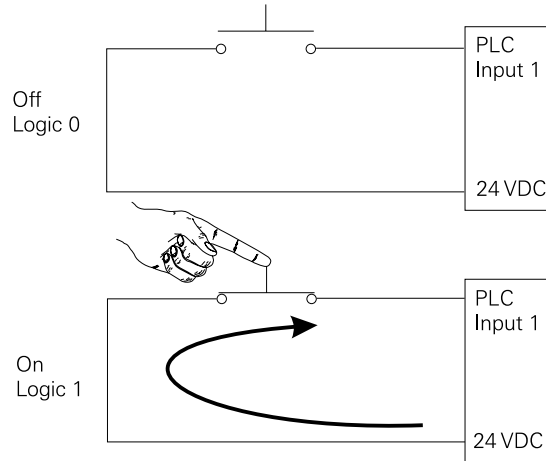
## Bits, Bytes, and Words

Each binary piece of data is a bit. Eight bits make up one byte. Two bytes, or 16 bits, make up one word.



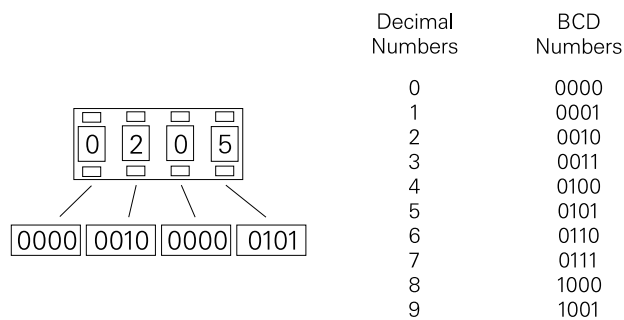
## Logic 0, Logic 1

Programmable controllers can only understand a signal that is On or Off (present or not present). The binary system is a system in which there are only two numbers, 1 and 0. Binary 1 indicates that a signal is present, or the switch is On. Binary 0 indicates that the signal is not present, or the switch is Off.



## BCD

Binary-Coded Decimal (BCD) are decimal numbers where each digit is represented by a four-bit binary number. BCD is commonly used with input and output devices. A thumbwheel switch is one example of an input device that uses BCD. The binary numbers are broken into groups of four bits, each group representing a decimal equivalent. A four-digit thumbwheel switch, like the one shown here, would control 16 (4 x 4) PLC inputs.



## Hexadecimal

Hexadecimal is another system used in PLCs. The hexadecimal system has the following characteristics:

16 digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
Base	16
Weights	Powers of base 16 (1, 16, 256, 4096 ...)

The ten digits of the decimal system are used for the first ten digits of the hexadecimal system. The first six letters of the alphabet are used for the remaining six digits.

A = 10	D = 13
B = 11	E = 14
C = 12	F = 15

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 digit represents the exact status of four binary bits. To convert a decimal number to a hexadecimal number the decimal number is divided by the base of 16. To convert decimal 28, for example, to hexadecimal:

$$16 \overline{) 28} \quad \begin{array}{l} 1 \text{ r } 12 \\ \underline{16} \\ 28 \end{array}$$

Decimal 28 divided by 16 is 1 with a remainder of 12. Twelve is equivalent to C in hexadecimal. The hexadecimal equivalent of decimal 28 is 1C.

The decimal value of a hexadecimal number is obtained by multiplying the individual hexadecimal digits by the base 16 weight and then adding the results. In the following example the hexadecimal number 2B is converted to its decimal equivalent of 43.

$$\begin{array}{l} 16^0 = 1 \\ 16^1 = 16 \\ B = 11 \end{array}$$

$16^1$	$16^0$
2	B

  
$$\begin{array}{r} 11 \times 1 = 11 \\ 2 \times 16 = 32 \\ \hline 43 \end{array}$$

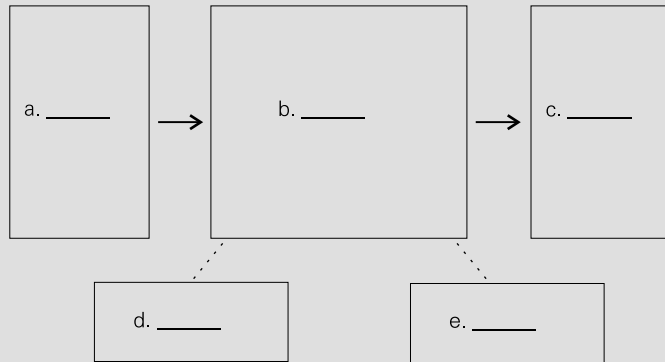
## Conversion of Numbers

The following chart shows a few numeric values in decimal, binary, BCD, and hexadecimal representation.

Decimal	Binary	BCD	Hexadecimal
0	0	0000	0
1	1	0001	1
2	10	0010	2
3	11	0011	3
4	100	0100	4
5	101	0101	5
6	110	0110	6
7	111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	0001 0000	A
11	1011	0001 0001	B
12	1100	0001 0010	C
13	1101	0001 0011	D
14	1110	0001 0100	E
15	1111	0001 0101	F
16	1 0000	0001 0110	10
17	1 0001	0001 0111	11
18	1 0010	0001 1000	12
19	1 0011	0001 1001	13
20	1 0100	0010 0000	14
.	.	.	.
.	.	.	.
126	111 1110	0001 0010 0110	7E
127	111 1111	0001 0010 0111	7F
128	1000 0000	0001 0010 1000	80
.	.	.	.
.	.	.	.
510	1 1111 1110	0101 0001 000	1FE
511	1 1111 1111	0101 0001 0001	1FF
512	10 0000 0000	0101 0001 0010	200

## Review 1

1. Identify the following:



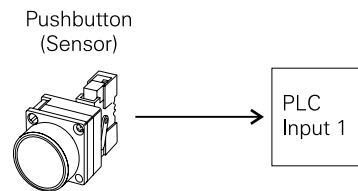
2. The binary number system has a base \_\_\_\_\_ .
3. The hexadecimal number system has a base \_\_\_\_\_ .
4. Convert a decimal 10 to the following:  
Binary \_\_\_\_\_  
BCD \_\_\_\_\_  
Hexadecimal \_\_\_\_\_

# Terminology

The language of PLCs consists of a commonly used set of terms; many of which are unique to PLCs. In order to understand the ideas and concepts of PLCs, an understanding of these terms is necessary.

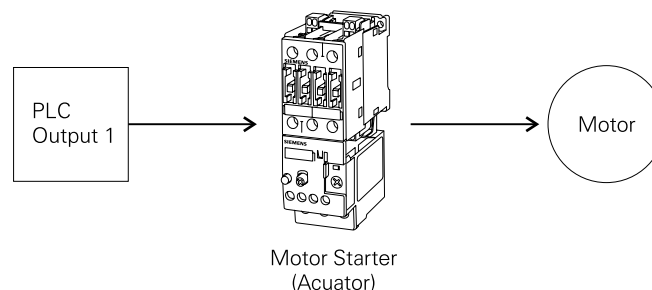
## Sensor

A sensor is a device that converts a physical condition into an electrical signal for use by the PLC. Sensors are connected to the input of a PLC. A pushbutton is one example of a sensor that is connected to the PLC input. An electrical signal is sent from the pushbutton to the PLC indicating the condition (open/closed) of the pushbutton contacts.



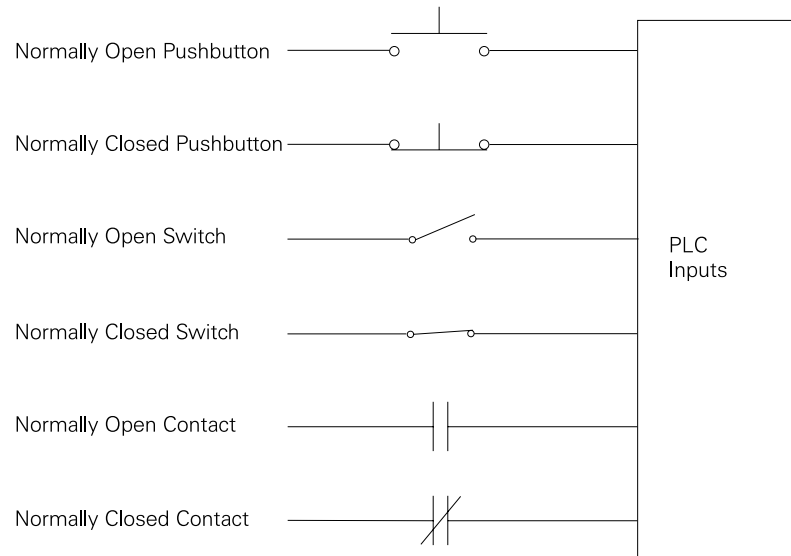
## Actuators

Actuators convert an electrical signal from the PLC into a physical condition. Actuators are connected to the PLC output. A motor starter is one example of an actuator that is connected to the PLC output. Depending on the output PLC signal the motor starter will either start or stop the motor.

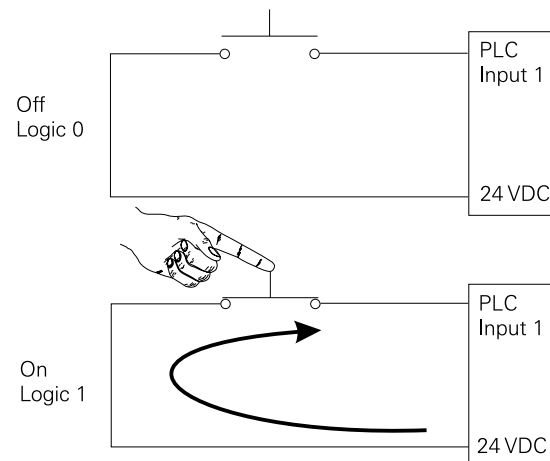


## Discrete Input

A discrete input, also referred to as a digital input, is an input that is either in an ON or OFF condition. Pushbuttons, toggle switches, limit switches, proximity switches, and contact closures are examples of discrete sensors which are connected to the PLCs discrete or digital inputs. In the ON condition a discrete input may be referred to as a logic 1 or a logic high. In the OFF condition a discrete input may be referred to as a logic 0 or a logic low.

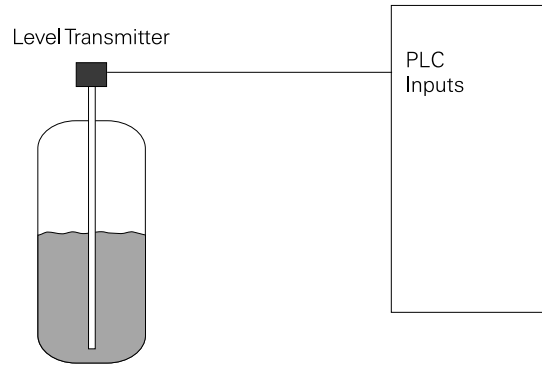


A Normally Open (NO) pushbutton is used in the following example. One side of the pushbutton is connected to the first PLC input. The other side of the pushbutton is connected to an internal 24 VDC power supply. Many PLCs require a separate power supply to power the inputs. In the open state, no voltage is present at the PLC input. This is the OFF condition. When the pushbutton is depressed, 24 VDC is applied to the PLC input. This is the ON condition.



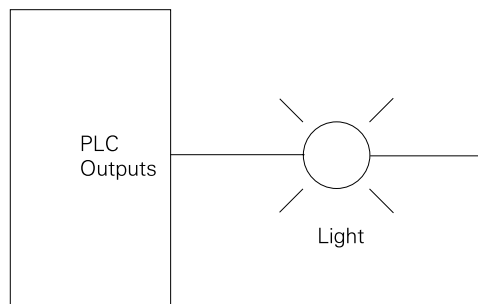
## Analog Inputs

An analog input is an input signal that has a continuous signal. Typical analog inputs may 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 tank. Depending on the level transmitter, the signal to the PLC can either increase or decrease as the level increases or decreases.



## Discrete Outputs

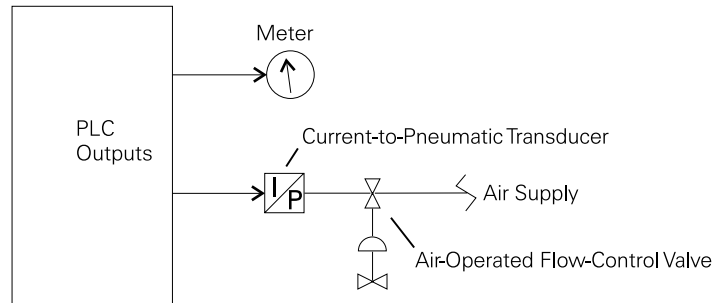
A discrete output is an output that is either in an ON or OFF condition. Solenoids, contactor coils, and lamps are examples of actuator devices connected to discrete outputs. Discrete outputs may also be referred to as digital outputs. In the following example, a lamp can be turned on or off by the PLC output it is connected to.





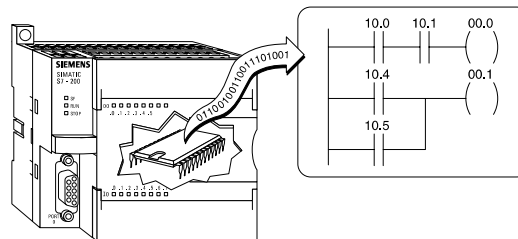
## Analog Outputs

An analog output is an output signal that has a continuous signal. The output may be as simple as a 0-10 VDC level that drives an analog meter. Examples of analog meter outputs are speed, weight, and temperature. The output signal may also be used on more complex applications such as a current-to-pneumatic transducer that controls an air-operated flow-control valve.



## CPU

The central processor unit (CPU) is a microprocessor system that contains the system memory and is the PLC decision-making unit. The CPU monitors the inputs and makes decisions based on instructions held in the program memory. The CPU performs relay, counting, timing, data comparison, and sequential operations.

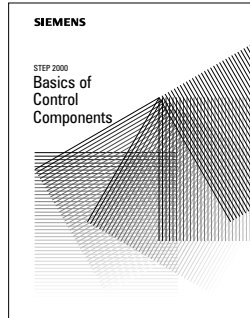


## Programming

A program consists of one or more instructions that accomplish a task. Programming a PLC is simply constructing a set of instructions. There are several ways to look at a program such as ladder logic, statement lists, or function block diagrams.

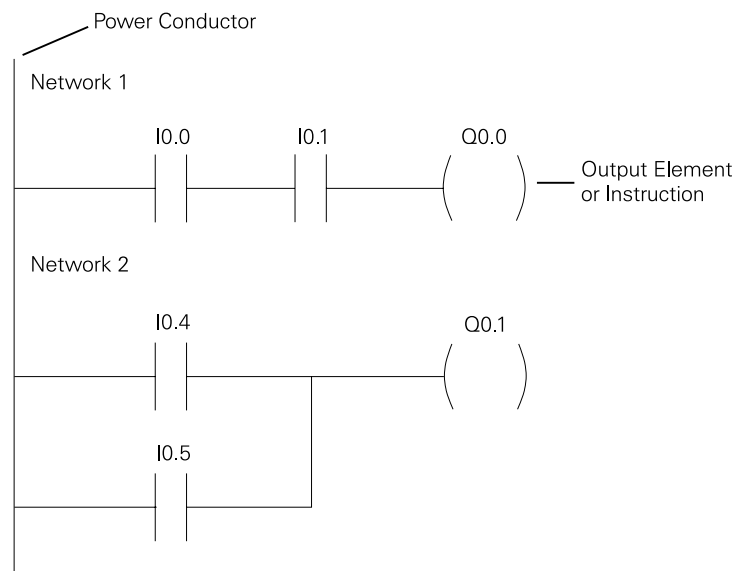
## Ladder Logic

Ladder logic (LAD) is one programming language used with PLCs. Ladder logic uses components that resemble elements used in a line diagram format to describe hard-wired control. Refer to the STEP 2000 course **Basics of Control Components** for more information on line diagrams.



## Ladder Logic Diagram

The left vertical line of a ladder logic diagram represents the power or energized conductor. The output element or 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, top-to-bottom. Rungs are sometimes referred to as networks. A network may have several control elements, but only one output coil.



In the example program shown example I0.0, I0.1 and Q0.0 represent the first instruction combination. If inputs I0.0 and I0.1 are energized, output relay Q0.0 energizes. The inputs could be switches, pushbuttons, or contact closures. I0.4, I0.5, and Q0.1 represent the second instruction combination. If either input I0.4 or I0.5 are energized, output relay Q0.1 energizes.

**Statement list**

A statement list (STL) provides another view of a set of instructions. The operation, what is to be done, is shown on the left. The operand, the item to be operated on by the operation, is shown on the right. A comparison between the statement list shown below, and the ladder logic shown on the previous page, reveals a similar structure. The set of instructions in this statement list perform the same task as the ladder diagram.

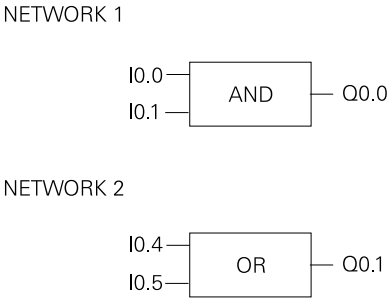
```

NETWORK 1
    LD      I0.0
    A      I0.1
    =      Q0.0

NETWORK 2
    LD      I0.4
    O      I0.5
    =      Q0.1
    
```

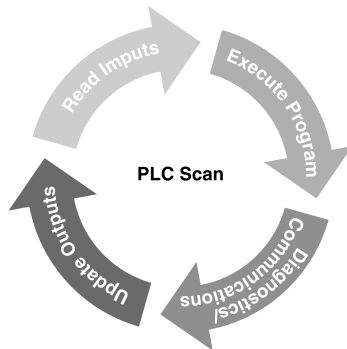
**Function Block Diagrams**

Function Block Diagrams (FBD) provide another view of a set of instructions. Each function has a name to designate its specific task. Functions are indicated by a rectangle. Inputs are shown on the left-hand side of the rectangle and outputs are shown on the right-hand side. The function block diagram shown below performs the same function as shown by the ladder diagram and statement list.



## 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. The application program is executed using the status of the inputs. Once the program is completed, the CPU performs internal diagnostics and communication tasks. The scan cycle ends by updating the outputs, then starts over. The cycle time depends on the size of the program, the number of I/Os, and the amount of communication required.



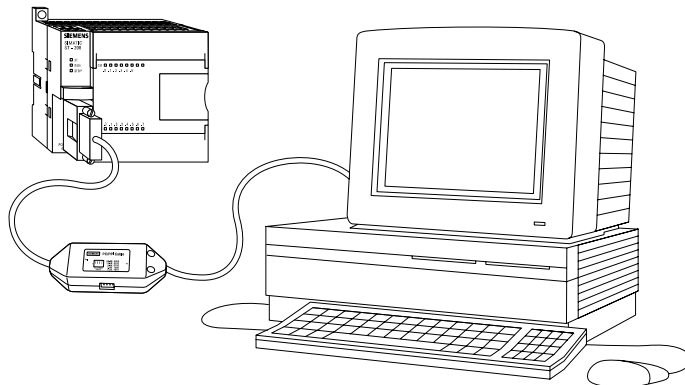
## Software

Software is any information in a form that a computer or PLC can use. Software includes the instructions or programs that direct hardware.



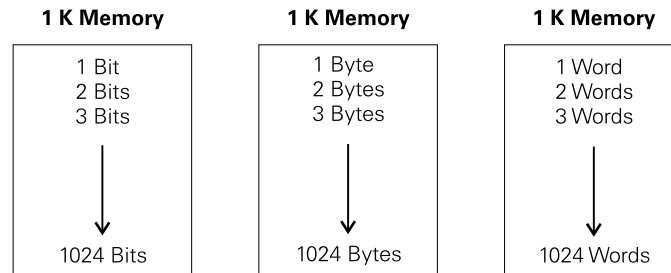
## Hardware

Hardware is the actual equipment. The PLC, the programming device, and the connecting cable are examples of hardware.



## Memory 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 ( $2^{10} = 1024$ ). This can be 1024 bits, 1024 bytes, or 1024 words, depending on memory type.



## RAM

Random Access Memory (RAM) is memory where data can be directly accessed at any address. Data can be written to and read from RAM. 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.

## ROM

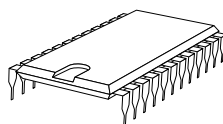
Read Only Memory (ROM) is a type of memory that data can be read from but not written to. This type of memory is used to protect data or programs from accidental erasure. ROM memory is nonvolatile. This means a user program will not lose data during a loss of electrical power. ROM is normally used to store the programs that define the capabilities of the PLC.

## EPROM

Erasable Programmable Read Only Memory (EPROM) provides some 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.

## Firmware

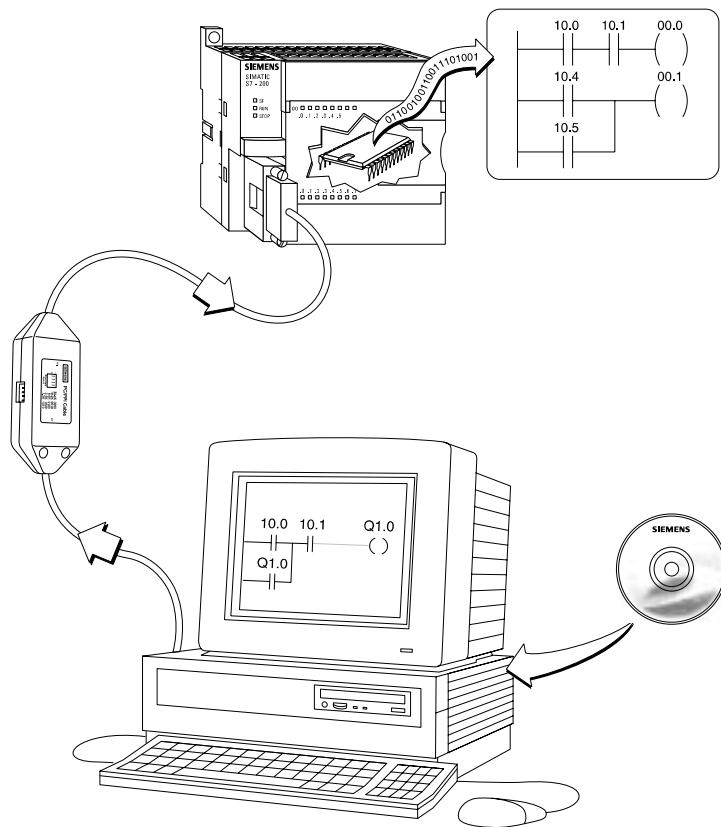
Firmware is user or application specific software burned into EPROM and delivered as part of the hardware. Firmware gives the PLC its basic functionality.



## Putting it Together

The memory of the S7-200 is divided into three areas: program space, data space, and configurable parameter space.

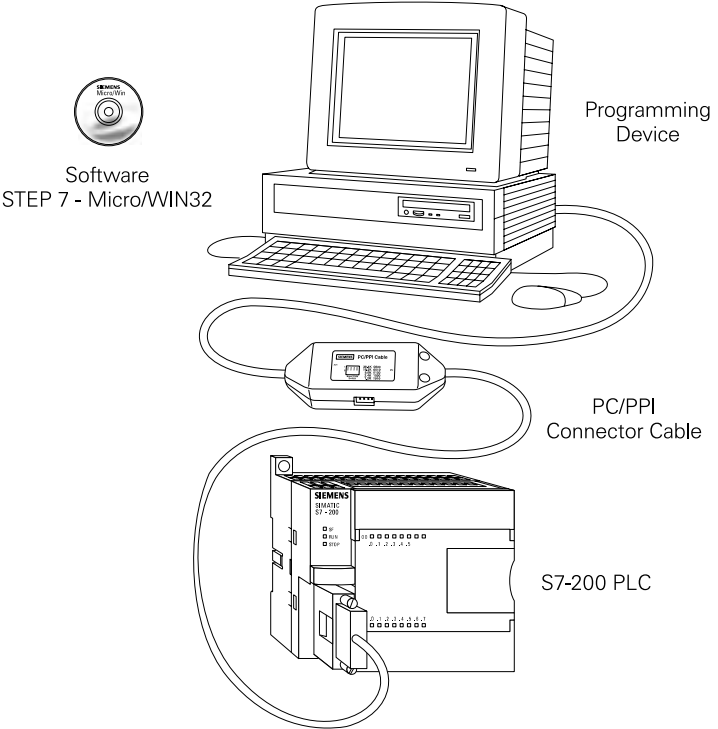
- Program space stores the ladder logic (LAD) or statement list (STL) program instructions. This area of memory controls the way data space and I/O points are used. LAD or STL instructions are written using a programming device such as a PC, then loaded into program memory of the PLC.
- Data space is used as a working area, and includes memory locations for calculations, temporary storage of intermediate results and constants. Data space includes memory locations for devices such as timers, counters, high-speed counters, and analog inputs and outputs. Data space can be accessed under program control.
- Configurable parameter space, or memory, stores either the default or modified configuration parameters.



# Basic Requirements

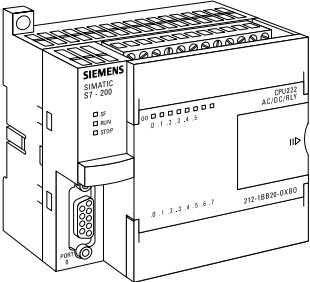
In order to create or change a program, the following items are needed:

- PLC
- Programming Device
- Programming Software
- Connector Cable



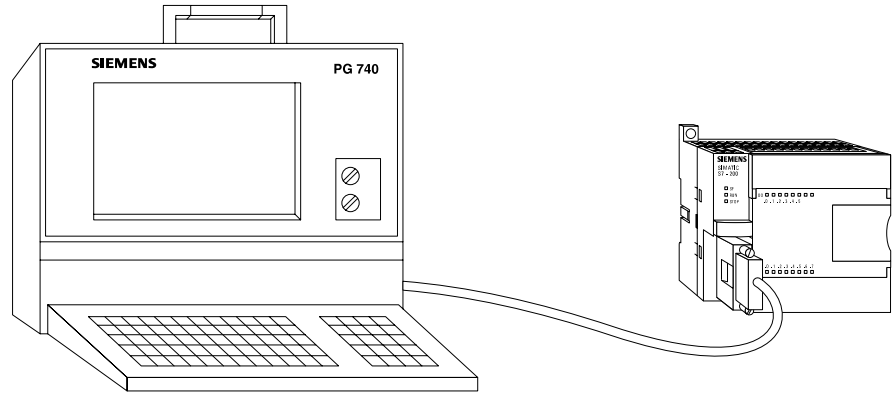
## PLC

Throughout this course we will be using the S7-200 because of its ease of use.

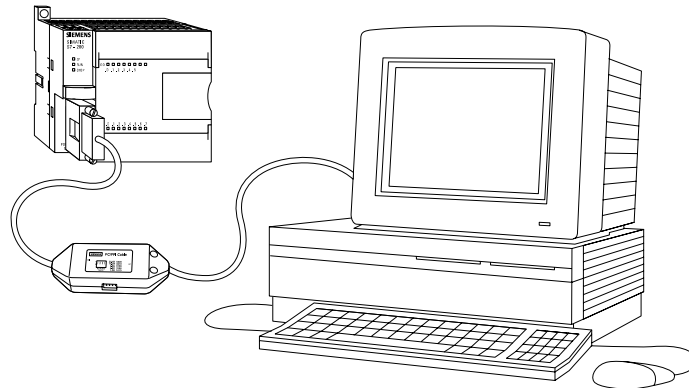


## Programming Devices

The program is created in a programming device (PG) and then transferred to the PLC. The program for the S7-200 can be created using a dedicated Siemens SIMATIC S7 programming device, such as a PG 720 (not shown) or PG 740, if STEP 7 Micro/WIN software is installed.



A personal computer (PC), with STEP 7 Micro/WIN installed, can also be used as a programming device with the S7-200.





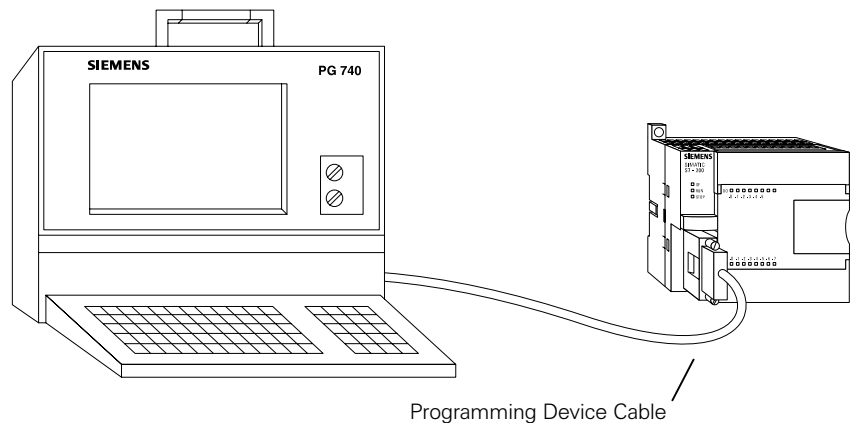
## Software

A software program is required in order to tell the PLC what instructions it must follow. Programming software is typically PLC specific. A software package for one PLC, or one family of PLCs, such as the S7 family, would not be useful on other PLCs. The S7-200 uses a Windows based software program called STEP 7-Micro/WIN32. The PG 720 and PG 740 have STEP 7 software pre-installed. Micro/WIN32 is installed on a personal computer in a similar manner to any other computer software.

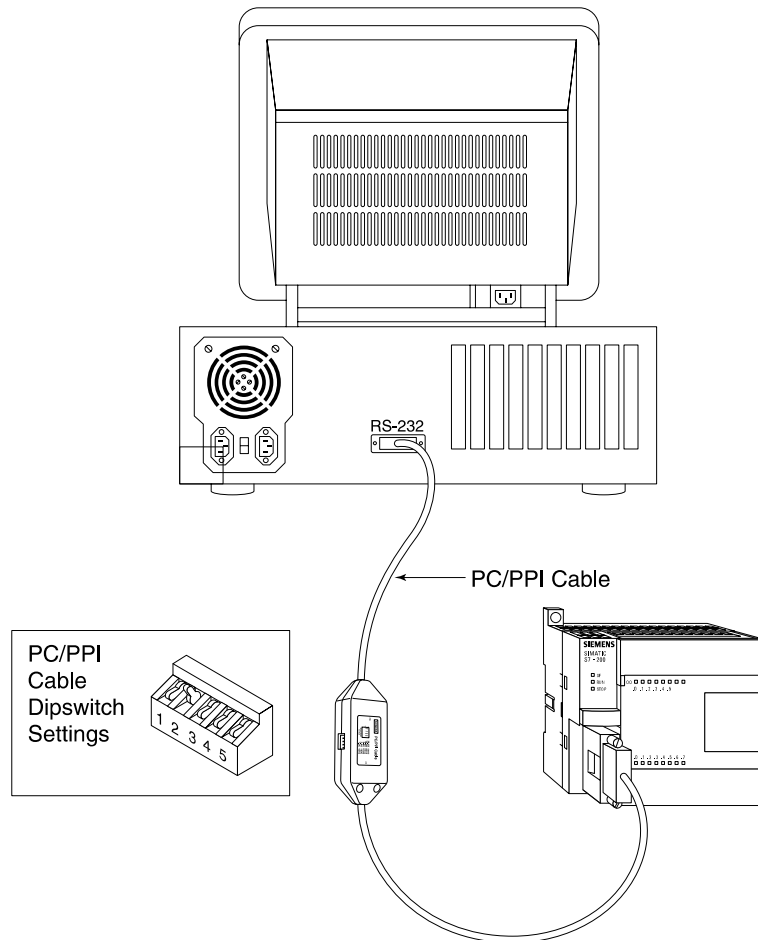


## Connector Cables PPI (Point-to-Point Interface)

Connector cables are required to transfer data from the programming device to the PLC. Communication can only take place when the two devices speak the same language or protocol. Communication between a Siemens programming device and the S7-200 is referred to as PPI protocol (point-to-point interface). An appropriate cable is required for a programming device such as a PG 720 or PG 740. The S7-200 uses a 9-pin, D-connector. This is a straight-through serial device that is compatible with Siemens programming devices (MPI port) and is a standard connector for other serial interfaces.



A special cable, referred to as a PC/PPI cable, is needed when a personal computer is used as a programming device. This cable allows the serial interface of the PLC to communicate with the RS-232 serial interface of a personal computer. DIP switches on the PC/PPI cable are used to select an appropriate speed (baud rate) at which information is passed between the PLC and the computer.



## Review 2

1. A switch or a pushbutton is a \_\_\_\_\_ input.
2. A lamp or a solenoid is an example of a \_\_\_\_\_ output.
3. The \_\_\_\_\_ makes decisions and executes control instructions based on the input signals.
4. \_\_\_\_\_ is a PLC programming language that uses components resembling elements used in a line diagram.
5. A \_\_\_\_\_ consists of one or more instructions that accomplish a task.
6. Memory is divided into three areas: \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_ space.
7. When talking about computer or PLC memory, 1K refers to \_\_\_\_\_ bits, bytes, or words.
8. Software that is placed in hardware is called \_\_\_\_\_.
9. Which of the following is not required when creating or changing a PLC program?
  - a. PLC
  - b. Programming Device
  - c. Programming Software
  - d. Connector Cable
  - e. Printer
10. A special cable, referred to as a \_\_\_\_\_ cable, is needed when a personal computer is used as a programming device.