

SIEMENS
Ingenuity for life

SIEMENS

DI a | 0 .1 .2 .3 .4 .5 .6 .7 | DI b | 0 .1 .2 .3 .4 .5
CPU 1214C
DC/DC/DC
DQ b | 0 .1 |
6 .7

Basics of PLCs

A quickSTEP Online Course

© Siemens Industry, Inc.

www.usa.siemens.com/step

Trademarks

Siemens and SIMATIC are trademarks of Siemens AG. Other trademarks are the property of their respective owners.

Course Topics



Welcome to Basics of PLCs. This course covers the following topics:

Chapter 1 - Introduction

- Overview
- Basic Concepts

Chapter 2 - S7-1200 PLCs

- S7-1200 PLC Overview
- S7-1200 with Safety Integrated

Chapter 3 - S7-1200 PLC Programming

- Programming Concepts
- LAD Programming Basics
- LAD Timers and Counters

Chapter 4 - Additional Information

- Additional S7-1200 Capabilities
- Additional Products

If you do not have an understanding of basic electrical concepts, you should complete Basics of Electricity before attempting this course.

Course Objectives

Upon completion of this course you will be able to...

- Describe important number systems and data types used by Siemens PLCs.
- Identify the major components of a PLC and describe their functions.
- Describe the parts of a typical PLC scan.
- Define the terms “functional safety” and “failsafe.”
- Identify the key features of S7-1200 standard and failsafe CPU models.
- Identify the types of S7-1200 PLC signal and communication modules.
- Summarize the capabilities of TIA Portal.
- Summarize the capabilities of STEP 7 (TIA Portal).
- Describe what is meant by “security integrated.”
- List the programming languages available for S7-1200 PLCs.
- Describe the concept of modular programming and the types of program and data blocks available for Siemens PLCs.
- Describe the operation of the most commonly used ladder diagram programming instructions.
- Describe what is meant by S7-1200 integrated technologies.

SIMATIC S7-1200 PLCs



Automation solutions must be compact, scalable, and flexible. Siemens SIMATIC S7-1200 PLCs are available as standard and failsafe versions. In addition to helping you learn basic PLC concepts, this course will help you understand the capabilities of S7-1200 PLCs, which can be optimally adapted to your individual requirements with pluggable signal and communication boards and modules.

SITRAIN® Training for Industry



Online Self-paced Learning – Programs with maximum flexibility so students can easily fit courses into their busy schedules



Virtual Instructor-led Learning - Classroom lectures delivered in the convenience of your home or office



Classroom Learning - Expert and professional instructors, proven courseware, and quality workstations combine for the most effective classroom experience possible at your facility or ours



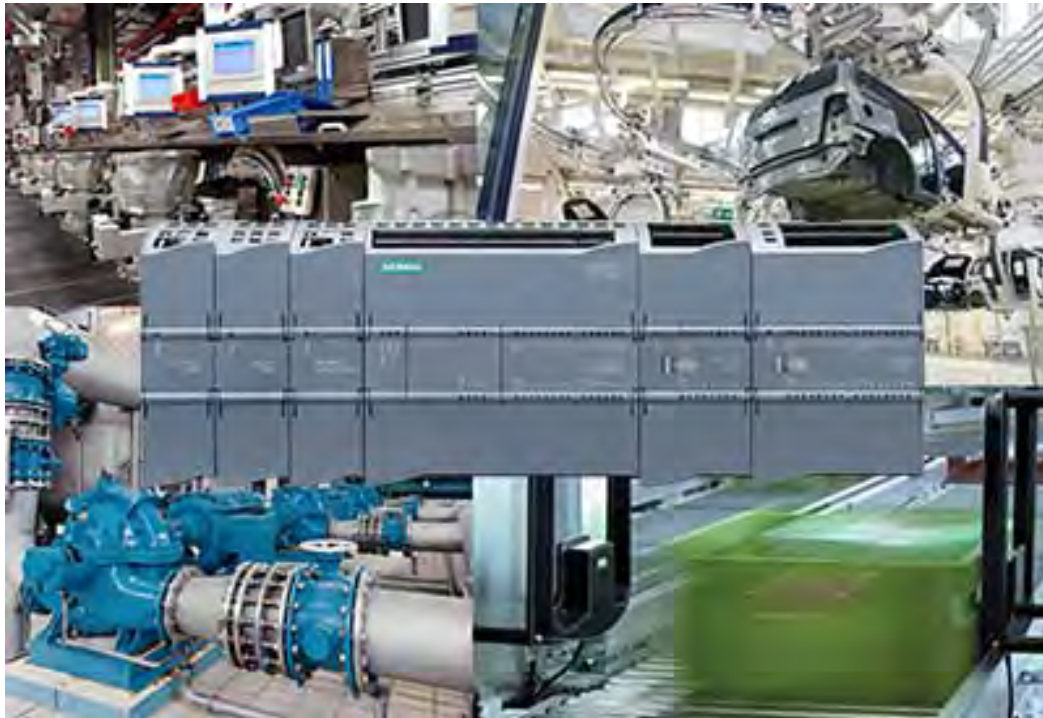
How-to Video Library - Quick, affordable, task-based learning options for a broad range of automation topics for training or purchase



Simulators - World-class simulation systems available for training or purchase

This course also describes learning options available from the Siemens SITRAIN USA organization and our global SITRAIN partners. For additional information: www.usa.siemens.com/sitrain

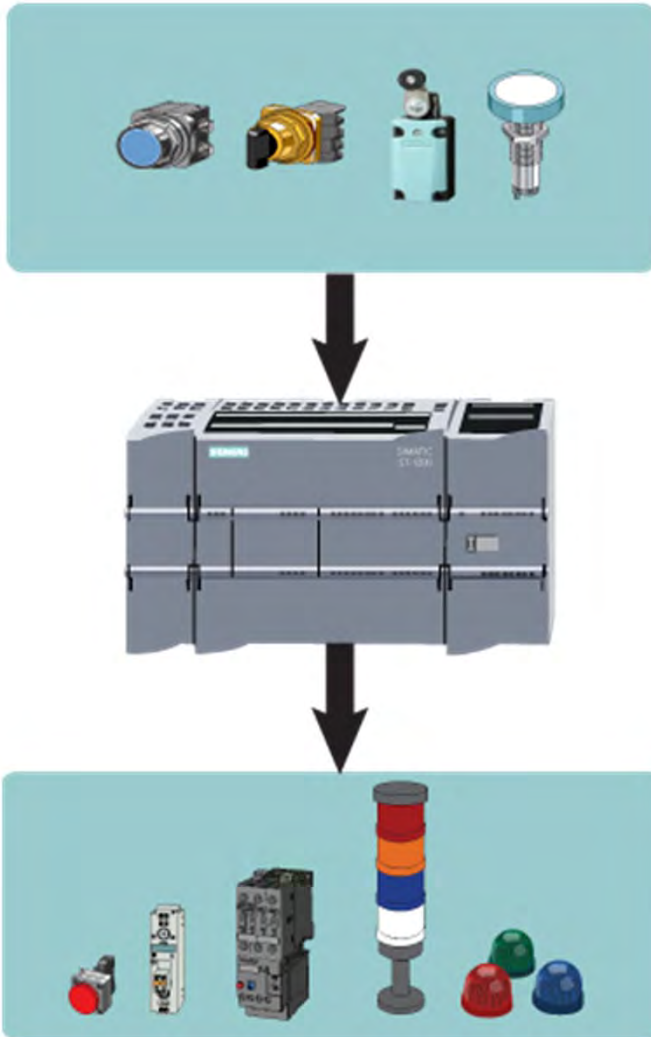
Chapter 1 - Introduction



This chapter covers the following topics:

- **Overview**
- **Basic Concepts**

Definition

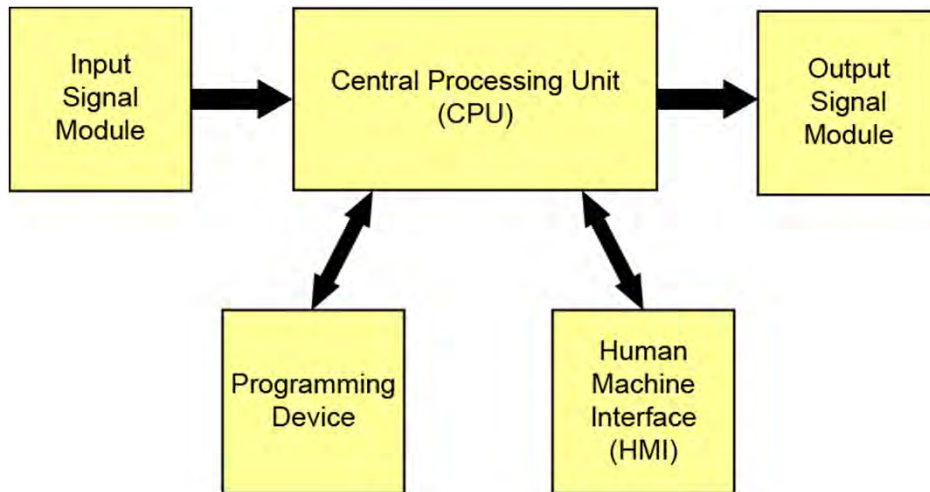


A programmable logic controller (PLC), also referred to as a programmable controller, is a type of computer commonly used in commercial and industrial control applications.

PLCs differ from office computers in the types of tasks 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 machine or process.

There are many types of PLCs, and they vary significantly in appearance and capabilities. Therefore, to keep this course simple, the examples used focus primarily on Siemens SIMATIC S7-1200 basic controllers. Although S7-1200 PLCs offer basic automation solutions, they also have many advanced features, and learning about S7-1200 PLCs will help you gain a good understanding of PLC capabilities.

Basic PLC Operation



The basic components of a PLC include input signal modules, a central processing unit (CPU), output signal modules, and a programming device. The types of input and output signal modules used by a PLC depend upon the types of input and output devices used.

Input signal modules convert the signals provided by input devices into logic signals that can be used by the CPU.

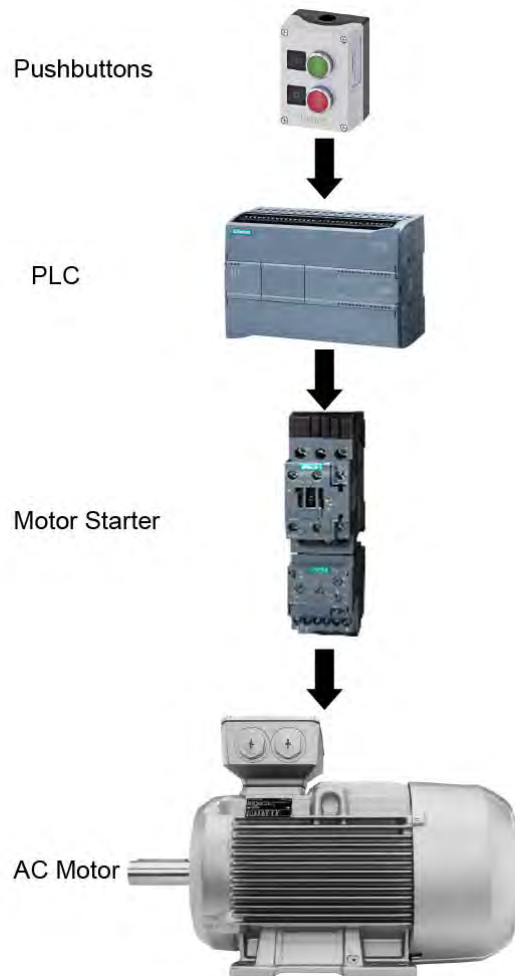
The CPU uses the values of inputs, outputs, and other variables as it executes the user program stored in its memory. The CPU then sends signals to update the statuses of outputs.

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

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

A control system may also incorporate one or more human machine interfaces (HMIs) to monitor and control a machine or process. HMIs are not PLC components, but work closely with the PLC.

Simple Example

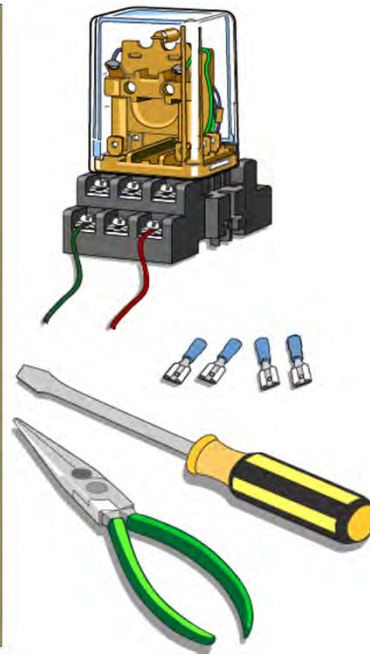
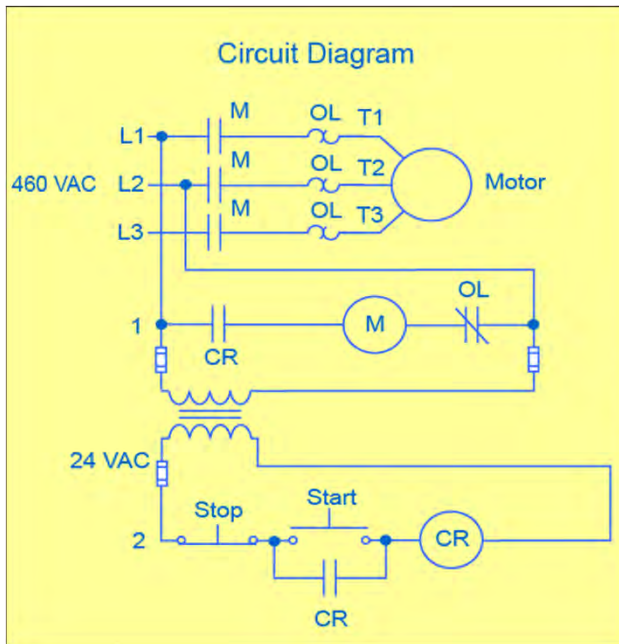


In this simple example, pushbuttons connected to PLC inputs are used to start and stop a motor connected to a PLC output through a motor starter.

No programming device is shown in this example because, once the PLC has been programmed, the PLC can perform its control tasks without the programming device.

Similarly, an HMI is not shown, because this is a simple control example. However, additional outputs from the PLC may control indicator lights that show whether the motor is stopped or running or indicate a fault, such as a motor overload.

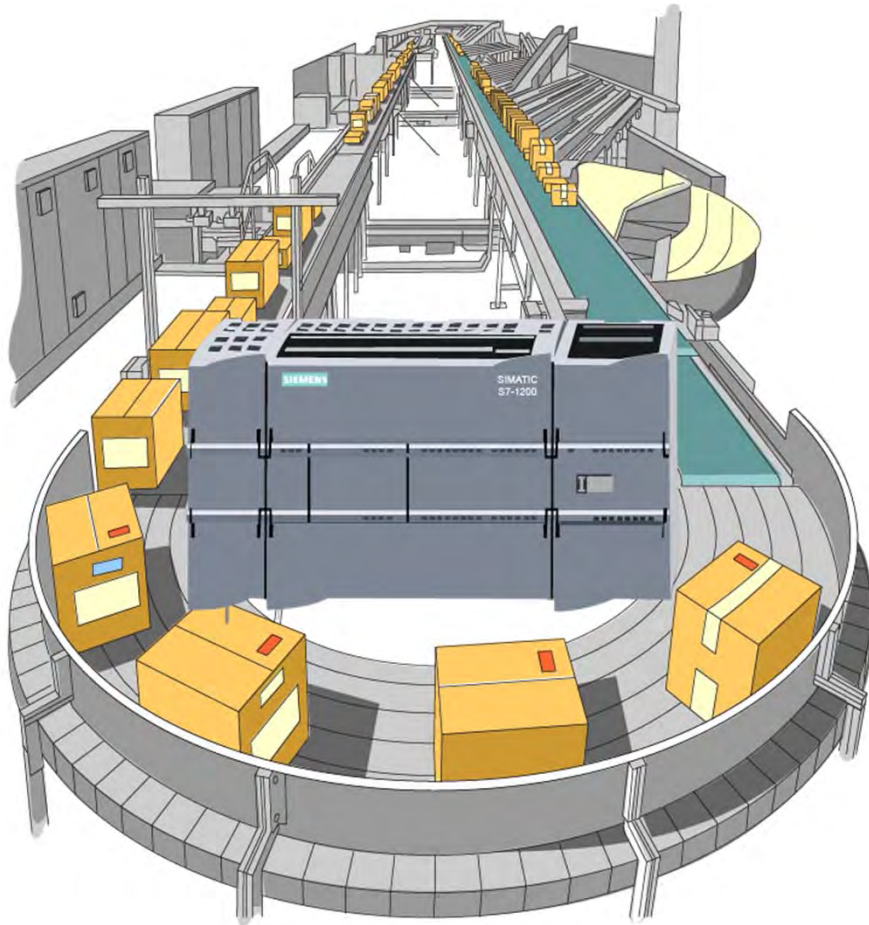
Hard-Wired Control



Prior to PLCs, control tasks were often performed by contactors, control relays, and other electromechanical devices with intricate interconnecting wires. This approach is often referred to as hard-wired control.

Although hard-wired control solutions are capable of performing some of the same tasks as PLCs, hard-wired control is generally more difficult to design, install, and maintain. In addition, the process of making even simple modifications to a hard-wired control solution can be difficult because the logic of the control system is determined by the interconnection of control wires and components.

Advantages of PLCs



Some of the advantages of PLCs when compared to hard-wired solutions are as follows:

- PLCs can perform more complex control tasks
- PLCs can communicate with other systems
- PLC systems are more reliable
- PLC systems can be more easily and more effectively documented
- PLC systems are easier to operate and maintain
- PLC system changes are easier to implement
- PLC applications can be duplicated faster and less expensively

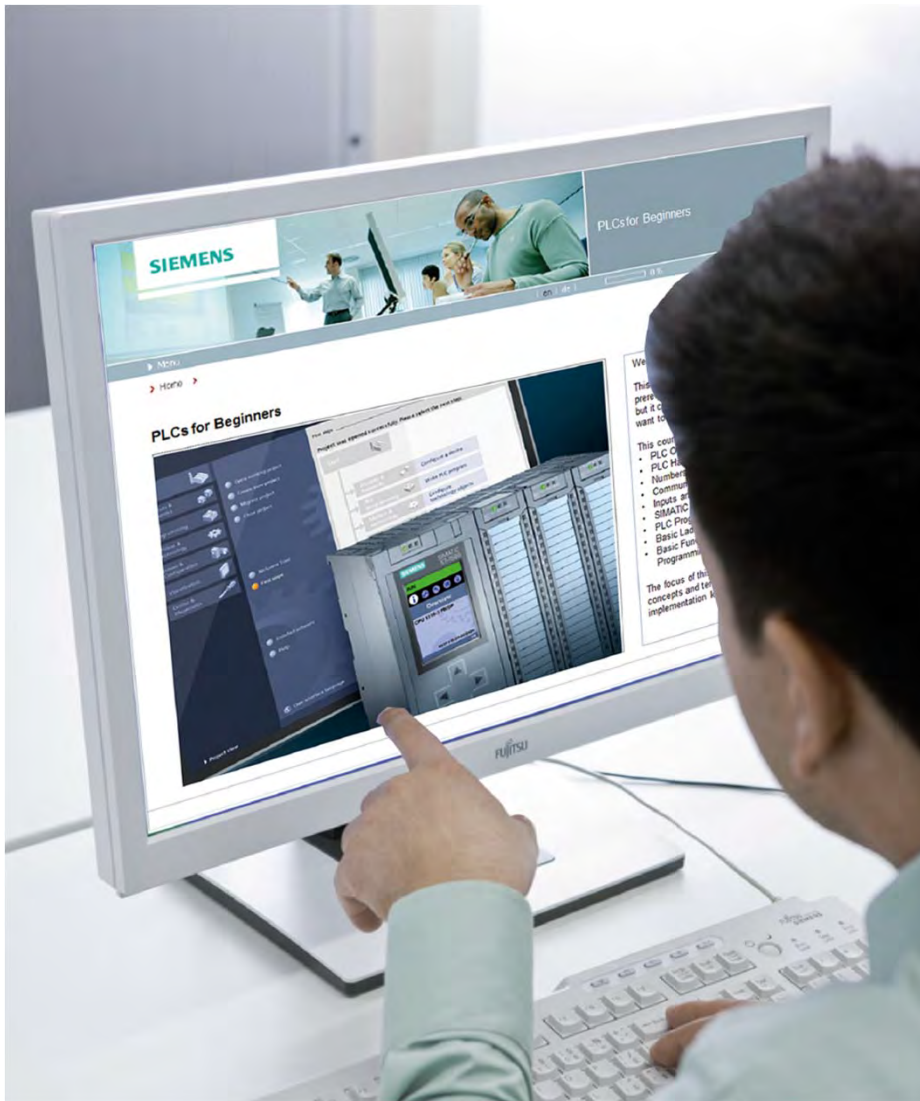
SIMATIC Products



Siemens SIMATIC products are the foundation upon which our Totally Integrated Automation (TIA) concept is based. Because the needs of end users and machine builders vary widely, the SIMATIC family includes a wide range of controllers, human machine interfaces (HMIs), and related products. For example, SIMATIC PLCs are available as conventional modular controllers, embedded automation products, and as PC-based controllers.

Modular SIMATIC S7 PLCs 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: S7-1200 basic controllers, ET 200SP distributed controllers, and S7-1500 advanced controllers. The focus of this course is on S7-1200 basic controllers.

Online Self-paced Learning



With Siemens online self-paced learning, students select the topics and set their own pace for completing chosen courses. All course material can be accessed online. Instruction starts upon completing the purchase of a subscription.

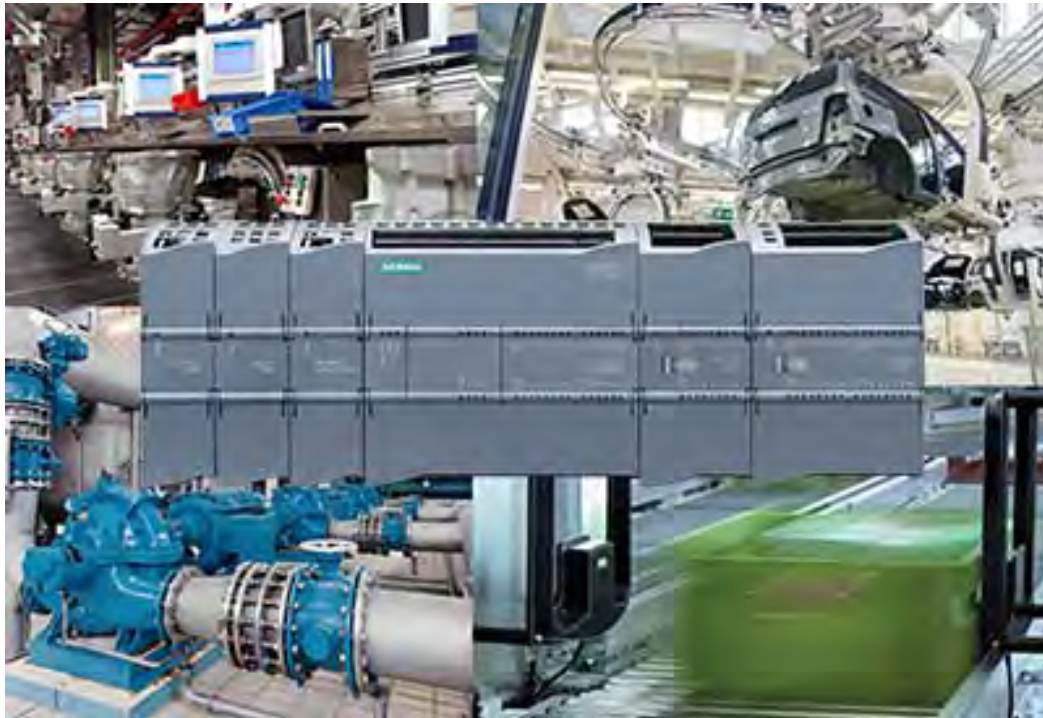
Students can choose from over 500 courses consisting of high-quality graphics, on-screen text, supporting voiceover narration, and interactive exercises. Features include printable course content for reference and underlined key vocabulary terms with definitions displayed with a simple mouse-over action.

Depending on the subscription purchased, you can choose any 10 or 25 courses or select from the entire online self-paced course catalog.

These courses are offered 24/7/365, so students can begin a subscription at any time. From the date of registration, students are given one year to complete their course selections.

For additional information: www.usa.siemens.com/sitrain

Chapter 1 - Introduction



This chapter covers the following topics:

- Overview
- **Basic Concepts**

Binary Number System

Decimal Number System

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

Binary Number System

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

8-Bit Binary Number Example

Most Significant Bit (MSB)				Least Significant Bit (LSB)			
↓							↓
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

0001100 in binary = 24 in Decimal

Because a PLC is a computer, it stores information in the form of on and off conditions represented by ones and zeros, 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.

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 position is referred to as the most significant bit or MSB, and the number in the far right-hand position 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.

The accompanying graphic shows an 8-bit binary number, but the number of bits used varies.

Converting from Binary to Decimal

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

$$\text{Decimal Value} = 16 + 8 = 24$$

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

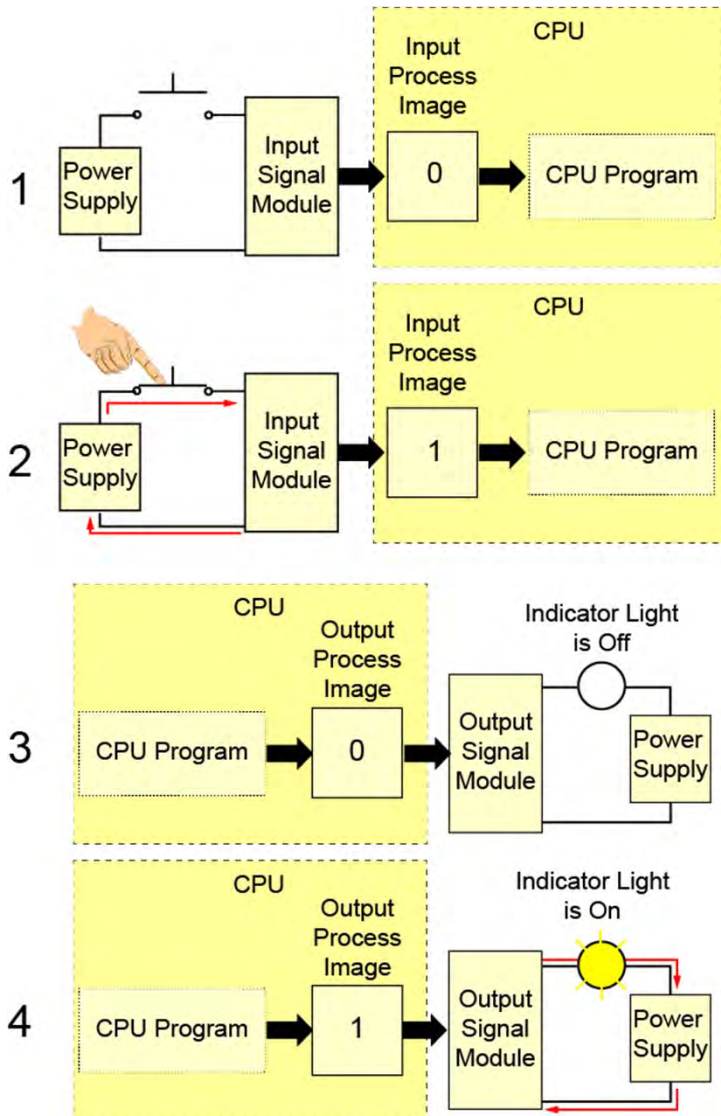
$$\text{Decimal Value} = 32 + 8 + 1 = 41$$

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

$$\text{Decimal Value} = 128 + 16 + 2 = 146$$

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.

Logic 0 and 1

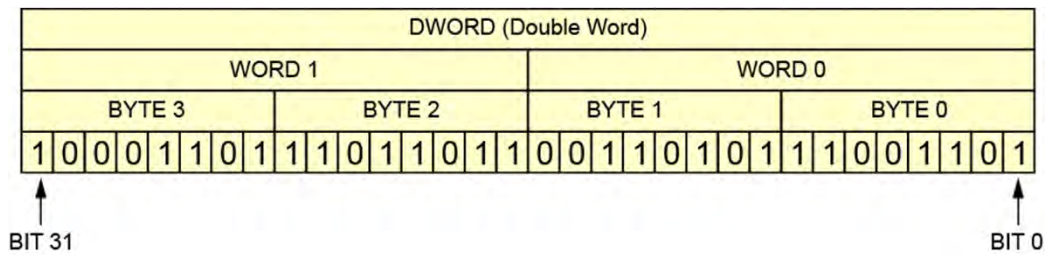


While PLCs are capable of sensing and generating analog values, internally, PLCs use signals that are off or on. These off and on conditions correspond to the binary values 0 and 1, also referred to as logic 0 and logic 1.

For example, as shown in graphic 1, when an input to a PLC is off, a 0 is stored in the corresponding position in the CPU's input process image. And, as shown in graphic 2, when the input is on, a 1 is stored in that position in the input process image. If that input is used in the CPU program, the status or change in status of the input process image bit can trigger other events.

As shown in graphics 3 and 4, the CPU program can cause an output process image status bit to be a 0 or a 1. If the corresponding output channel on an output signal module is wired to an output device, the output device will be off when the status bit is a 0 and on when the status bit is a 1.

Bits, Bytes, Words, and Double Words



Some operations performed by a PLC use individual binary bits. Other PLC operations group binary bits to represent numerical values or various conditions. The number of bits used by an operation depends on the instruction.

Some instructions operate on 8 consecutive binary bits, referred to as a byte. Other instructions operate on larger groups such as a word (16 consecutive binary bits) or double word (32 consecutive binary bits).

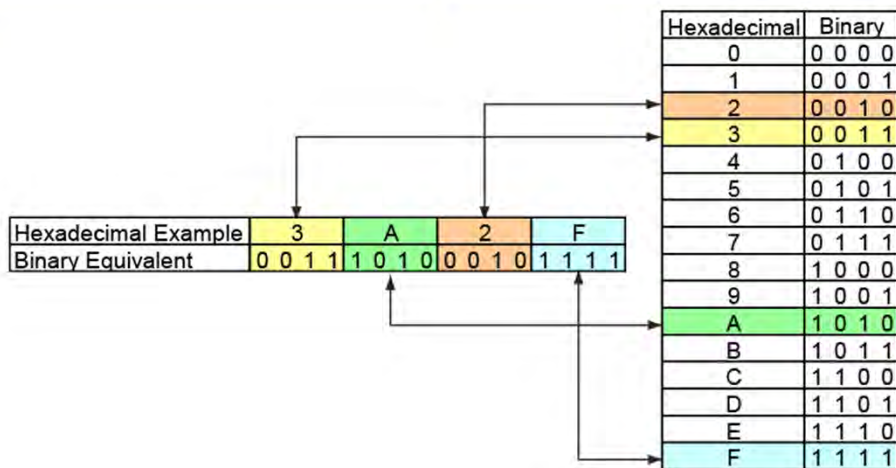
Hexadecimal

Binary Number System

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

Hexadecimal Number System

16 characters: 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, ...)



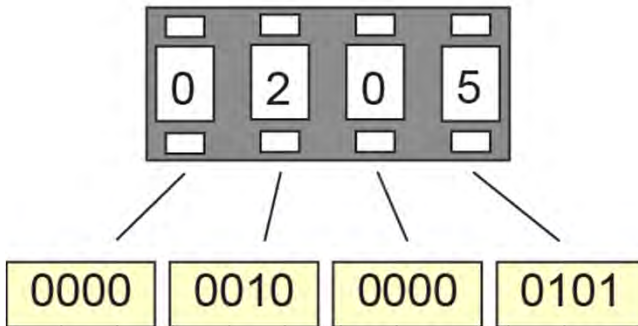
Hexadecimal is another number system used by PLCs. Each position in a hexadecimal number represents a power of 16.

The hexadecimal system uses 16 characters. The ten digits of the decimal system are used to represent the first ten characters of the hexadecimal system. The first six letters of the alphabet are used for the remaining six characters.

One of the reasons the hexadecimal system is used by PLCs is because it allows the statuses 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 is equivalent to a four-bit binary value. If you know the corresponding binary bits for each of the hexadecimal characters, you can quickly convert a hexadecimal number of any length to binary. The accompanying example, shows the equivalent binary value for the hexadecimal number 3A2F.

Binary Coded Decimal (BCD)



Decimal	BCD
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

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 with each decimal digit corresponding to four PLC digital 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.

Data Types

Binary Numbers	Short Form	Size
Boolean	Bool	1 Bit
Byte	Byte	8 Bits
Word	Word	16 Bits
Double Word	Dword	32 Bits
Integers		
Unsigned Short	USInt	8 Bits
Unsigned	UInt	16 Bits
Unsigned Double	UDInt	32 Bits
Signed Short	SInt	8 Bits
Signed	Int	16 Bits
Signed Double	DInt	32 Bits
Real Numbers		
Real Number	Real	32 Bits
Long Real Number	LReal	64 Bits

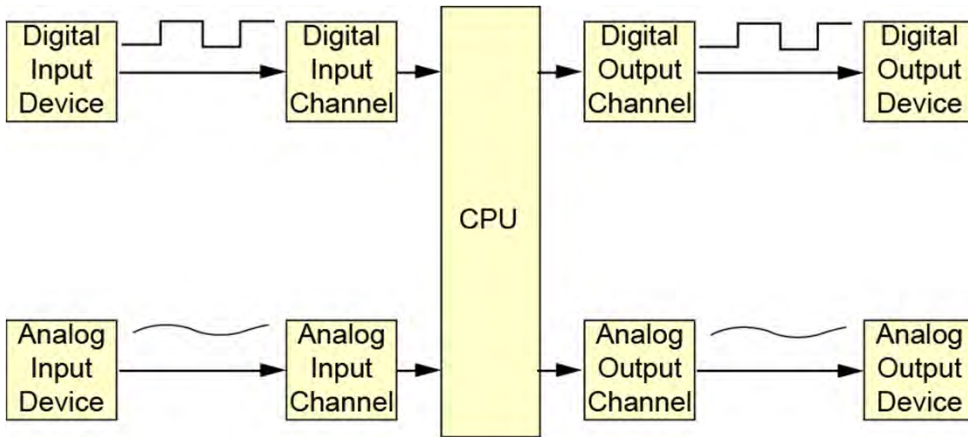
Binary bits can be used independently or as a group. When used as a group, they are used to represent numerical values as well as other types of data. In order to know how a bit or bit string will be interpreted by a PLC, you must know the data type, which is the PLC's description of the data.

PLC data types of various lengths are specified for binary numbers, integers (also called whole numbers), real numbers (also called floating point numbers), date and time, characters, parameters, system data, and other types of data.

Because the number of data types has increased overtime, not all data types are available for all SIMATIC S7 PLCs. The accompanying graphic shows the binary number, integer, and real number data types available for S7-1200 PLCs.

Real numbers are numbers expressed in scientific notation, To convert a decimal number to scientific notation move the decimal point to the left or right to create a mantissa that is greater than or equal to 1, but less than 10. Then multiply the mantissa by a power of ten to compensate for the decimal point movement.

Inputs and Outputs



Input devices are usually switches or sensors that send electrical signals to the CPU through input channels on input signal modules. Similarly, output devices receive electrical signals from the CPU through output channels on output signal modules. These output devices are sometimes called actuators and are used to control a machine or process.

PLCs have two broad categories of inputs and outputs (I/O), digital I/O and analog I/O. Digital I/O devices, also called discrete I/O devices, are either on or off and are connected to digital I/O channels. Analog I/O devices use continuously variable voltage or current signals and are connected to analog I/O channels.

Each analog input signal is converted by an analog input signal module to a stream of numerical values represented in binary. This is necessary for the CPU to process this information. Because each analog output device requires a variable voltage or current signal, the stream of numerical values provided from the CPU is converted by an analog output signal module to an analog signal compatible with the associated device.

CPU



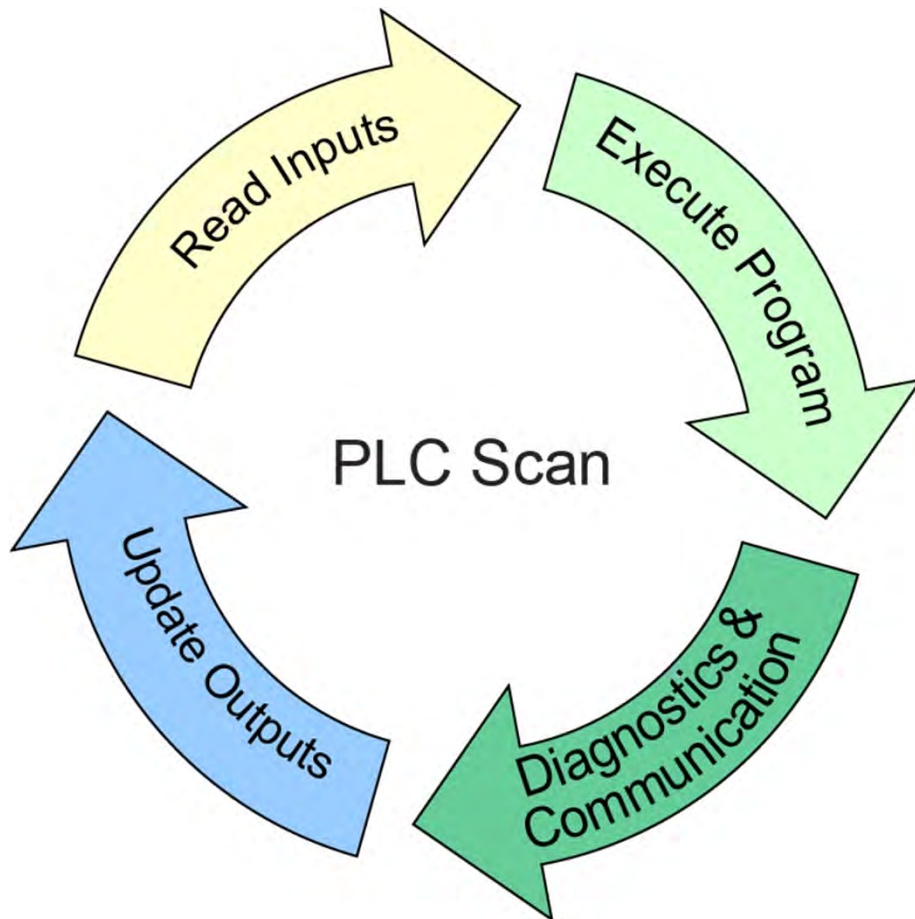
S7-1200 CPU

The central processing unit (CPU) is a microprocessor-based system that contains the system memory and is the PLC's decision-making unit.

The CPU monitors the inputs, outputs, and other variables and makes decisions based on instructions stored in its user program memory.

Some SIMATIC S7 CPUs have input and output points in the same enclosure with the CPU. For example, the S7-1200 CPU shown in the accompanying graphic has 14 digital inputs, 10 digital outputs, 2 analog inputs, and 2 analog outputs.

CPU Scan



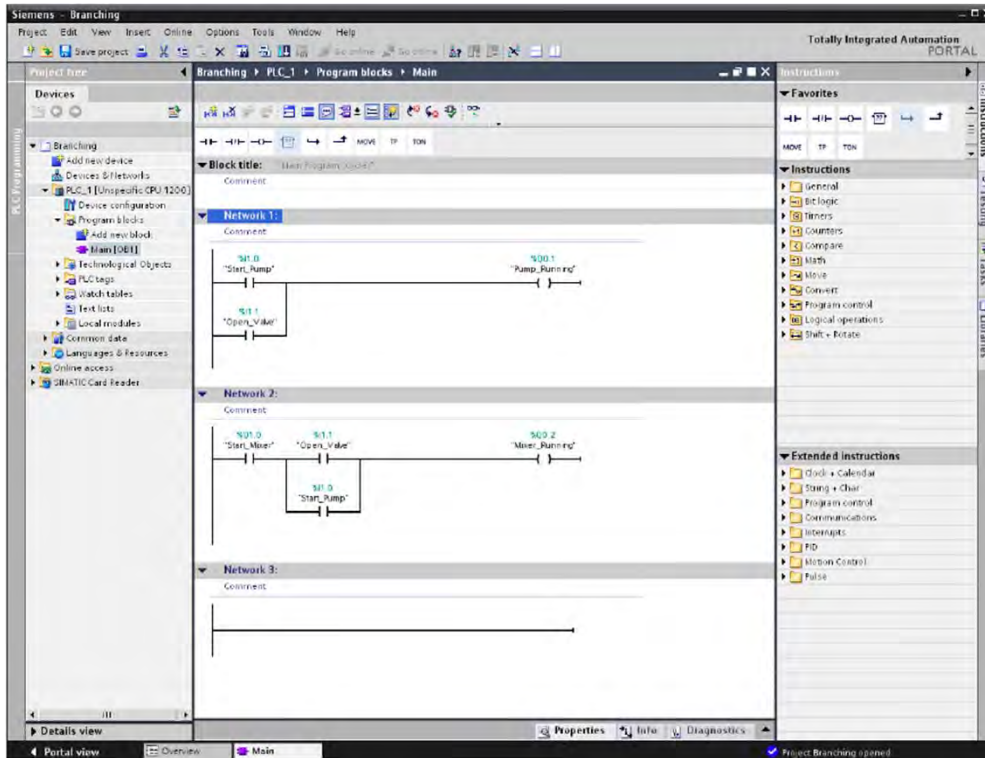
The PLC's user program is executed by the CPU as part of a repetitive process referred to as a scan. After startup, a typical CPU scan includes the following steps.

- The CPU reads the statuses of inputs.
- The CPU executes the user program.
- The CPU performs internal diagnostic and communication tasks.
- The CPU updates the statuses of outputs.

This process is repeated continuously as long as the PLC is in the run mode.

The time required for a scan depends on the capabilities of the CPU, the size of the user program, the number of inputs and outputs, and the amount of communication required. However, because PLC CPUs are very fast, this time is typically measured in milliseconds. This means that the response time for a PLC is also very fast.

User Program



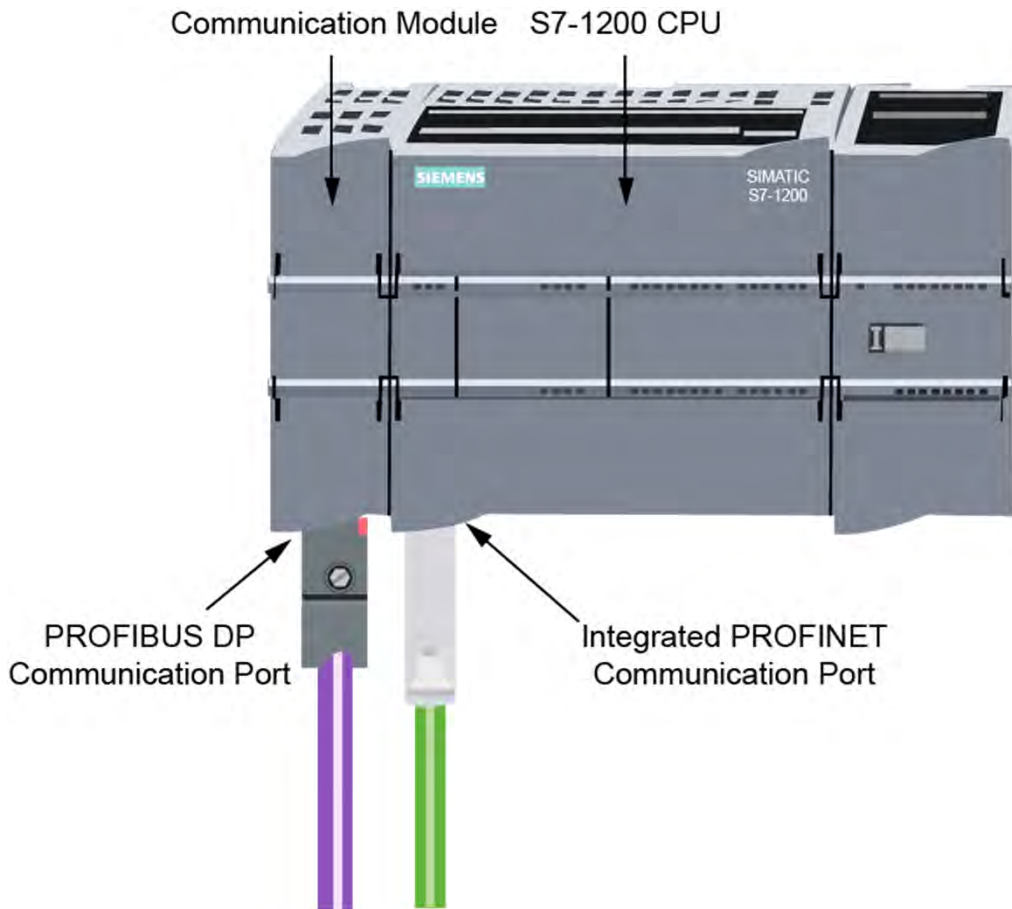
Ladder Diagram (LAD) Program

The first PLCs were designed for use in the automotive industry in the late 1960s. Prior to that time, control of an auto assembly line relied heavily on electromechanical relays, contactors, timers, and related devices.

Because the hard-wired circuits replaced by PLCs used control circuit diagrams referred to as ladder diagrams, early PLCs used ladder diagram software programs, sometimes referred to as ladder logic, to make it easier for someone familiar with control circuits to program a PLC. Unfortunately, every PLC manufacturer had its own version of ladder diagram programming. This variation in programming grew as PLCs were developed to handle a wider range of tasks.

Today, most PLCs can still use ladder diagram (LAD) programming, but the IEC 61131 international standard now defines this PLC programming language. This same standard also defines other types of programming languages available for PLCs, so that they can be used for the even the most complex applications.

PLC Communication



PLCs use a variety of communication technologies. The most basic type of communication used is serial communication, where bits are sent and received one at a time. Serial communication is still used with some devices; however, more often, PLCs use network communication.

For example, Industrial Ethernet is a high-performance network that uses industrial-grade switching technology. An Industrial Ethernet switch is an active network component that allows multiple devices to communicate simultaneously at high speeds.

PROFINET is an open Industrial Ethernet standard and the leading Industrial Ethernet standard world-wide. PROFINET IO, the most widely-used form of PROFINET, handles both non-time-critical IT communications and the full range of real-time control communications.

Another network type used is PROFIBUS, which is an open fieldbus standard. A fieldbus is a multi-drop network that provides a standardized approach for communication with devices commonly used for factory automation or process control. The version of PROFIBUS most widely used in factory automation applications is PROFIBUS DP.

Virtual Instructor-led Learning



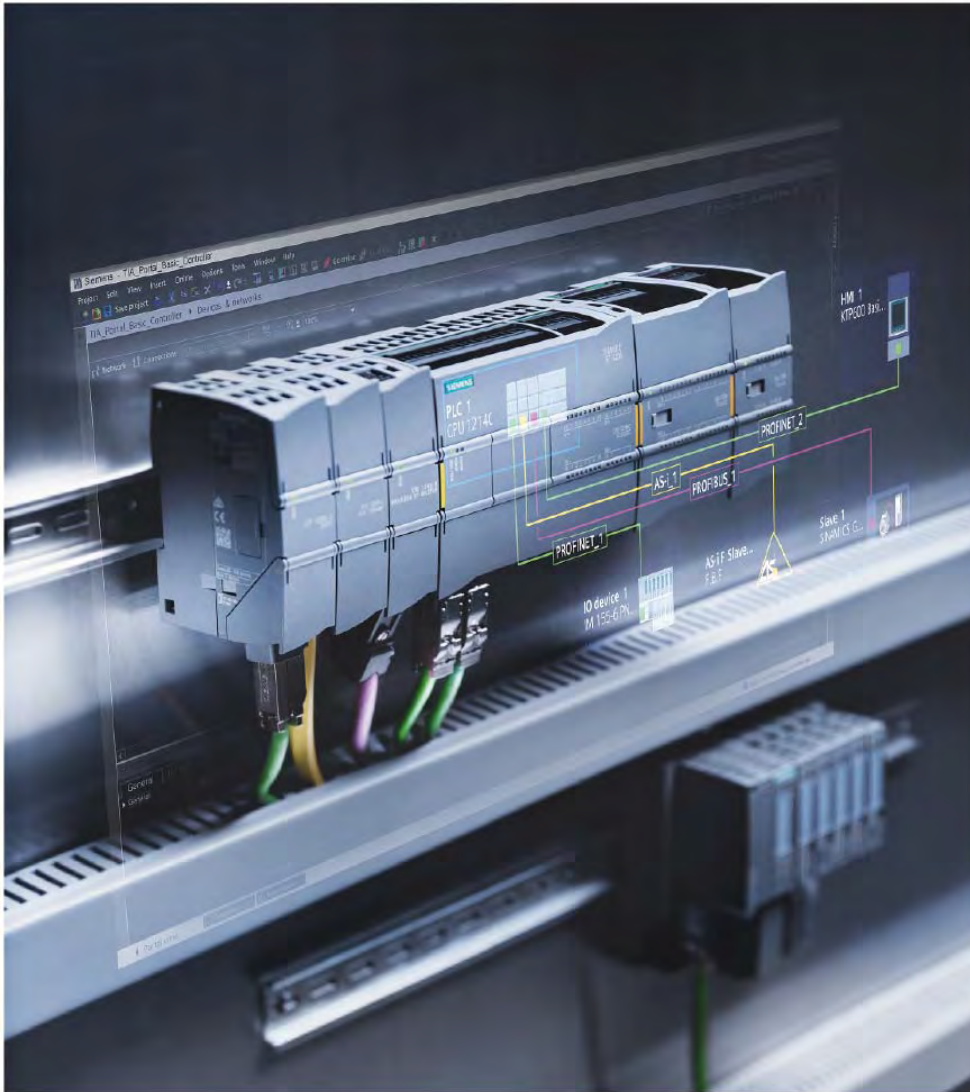
Siemens virtual instructor-led courses offer students a live, classroom experience with the convenience and cost savings of online learning. These courses provide hands-on instruction and live interaction, delivered anywhere an internet connection is available.

Scheduled courses are typically 10-hour agendas presented Monday through Friday in two-hour sessions. These sessions provide students with lecture, demonstration, lab exercises, and Q&A sessions – all presented by Siemens subject matter experts.

For the full course duration, students can complete assignments and reinforce classroom instruction using a virtual cloud-based application providing 24/7 access to fully functional Siemens software such as SIMATIC STEP 7 and PLCSIM.

For additional information: www.usa.siemens.com/sitrain

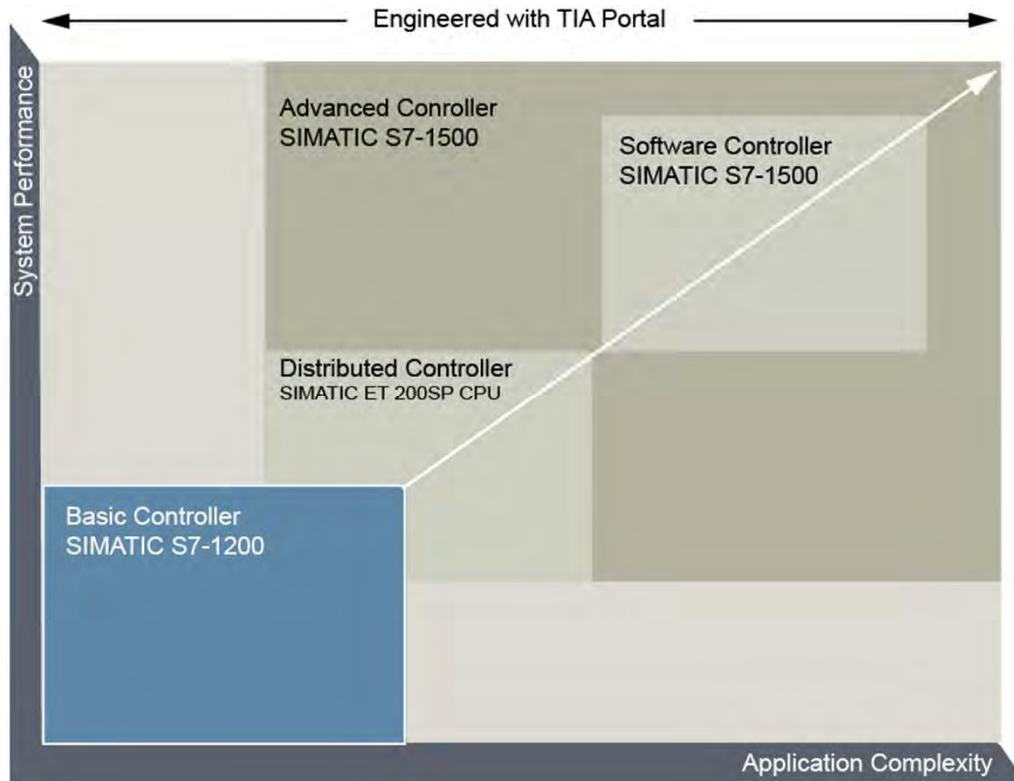
Chapter 2 - S7-1200 PLCs



This chapter covers the following topics:

- **S7-1200 PLC Overview**
- **S7-1200 with Safety Integrated**

S7-1200 PLC System Overview

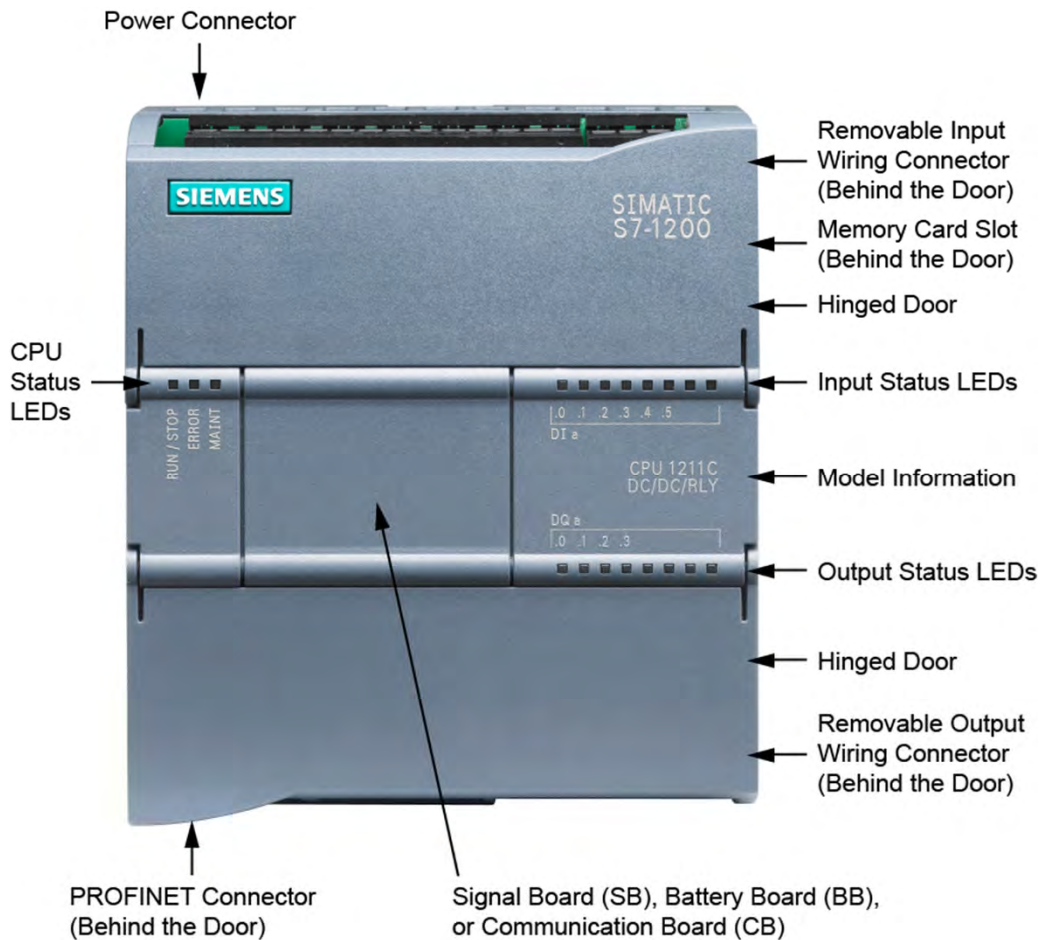


SIMATIC S7-1200 basic controllers are part of the SIMATIC S7 family of PLCs and are designed to handle a wide variety of small to medium-sized automation tasks.

Like more advanced S7 PLCs, S7-1200 controllers are configured and programmed using TIA Portal software which allows efficient engineering approaches to be employed for Siemens S7 PLCs, HMIs, and other automation products.

Because automation tasks vary in complexity, Siemens offers other controller options that allow common design concepts and approaches to be employed over the full range of applications.

S7-1200 CPU Features



S7-1200 CPUs are available in three versions, standard CPUs, failsafe CPUs (described later in this course), and SIPLUS CPUs (for use in extreme environmental conditions).






S7-1200 CPUs have internal memory for various functions including load memory, which stores the user program, data, and configuration information. Alternatively, load memory can be stored on a SIMATIC memory card installed in a slot behind the upper hinged door.

S7-1200 CPUs have one or two PROFINET ports.

S7-1200 CPUs have a small number of inputs and outputs in the same enclosure with the CPU. By flipping down the hinged doors, you can easily gain access to removable input and output wiring connectors. Status LEDs associated with the I/O points and CPU status LEDs are visible on the front of the CPU.

A signal board can be installed in the CPU to add digital or analog input or output channels. Alternatively, a battery board can be installed to provide long term backup for the CPU's real-time clock, or a communications board can be installed to provide a serial communication port.

S7-1200 Standard CPU Models

CPU	PROFINET Ports	On-board I/O		Expansion	
		Digital	Analog	Signal Modules	Comm. Modules
 1211C	1	6 in/4 out	2 in	0	3
 1212C	1	8 in/6 out	2 in	2	3
 1214C	1	14 in/10 out	2 in	8	3
 1215C	2	14 in/10 out	2 in/2 out	8	3
 1217C	2	14 in/10 out	2 in/2 out	8	3

Key features of S7-1200 standard CPUs are shown in the accompanying graphic. All models are 100 millimeters (mm) high by 75 mm deep, but vary in width from 90 mm for CPU 1211C and CPU 1212C to 150 mm for CPU 1217C.

Each CPU model, except CPU 1217C, is available in the following three power configurations: DC/DC/DC, DC/DC/RLY, and AC/DC/RLY. CPU 1217C is available in the DC/DC/DC power configuration.

The first two letters designate the type of CPU power required. DC indicates that 24 VDC is required, and AC indicates that 120 or 230 VAC is required.

The second two letters indicate the type on-board digital input channels provided. In all cases, 24 VDC input channels are provided.

The remaining letters indicate the type of on-board digital output channels provided. DC indicates that 24 VDC output channels are provided, and RLY indicates that relay output channels are provided. Relay outputs can be used with AC or DC power.

S7-1200 CPU Memory



For S7-1200 PLCs a SIMATIC memory card can be used instead of internal load memory.

Memory Type	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Work Memory	30 kbytes	50 kbytes	75 kbytes	100 kbytes	125 kbytes
Load Memory	1 Mbyte	2 Mbytes	4 Mbytes	4 Mbytes	4 Mbytes
Retentive Memory	10 kbytes	10 kbytes	10 kbytes	10 kbytes	10 kbytes
Input Image	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes
Output Image	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes
Bit Memory	4096 bytes	4096 bytes	8192 bytes	8192 bytes	8192 bytes

Memory sizes for S7-1200 Standard CPUs

S7-1200 CPUs have the following user memory types.

- Load memory is non-volatile memory for the user program, program data, and configuration information. Load memory is located either in the CPU or on a SIMATIC memory card. SIMATIC memory cards are available in a range of sizes, including sizes that are greater than the internal load memory for any CPU. A SIMATIC memory card can also be used to transfer load memory content to another CPU.
- Work memory is volatile storage for some elements of the user program. The CPU copies these elements from load memory into work memory during program execution. This is volatile memory, so the information is lost when CPU power is lost, but is restored by the CPU when power resumes.
- Retentive memory is non-volatile storage for a limited quantity of work memory content.
- Much of the data used by the user program is stored in data blocks in the user program. However, CPUs also have global memory areas that are accessible by the user program. These include bit memory (M) and image tables for inputs (I) and outputs (Q).

CPU Modes and CPU Status LEDs



The current mode for the CPU is indicated by the RUN/STOP CPU status LED. S7-1200 CPUs have the following three operating modes.

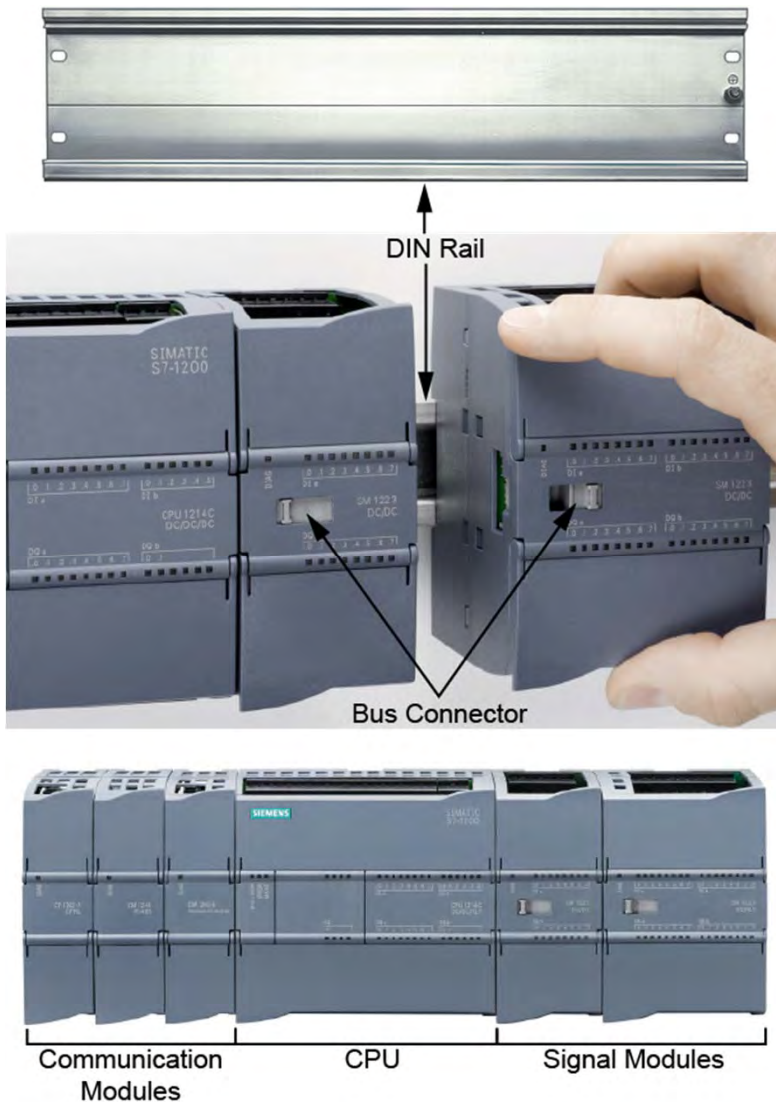
- In the STOP mode, the CPU is not executing the user program and a new program can be downloaded to the CPU.
- In the STARTUP mode, the CPU executes any startup logic, if present.
- In the RUN mode, the CPU executes the PLC scan repeatedly.

The CPU mode is determined by a software setting and whether or not a critical CPU error has occurred. In the event of a critical error, the CPU goes into the STOP mode.

Adjacent to the RUN/STOP status LED are the ERROR and MAINT status LEDs .

LED	Condition	Indication
RUN/STOP	STOP Mode	Constant Yellow
	STARTUP Mode	Alternatively Flashing Green/Yellow
	RUN Mode	Constant Green
ERROR	CPU, Memory Card, or Configuration Error	Flashing Red
	Defective Hardware	Constant Red
MAINT	When a Memory Card is Inserted	Flashing Yellow
	Maintenance Requested	Constant Yellow

Module Installation



S7-1200 hardware has a compact design to save control panel space. In addition, for mounting flexibility, S7-1200 modules can be mounted in either a horizontal or vertical position.

All SIMATIC S7-1200 modules have built-in clips that allow for easy and convenient mounting on a standard 35 mm DIN rail. These clips can also be snapped into an extended position to provide mounting holes for panel mounting.

When additional I/O is required beyond what is available on-board, all CPUs, except the CPU 1211C, allow signal modules to be mounted to the right of the CPU. CPU 1212C permits the addition of two signal modules, and the remaining CPUs permit the addition of eight signal modules. Once the signal module is in place, the bus connector on the front of the module is moved to the left to allow the module to communicate with the CPU.

All S7-1200 CPUs have at least one built-in PROFINET port, CPU 1215C and CPU 1217C have two PROFINET ports. For other types of communication, all CPUs can be equipped with up to three communication modules mounted to the left of the CPU.

Signal Boards



Digital input devices provide an on or off signal to a PLC, and digital output devices turn on or off in response to a PLC signal.

Analog input devices provide a variable current or voltage, such as 4 to 20 milliamps or 0 to 10 volts, representing a condition in a machine or process, and analog output devices require a similarly variable signal from the PLC. In addition, analog input devices such as thermocouples and resistance temperature detectors (RTDs) provide variable signals that require special processing.

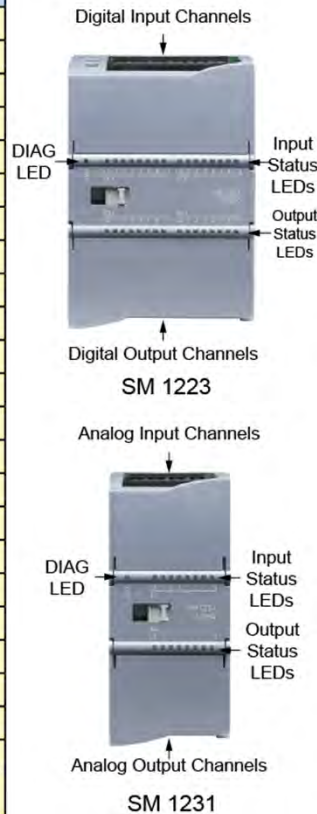
A digital I/O device connects to an S7-1200 PLC through to a digital I/O channel on a CPU, digital signal board, or digital signal module. Similarly, an analog I/O device connects to an S7-1200 PLC through an analog I/O channel on a CPU, analog signal board, or analog signal module.

A signal board can be added to the CPU to increase the CPUs I/O channels without increasing the size of the CPU. As shown in the accompanying graphic, various signal boards are available to add digital or analog I/O channels.

Signal Board Designation	Digital or Analog	Channels	Channel Type
SB 1221	Digital	4 Inputs	24 VDC (Source), 200 kHz
SB 1221	Digital	4 Inputs	5 VDC (Source), 200 kHz
SM 1222	Digital	4 Outputs	24 VDC (Sink/Source), 200 kHz
SM 1222	Digital	4 Outputs	5 VDC (Sink/Source), 200 kHz
SB 1223	Digital	2 Inputs	24 VDC (Source), 200 kHz
		2 Outputs	24 VDC (Sink/Source), 200 kHz
SB 1223	Digital	2 Inputs	5 VDC (Source), 200 kHz
		2 Outputs	5 VDC (Sink/Source), 200 kHz
SB 1223	Digital	2 Inputs	24 VDC (Sink)
		2 Outputs	24 VDC (Source)
SB 1231	Analog	1 Input	Selectable Ranges
SB 1231	Analog	1 Input	RTD
SB 1231	Analog	1 Input	24 VDC (Source)
SB 1232	Analog	1 Output	Selectable Ranges

Signal Modules

Signal Module Designation	Digital or Analog	Channels	Channel Type
SM 1221	Digital	8 Inputs	24 VDC (Sink/Source)
SM 1221	Digital	16 Inputs	24 VDC (Sink/Source)
SM 1222	Digital	8 Outputs	24 VDC (Source)
SM 1222	Digital	16 Outputs	24 VDC (Source)
SM 1222	Digital	8 Outputs	Relay
SM 1222	Digital	16 Outputs	Relay
SM 1223	Digital	8 Inputs	24 VDC (Sink/Source)
		8 Outputs	24 VDC (Source)
		16 Inputs	24 VDC (Sink/Source)
SM 1223	Digital	16 Outputs	24 VDC Outputs (Source)
		8 Inputs	120/230 VAC (Sink/Source)
SM 1223	Digital	8 Outputs	Relay
SM 1231	Analog	4 Inputs	Selectable Ranges
SM 1231	Analog	8 Inputs	Selectable Ranges
SM 1231	Analog	4 Inputs	Thermocouple
SM 1231	Analog	8 Inputs	Thermocouple
SM 1231	Analog	4 Inputs	RTD
SM 1231	Analog	8 Inputs	RTD
SM 1232	Analog	2 Outputs	Selectable Ranges
SM 1232	Analog	4 Outputs	Selectable Ranges
SM 1234	Analog	4 Inputs	Selectable Ranges
		2 Outputs	Selectable Ranges
SM 1238	Analog	Input	Records various electrical values for voltages up to 480 VAC



The accompanying graphic shows a combination digital input/output signal module and a combination analog input/output signal module and lists the various standard signal module types available.

S7-1200 PLC signal modules have removable terminal strips that speed module replacement by allowing field wiring to remain in place when a module is removed.

Signal modules have channel status LEDs with numbers that match the corresponding input or output terminals. The indicators are numbered in groups with the first input or output per group associated with terminal “.0” for that group.

A digital signal module I/O channel LED is green and turns on or off to indicate the on or off state of the individual input or output. An analog signal board or module channel LED is green when the channel has been configured and is active. An analog channel LED is red when there is an error condition.

In addition, all signal modules have a diagnostic (DIAG) LED that is green when the module is operational and red when the module is defective or non-operational.

Energy Meter Module



The accurate recording of energy consumption is essential for enterprise energy management. The SM 1238 Energy Meter module provides cost-efficient, entry-level, machine-oriented energy management. At just 45 mm wide, this module is extremely compact and saves space in the control cabinet.

The Energy Meter module enables energy management with up to 200 measured energy values. It records electrical values in a single or three-phase network and measures voltages up to 480 VAC. External current transformers (1 A or 5 A) with a conversion factor of up to 10,000 are used to sense the current.

The measured energy data can be processed directly in the CPU and visualized on an HMI such as a SIMATIC HMI Basic or Comfort Panel.

I/O Link Master Module

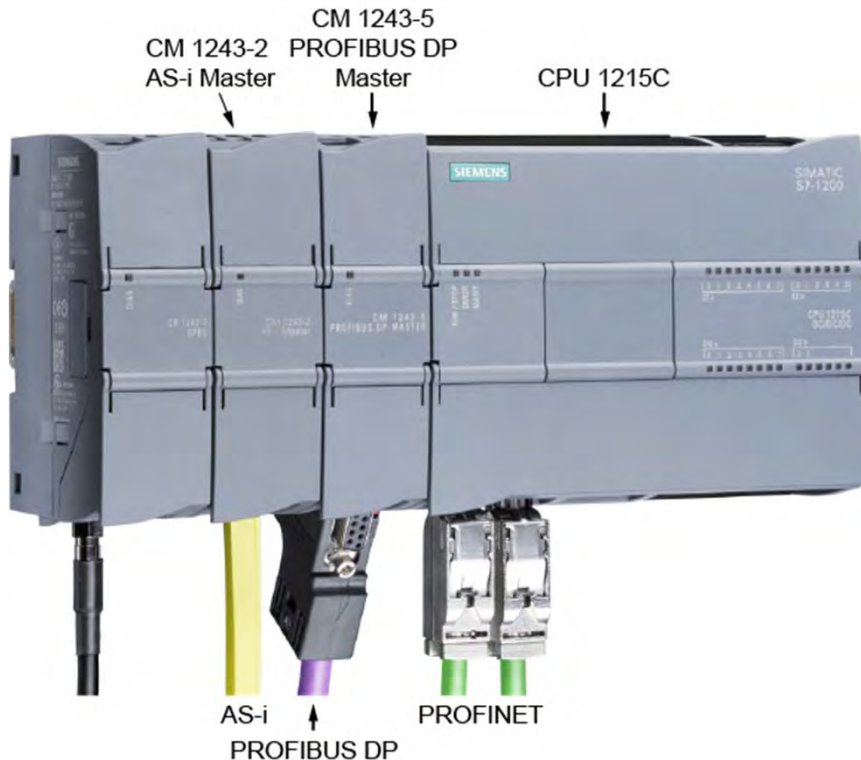


SM 1278 4 x IO-Link Master

IO-Link is an open communication standard developed by the IO-Link research group of the PROFIBUS & PROFINET International organization in response to the need for a simple, low-cost way to allow actuator and sensor communication in a concentrated area, such as for a control panel or individual machine. IO-Link is a point-to-point system, not a fieldbus. It uses an unshielded, three-wire cable, a 24 VDC power supply, and an IO-Link master.

The SM 1278 4 x IO-Link Master is a 4-port module that functions as both a signal module and a communication module. Each port can operate in the IO-Link mode, for connection to an IO-Link device, as a 24 VDC digital input, or as a 24 VDC digital output.

Communication Board and Modules



All S7-1200 CPU models have one or two built-in PROFINET ports and can also have one CB 1241 communication board installed in the CPU to provide an RS-485 serial communication port for point-to-point communication. S7-1200 PLCs can have up to three communication modules mounted to the left of the CPU.

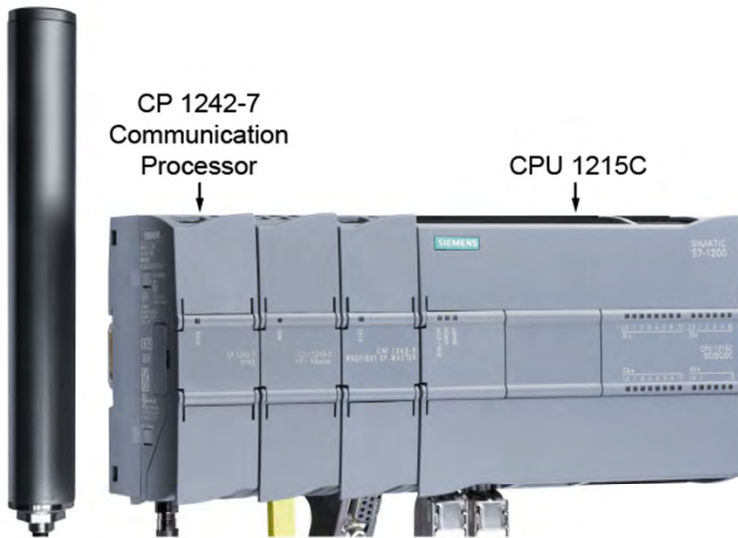
S7-1200 PLC communication modules are available for PROFIBUS DP, AS-Interface (AS-i), or other fieldbuses and for point-to-point serial communication or wireless communication.

A CM1243-5 PROFIBUS DP Master module can be installed to allow an S7-1200 PLC to function as a master device, or a CM 1242-5 PROFUS DP Slave module can be installed, to allow an S7-1200 PLC to function as a slave device.

AS-Interface (AS-i) is a fieldbus that simplifies the interconnection of actuators and sensors. A CM 1243-2 AS-i Master module can be installed to allow an S7-1200 PLC to function as an AS-i master.

TeleControl

GSM/GPRS Antenna



CP 1243-1

TeleControl refers to the connection of control stations spread out over a wide area to one or more central control systems.

An S7-1200 PLC functioning as a remote station, also called a remote terminal unit (RTU), requires a communication processor (CP) to securely communicate with the central control system. The CP incorporates a combination of security functions including a firewall and protocols for data encryption.

GPRS CP 1242-7 communication processor supports TeleControl applications using general packet radio service (GPRS).

CP 1243-1 communication processor supports TeleControl applications via an Ethernet network or the Internet.

CP 1243-1 DNP3 and CP 1243-1 IEC communication processors support TeleControl with standardized remote control protocols.

Scalable and Flexible Design

Communication Modules

Up to 3 communication modules can be added to any S7-1200 CPU.
 Up to 62 ASi devices can be connected to an AS-i Master module.
 Up to 16 PROFIBUS DP slaves can be connected to a PROFIBUS DP Master module.
 TeleControl can be used to monitor and control remote S7-1200 PLCs.

Signal Modules

Up to 8 signal modules can be added to the largest S7-1200 CPUs.
 A broad range of digital and analog signal modules are available.
 Up to 4 IO-Link devices can be connected to an IO-Link Master Module.
 Energy Meter module provides cost-efficient energy management.



Integrated PROFINET Interface

All S7-1200 CPUs have at least 1 PROFINET port.
 CPU 1215C and CPU 1217C have 2 PROFINET Ports.

Memory

All S7-1200 CPUs have non-volatile load memory for the user program, data, and configuration information.
 CPUs with up to 4 MB of load memory are available.
 Optionally, SIMATIC memory cards are available in a range of sizes.

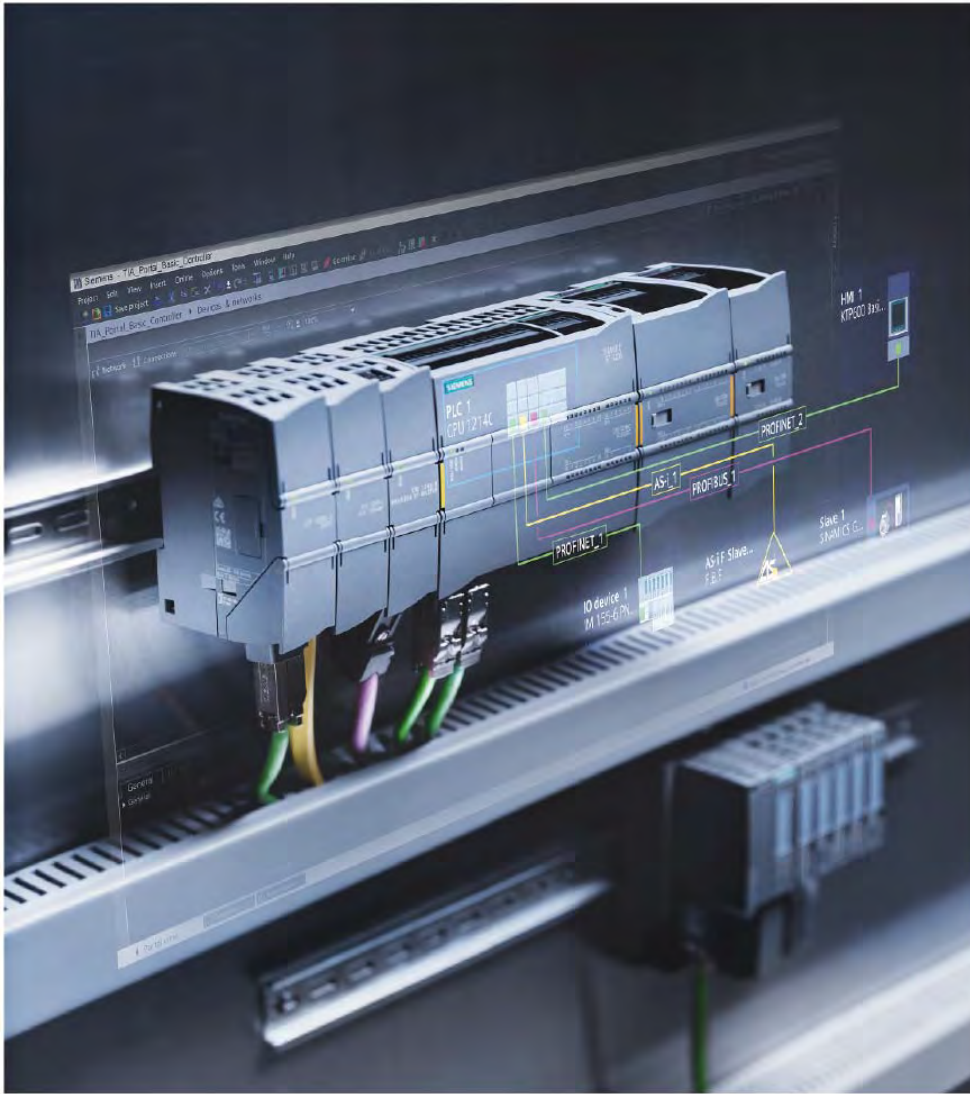
On-board I/O

All S7-1200 CPUs have integrated I/O channels.
 CPU 1217C has 15 digital inputs, 10 digital outputs,
 2 analog inputs, and 2 analog outputs.

Signal Board

One signal board can be added to an S7-1200 CPU without affecting the physical size of the CPU.

Chapter 2 - S7-1200 PLCs



This chapter covers the following topics:

- S7-1200 PLC Overview
- **S7-1200 with Safety Integrated**

Functional Safety



Safety is a relative term because it is impossible to eliminate all risk from a system or operation. From a practical perspective, this means that a prime objective of system design is to eliminate unacceptable risks to people and property.

An important aspect of risk elimination is achieving functional safety, which eliminates unacceptable risk by ensuring that all parts of a machine or system function in a safe manner. Failsafe is another term often used in this context. Even the best designed equipment has some potential for failure, but a failsafe device or system is one that will not cause injury or damage when a failure occurs.

Achieving functional safety requires that the safety-related parts of the protection and control systems function correctly. In addition, the systems must behave in such a way that either the plant remains in a safe state, or it is put into a safe state if a fault occurs.

To accommodate the wide range of functional safety applications, Siemens offers the full range of safety-related products from basic components to the most innovative solutions.

European Machinery Directive



Siemens Safety Evaluation Tool

Although safety standards vary throughout the world, you may need to know about the European Machinery Directive even if you do not work in a European manufacturing plant. Compliance with this directive is required for machinery operated in many European countries regardless of where the machinery was designed and manufactured. In addition, some international companies with factories in Europe design machinery intended for use outside of Europe consistent with the Machinery Directive to avoid the cost and complexity of multiple machine designs.

Conforming to the Machinery Directive requires machine manufacturers to perform a risk assessment of a machine design. This risk assessment must identify and evaluate all hazards associated with a machine. In addition, the machine manufacturer must take steps to reduce the risks to an acceptable residual level and must document proof of compliance.

To simplify the process of risk assessment, Siemens offers a free, online Safety Evaluation Tool that guides the user step-by-step through an evaluation of the safety functions of a machine and provides a report that can be integrated in the machine's documentation as proof of functional safety.

Safety Integrated for Factory Automation






While localized safety applications are common for individual machines or factories with limited automation, many factories require safety systems that integrate with automation solutions. Siemens Safety Integrated for Factory Automation solutions include automation systems, drive technology, operator control and monitoring systems, safe industrial controls, and the software to simplify component and system engineering.

Safety Integrated products and systems permit the seamless integration of safety technology with standard automation. This integration of standard and safety technology offers considerable benefits for enhanced competitiveness for both machine manufacturers and the end users of their machines.

Machine manufacturers benefit from reduced hardware and significantly simplified engineering that speeds the design of machines and systems. End users benefit from the increased safety and productivity provided by the use of safe machines and systems.

S7-1200 Failsafe CPUs

CPU	PROFINET Ports	On-board I/O		Expansion	
		Digital	Analog	Signal Modules	Comm. Modules
 1212FC	1	8 in/6 out	2 in	2	3
 1214FC	1	14 in/10 out	2 in	8	3
 1215FC	2	14 in/10 out	2 in/2 out	8	3

Memory Type	CPU 1212FC	CPU 1214FC	CPU 1215C
Work Memory	50 kbytes	75 kbytes	100 kbytes
Load Memory	2 Mbytes	4 Mbytes	4 Mbytes
Retentive Memory	10 kbytes	10 kbytes	10 kbytes
Input Image	1024 bytes	1024 bytes	1024 bytes
Output Image	1024 bytes	1024 bytes	1024 bytes
Bit Memory	4096 bytes	8192 bytes	8192 bytes

S7-1200 failsafe CPU 1212FC, CPU 1214FC, and CPU 1215FC are available in the following power configurations: DC/DC/DC and DC/DC/RLY. This means that the CPU and on-board input channel power must be 24 VDC for all models. The last two letters designate the on-board output channel power requirements. DC indicates that 24 VDC is required and RLY indicates that relay output channels are provided. Relay outputs can be used with AC or DC power.

An S7-1200 failsafe system requires a failsafe CPU and failsafe signal modules. The CPU's on-board I/O channels are not failsafe, but can be used for other control functions.

S7-1200 failsafe CPUs can be used for both standard and safety-related applications. S7-1200 standard signal modules, communication modules, and signal boards can be used in the same system to perform functions that do not require a rated Safety Integrity Level. This reduces the wiring required in comparison to conventional solutions with separate safety systems.

S7-1200 Failsafe Signal Modules



SM 1226 F-DI 16 x 24 VDC



SM 1226 F-DQ 4 x 24 VDC



SM 1226 F-DQ 2 x Relay

The accompanying graphic shows the available S7-1200 failsafe signal modules.

SM 1226 F-DI 16 x 24 VDC has 16 input channels for connection to 24 VDC sensors or switches. The module has two sensor supplies that can each power eight external inputs.

SM 1226 F-DQ 4 x 24 has four output channels that are each rated for connection to 24 VDC at up to a 2.0 A.

SM 1226 F-DQ 2 x Relay has two output channels. Each channel has two contacts in series controlled by independent relay coils.

S7-1200 failsafe signal modules have two processors. The processors monitor each other and verify that they are executing the same code at the same time, automatically test the I/O circuits, and set the module to a safe state in the event of a fault.

Communication between the failsafe CPU and the failsafe signal modules is verified using the PROFIsafe protocol.

SIMATIC S7-1200 with Safety Integrated

Safety Integrated Products and Systems
 Siemens Safety Integrated products and systems permit the the seamless integration of safety technology with standard automation. Machine manufacturers benefit from reduced hardware and significantly simplified engineering that speeds the design of machines and systems. End users benefit from the increased safety and productivity provided by the use of safe machines and systems.

SIMATIC S7-1200 with Safety Integrated
 Standard and failsafe automation with just one controller with integrated PROFINET communication. This reduces the wiring required in comparison to conventional solutions with separate safety systems.



S7-1200 Failsafe CPUs
 The failsafe CPU executes your safety program along with standard applications programs. Communication between the failsafe CPU and failsafe signal modules is verified using the PROFIsafe protocol.

S7-1200 Failsafe Signal Modules
 S7-1200 failsafe signal modules have two processors. The processors monitor each other and verify that they are executing the same code at the same time, automatically test the I/O circuits, and set the module to a safe state in the event of a fault.

Chapter 3 - S7-1200 PLC Programming



This chapter covers the following topics:

- **Programming Concepts**
- LAD Programming Basics
- LAD Timers and Counters

TIA Portal



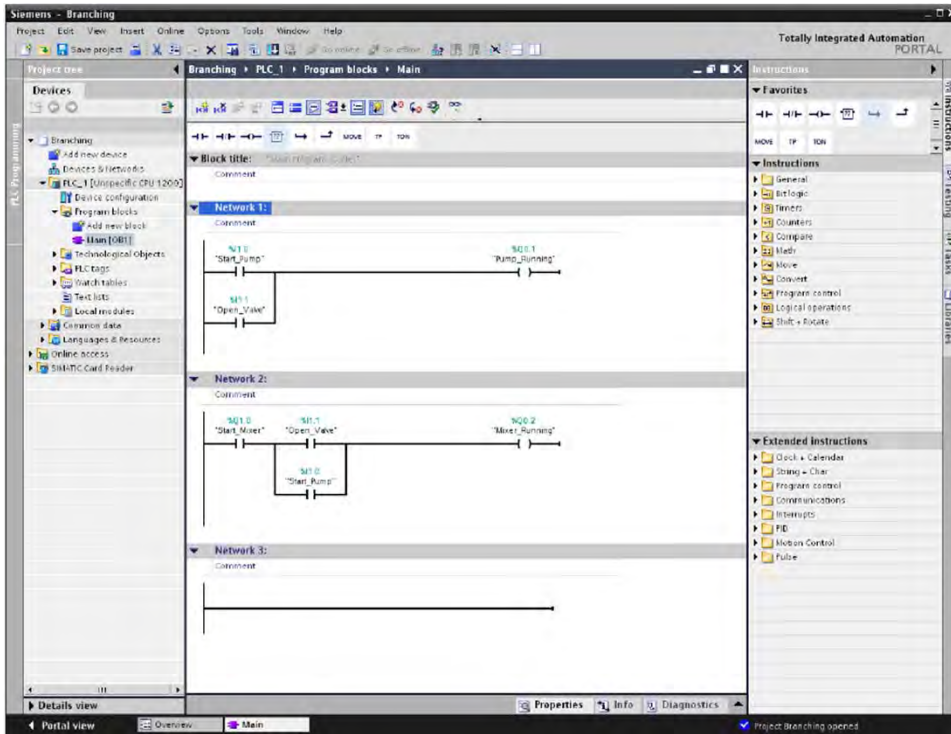
Siemens Totally Integrated Automation Portal (TIA Portal) is engineering software that is used in all phases of the design, operation, and maintenance of systems that can include Siemens PLCs, HMIs, PCs, electronic drives, and related devices.

TIA Portal combines the software editors needed for these various tasks in one engineering tool with a common layout and navigation design. This integrated, intuitive approach speeds the learning process and also allows experienced users to function more efficiently.

At the start of a project, the user can choose between the portal view, which guides the user through each engineering step, and the project view, which offers fast access to all the relevant tools. With one click a user can toggle between views.

The portal view is a good place to start for a new user or anyone who wants to logically proceed with the development of a new project or continue with the development of an existing project. The available tasks are clearly identified, such as “Create new project,” “Configure a device,” or “Write PLC program.”

STEP 7 Basic (TIA Portal)



STEP 7 Basic (TIA Portal) is used for the configuration and programming of S7-1200 PLCs and related devices. This can also be done using STEP 7 Professional (TIA Portal); however, STEP 7 Basic (TIA Portal) is a more economical approach when other PLC models are not being used. Both STEP 7 Basic (TIA Portal) and STEP 7 Professional (TIA Portal) simplify project design and allow for online diagnostics of operational systems.

STEP 7 Basic supports the following IEC 61131 PLC programming languages:

- Ladder diagram (LAD), shown in the accompanying figure, is a graphical programming language originally derived from electromechanical control circuit diagrams, but which has been expanded to include much more advanced capabilities.
- Function block diagram (FBD), is a graphical programming language similar in capabilities to LAD.
- Structured control language (SCL) is a high-level, text-based PLC programming language. SCL is particularly suitable for the high-speed programming of complex algorithms and arithmetic functions.

Security Integrated



Security functionalities are integrated in TIA Portal and S7-1200 to provide know-how protection, copy protection, access protection, and manipulation protection.

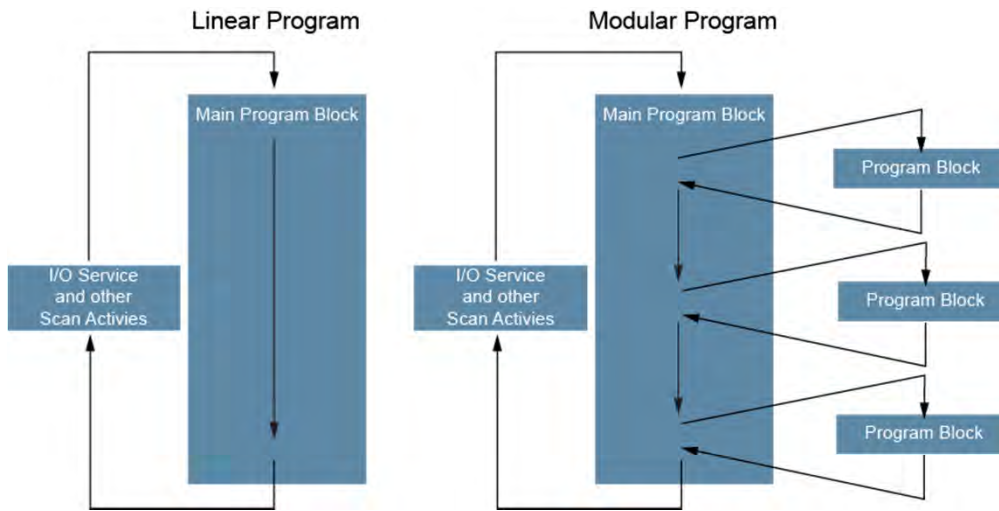
Know-how protection - Algorithms can be protected against unauthorized access and modifications. Machines are protected from unauthorized replication.

Copy protection - Program blocks on the SIMATIC Memory card are linked to the serial number of the memory card and will only run with the intended memory card.

Access protection - Access protection functionality offers security against unauthorized project changes. Permission levels can be used to grant separate rights to different user groups.

Manipulation protection - The system helps to protect the data being transmitted to the controller from unauthorized manipulation and detects manipulated firmware updates. The system provides protection from network attacks such as infiltration of fake/recorded network communication.

Modular Programming

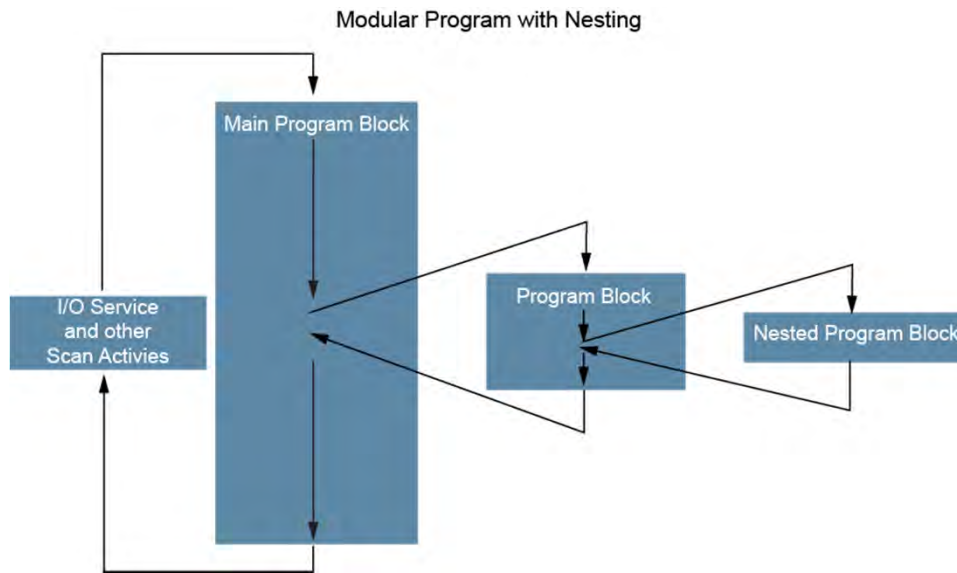


A PLC program can be organized as a linear program or a modular program. As shown on the left in the accompanying graphic, a linear program has all the instructions in one block and executes these instructions in sequence in each PLC scan. A linear program works well when the program is simple and does not need to vary from one PLC scan to another.

As shown on the right in the accompanying graphic, a modular program is composed of multiple program blocks. Some program blocks are executed in each PLC scan and other program blocks are executed under special circumstances, such as during startup, at specific times, or when an error or interrupt occurs. An interrupt is a special signal or condition that causes a PLC to immediately perform a specific task.

In programming terminology, when a program block initiates another program block, the initiating block is said call the other block. This causes the calling program block to temporarily halt its execution until the called program block completes its execution. Then the calling block resumes its execution.

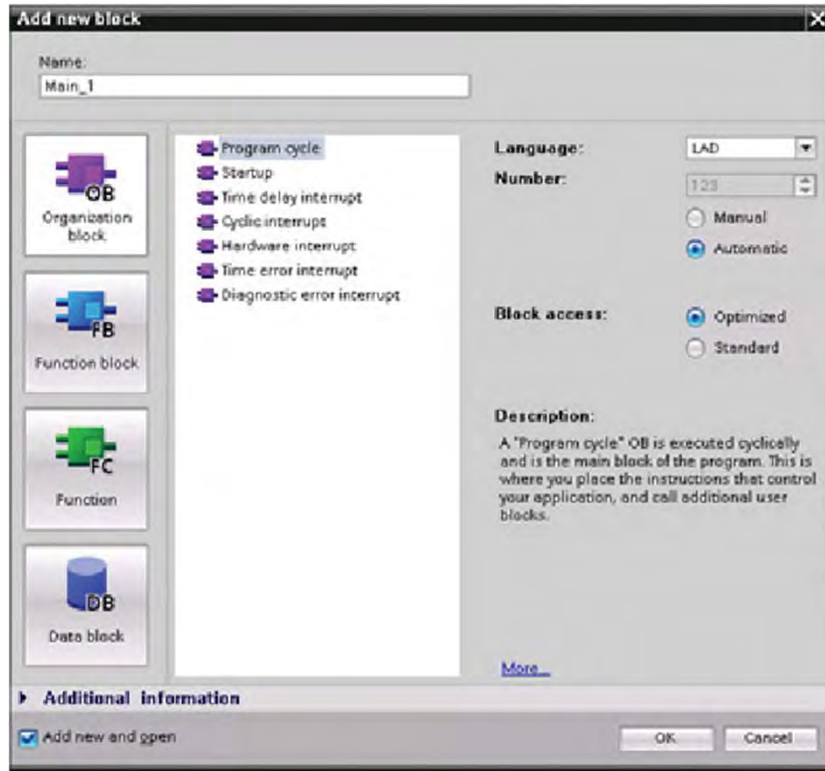
Nesting



As shown in the accompanying graphic, program blocks can also be nested. This means that one program block can call another program block which can call another program block. This provides additional flexibility in programming.

As compared to a linear program, a modular program with program blocks that perform functional tasks is easier to understand, develop, and debug. In addition, a modular program often performs better because it can vary the tasks from one PLC scan to another based on the needs of the application.

Program Blocks

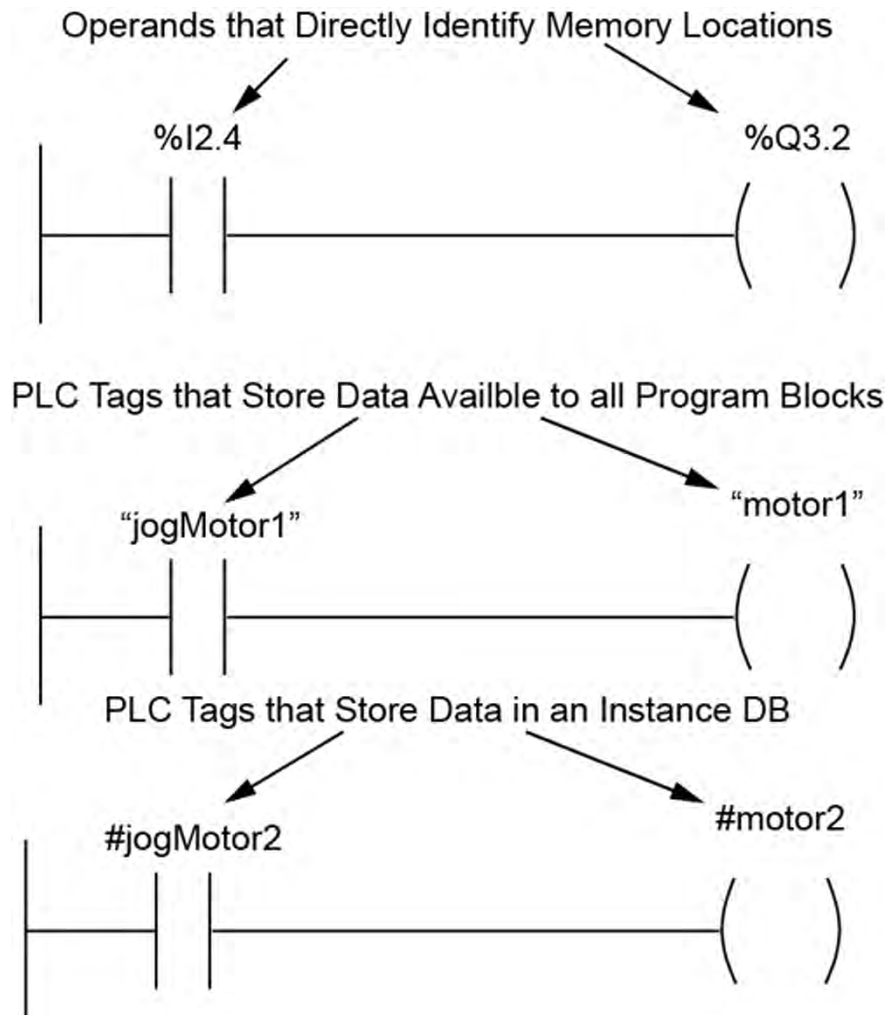


An S7 PLC program can include data blocks (DBs) and three types of program blocks: organization blocks (OBs), function blocks (FBs), and functions (FCs). Program blocks are blocks that include instructions.

Organization blocks (OBs) define the structure of the program. Every program must have at least one OB. If it has only one OB, that block is identified as OB 1. OB 1 is the main program block, and it controls the execution of the user program. OB 1 is also called a program cycle OB because it is executed each scan cycle while the CPU is in RUN mode. A modular program can be composed of multiple OBs, each with a unique number.

FCs and FBs are often referred to as subroutines. A subroutine is a portion of a program that is executed when certain conditions are met. FCs and FBs contain the program code that performs specific tasks. An FB uses an associated data block, called an instance DB, that stores data for that FB. Another type of data block, called a global data block, contains data that is available to any program block. An FC does not have an instance data block, and the output data values from the FC must be written to a memory address or to a global DB.

Assigning Memory Areas to Instructions



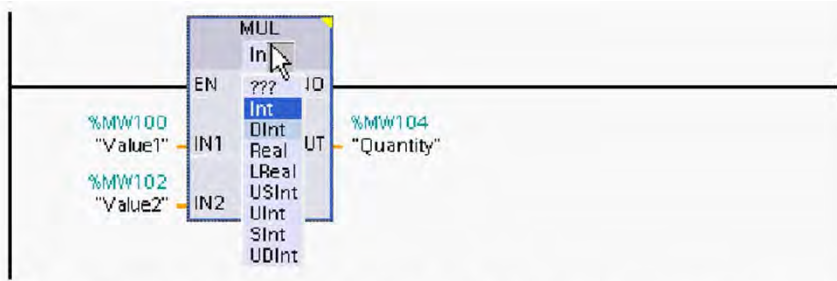
An S7 PLC CPU provides the following options for storing data associated with a user program.

- Global memory is memory that is accessible by all program blocks. Global memory includes a variety of specialized memory areas, such as inputs (I), outputs (Q), and bit memory (M).
- Data blocks (DBs) are used to store data. A global DB stores data that can be used by any program block, while an instance DB stores data for a specific function block (FB).

When you program instructions you must specify which data values the instruction will process. These values are referred to as operands. An operand can be a constant, an operand which directly identifies a memory location, or a tag.

Tags are symbolic alphanumeric operands that are useful because they aid understanding of the program. Tags are listed in a PLC tag table, which also identifies the associated memory location and other information. Tags appear in the program in quotation marks when the tag data is available to any program block and preceded by # when the data is associated with an instance DB.

Data Types



The data type for an instruction is selected during programming

Data type is a short way of saying a description of the data. There are a variety of data types, which are often represented by shorthand names such as DWord or SInt. These shorthand names are useful to understand when developing a program.

For example, the accompanying upper graphic shows a that the data type for an instruction is selected during programming. When making this selection, it is important to make sure that the data type is compatible with the operands associated with the instruction.

Name	Data type	Address	Retain	Visibl...	Acces...	Comment
P_Operation	Bool	%Q4.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Indicator light Operation ...
P_Fault	Bool	%Q5.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Indicator light Conveyor L...
S_OperationON	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Momentary contact Oper...
S_OperationOFF	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Momentary contact Oper...
S_Right	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Momentary contact Jog c...
S_Left	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Momentary contact Jog c...
M_aux_LB	Bool	%M16.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edge auxiliary memory bi...
M_Jog_Right	Bool	%M16.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory bit Jog conveyor ...
M_Jog_Left	Bool	%M16.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory bit Jog conveyor ...
M_Auto_Right	Bool	%M16.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory bit conveyor RIG...
M_Conv_Fault	Bool	%M17.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory bit Conveyor fault
M_aux_count	Bool	%M18.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edge auxiliary memory c...
M_Ack_HMI	Bool	%M31.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Conveyor fault acknowle...
MW_ACT	Int	%MW20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory word, ACTUAL qu...
MW_SET	Int	%MW22		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory word, SETPOINT ...
M_aux_Op_on	Bool	%M15.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edge auxiliary memory bi...
M_ACT=SET	Bool	%M18.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTUAL quantity of trans...
M_Timeout_right	Bool	%M150.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Lock-out time right
M_Timeout_left	Bool	%M150.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Lock-out time left
M_max_Fault	Bool	%M17.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	max. Number of Faults re...

PLC Tag Table

The data type is also among the values that must be selected when a tag is set up. The PLC tag table shows the data type and other important information for PLC tags.

STEP 7 Safety (TIA Portal)



SIMATIC STEP 7 Safety Basic (TIA Portal) or STEP 7 Safety Advanced (TIA Portal) are used for the configuration and programming of failsafe applications. STEP 7 Safety Basic (TIA Portal) is used with either STEP 7 Basic (TIA Portal) or STEP 7 Professional (TIA Portal). STEP 7 Safety Advanced (TIA Portal) requires STEP 7 Professional (TIA Portal).

All the configuration and programming tools needed for a safety program are integrated into the STEP 7 user interface and use a common project structure. You create a safety program using the program editor and program failsafe FBs and FCs in FBD or LAD programming languages using the instructions from the safety package.

Safety checks are automatically performed and additional failsafe blocks for error detection and error reaction are inserted when the safety program is compiled. This ensures that failures and errors are detected and appropriate reactions are triggered to keep the system in a safe state.

A standard program can function with a safety program in a failsafe CPU because the safety-related data is protected from being unintentionally affected by data of the standard user program.

Classroom Learning



Studies indicate that when students practice what they have learned in a classroom setting they retain 75% of the lesson, as compared with lecture-only settings where they retain just 20% of the lesson.

Our learning content is reviewed and approved by Siemens technical and operational experts to ensure compliance with the highest industry, health, safety, and environmental standards. Siemens simulator workstations provide a safe and risk-free platform for job training, project testing, design engineering, and troubleshooting.

We combine technology and industry experience to deliver highly effective, customized learning programs.

- Job targeted courses
- Hands-on learning and skill building
- System-level training approach
- Extensive schedule of classes
- Various media and course length options
- On-site and custom courses
- Multiple training center locations
- Packaged services and products

For additional information: www.usa.siemens.com/sitrain

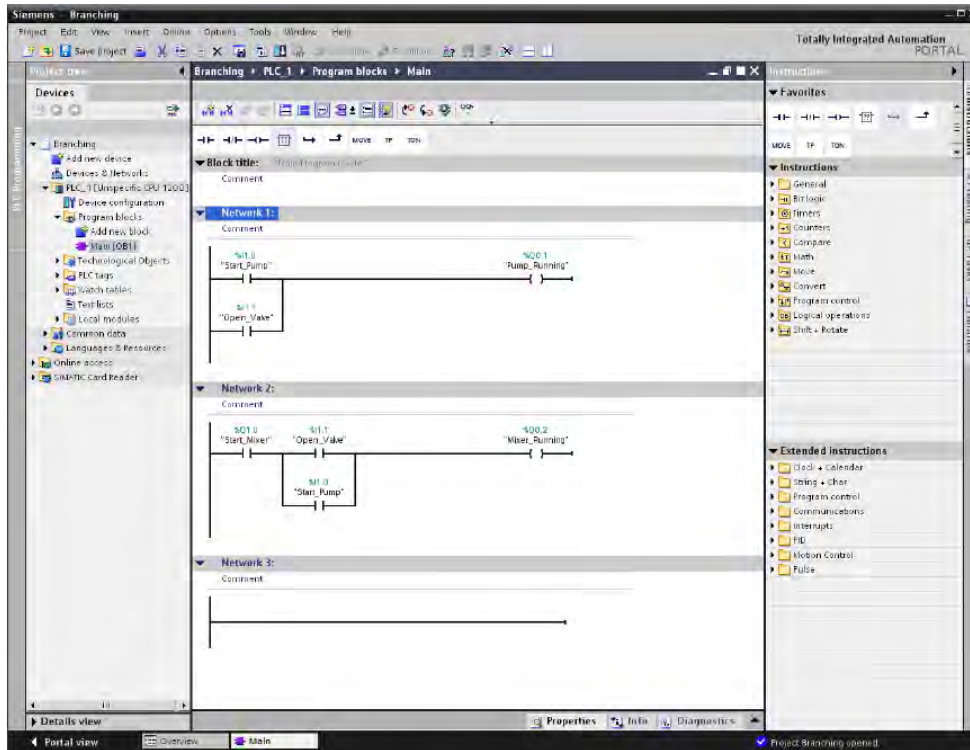
Chapter 3 - S7-1200 PLC Programming



This chapter covers the following topics:

- Programming Concepts
- **LAD Programming Basics**
- LAD Timers and Counters

Ladder Diagram (LAD) Programming



Ladder Diagram (LAD) programming is one of the programming languages defined in the IEC 61131 specification. LAD is a graphical programming language originally derived from the diagrams used in electromechanical control circuits, but which has been expanded to include much more advanced capabilities.

A LAD program or program block is composed of networks. For example, the accompanying graphic shows two completed networks and one network yet to be programmed. The networks in a program or program block are numbered consecutively beginning with network 1, the top network. Because the networks in a LAD program resemble the rungs of a ladder, a network is sometimes referred to a rung.

Each network is made up of instructions that are interconnected both graphically and in the order of execution of instructions. As a result, when a network is scanned, it is scanned from left to right beginning with the instructions closest to the power rail, which is the vertical line shown on the left in each network.

Instructions and Operands

Normally Open (NO)
Contact Instruction

%I3.2
"jogMotor2"



Output Coil
Instruction

%Q4.1
"motor2"

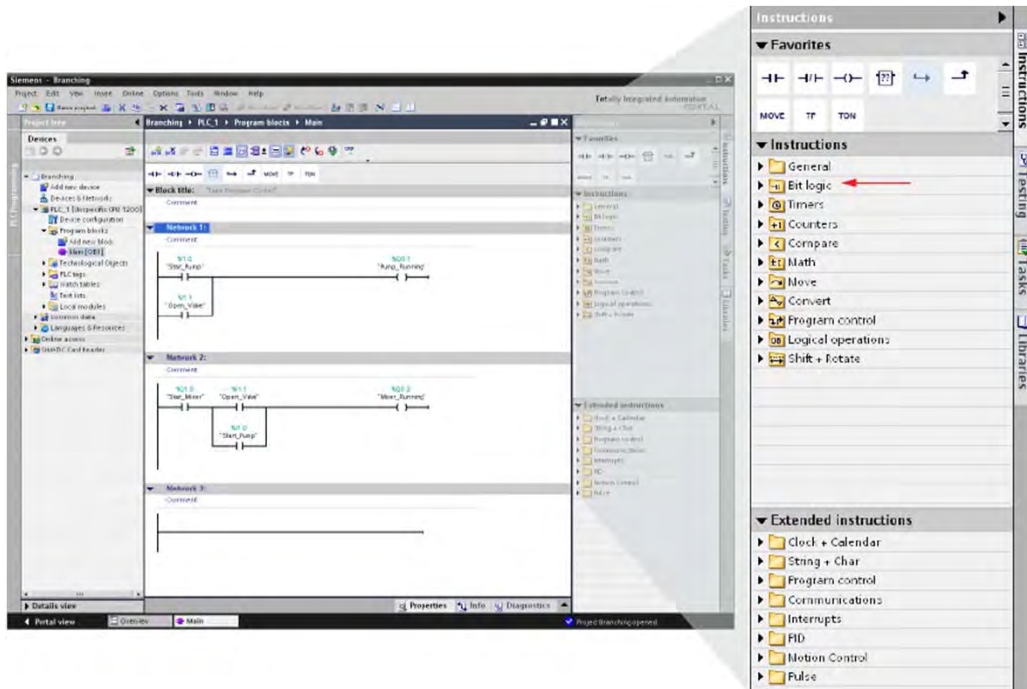
PLCs store information as binary numbers. One of the ways that PLCs use these binary bits is to represent the on or off condition of inputs and outputs.

Input (I) and output (Q) bits in the process image along with a type of memory called bit memory (M) are collectively called global memory because these memory areas are available throughout the user program.

Most instructions in a PLC program interact with one or more bits, bytes, words, double words in the CPU memory. The amount of memory used by an instruction depends on the type of instruction. The specific memory location or locations used are identified by one or more operands associated with an instruction.

For example, the instructions shown in the accompanying graphic each use a one-bit memory location. The operand for the normally open contact instruction indicates that this instruction is controlled by the I3.2 status bit, and the operand for the output coil instruction indicates that this instruction controls the Q4.1 status bit. Often times only tags such as "jogMotor2" or "motor2" are displayed, but tags also define memory locations and related information.

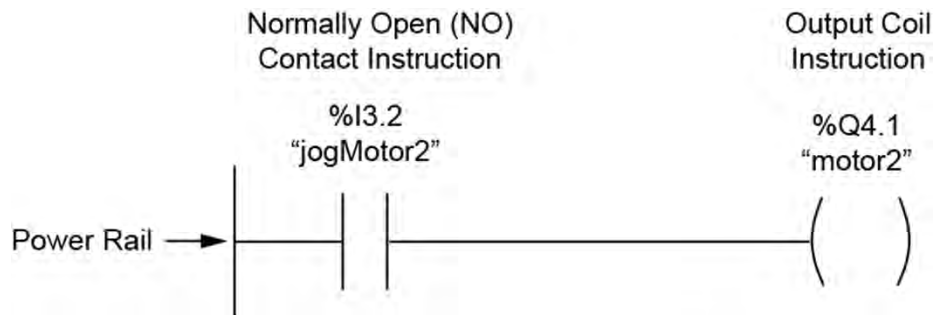
Bit Logic Instructions



Because the LAD programming language includes many instructions, STEP 7 groups instructions into categories. As shown in the accompanying graphic, there are a number of basic and extended instruction categories.

This lesson provides examples of some of the instructions in the bit logic category. Bit logic instructions are so called because they perform logical operations using single-bit memory locations.

Power Flow



Because LAD programming was originally based on control circuit diagrams, some electrical terminology is used in describing a LAD program. For example, even though a network in a LAD program is an arrangement of computer instructions, each network in a LAD program is said to control power flow from the power rail.

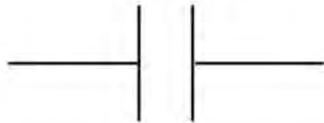
As the PLC scans the user program from top to bottom in a program block, each network is scanned from left to right. Scanning a network means that the PLC's CPU is examining the logical conditions described by the instructions to determine if power flows through the network. In programming terminology, this examination of logical conditions is called solving the logic and the solution of a configuration of instructions is called the result of logic operation (RLO).

Using the simple network shown in the accompany graphic as an example, when the normally open contact instruction permits power flow to the output coil instruction, output Q4.1 is on. When the normally open contact instruction does not permit power flow, output Q4.1 is off.

Contact and Coil Instructions

Normally Open (NO)
Contact Instruction

%I3.2
"startMotor3"



Normally Closed (NC)
Contact Instruction

%I4.5
"limitSwitch5"



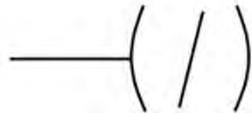
Output Coil Instruction

%Q5.1
"motor3"



Inverted Output Coil Instruction

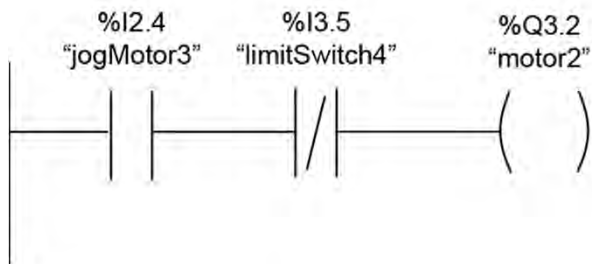
%Q3.5
"pilotLight3"



The accompanying graphic shows four of the most often used LAD instructions, the normally open and normally closed contact instructions and the output coil and inverted output coil instructions. It is important not to confuse the names of these instructions with any electrical device. Therefore, you should to memorize the following rules and use them when interpreting a bit logic network.

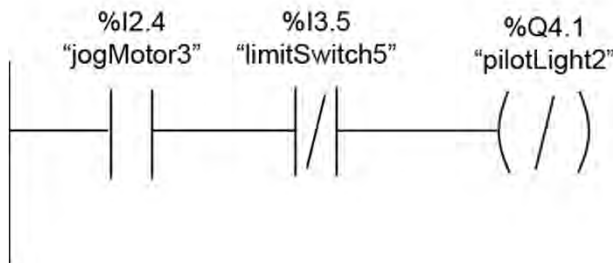
- **Normally open contact instruction:** The contact is closed when the status bit specified by the operand is 1. When the status bit specified by the operand is a 0, the contact is open and will not pass power flow.
- **Normally closed contact instruction:** The contact is closed when the status bit specified by the operand is 0. When the status bit specified by the operand is 1, the contact is open and will not pass power flow.
- **Output coil instruction:** This instruction sets the status bit specified by the operand to 1 when the instruction receives power flow. When this instruction does not receive power flow, the status bit specified by the operand is 0.
- **Inverted output coil instruction:** This instruction sets the status bit specified by the operand to 1 when there is no power flow to the instruction. When this instruction receives power flow, the status bit specified by the operand is 0.

Contact and Coil Instruction Examples



Truth Table

I2.4	I3.5	Q3.2
0	0	0
0	1	0
1	0	1
1	1	0



Truth Table

I2.4	I3.5	Q4.1
0	0	1
0	1	1
1	0	0
1	1	1

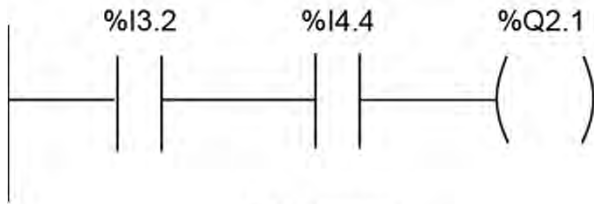
To understand how these rules work, consider the two networks shown in the accompanying graphic. For each network there is an associated truth table shown to the right of the network. A truth table is one way to show all the possible logical combinations associated with a network and the resulting logic condition for each combination.

First, consider the truth table conditions for the top network. Note that for both contacts to be closed at the same time I2.4 must be 1 and I3.5 must be 0. This is the only condition that will cause power flow to the output coil instruction, resulting in Q3.2 being set to 1.

Next, consider the truth table conditions for the bottom network. Because the final instruction is an inverted output coil instruction, anytime there is no power flow to this instruction, output Q4.1 will be set to 1. The only time that Q4.1 is 0 is when I2.4 is 1 and I3.5 is 0.

AND, OR, and Exclusive OR Functions

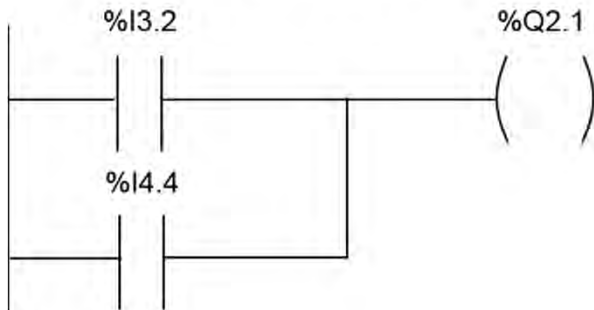
AND Function



Truth Table

I3.2	I4.4	Q2.1
0	0	0
0	1	0
1	0	0
1	1	1

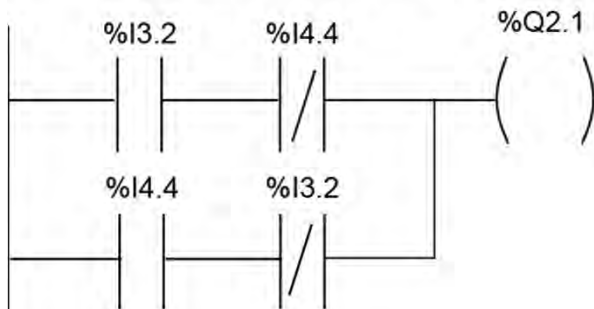
OR Function



Truth Table

I3.2	I4.4	Q2.1
0	0	0
0	1	1
1	0	1
1	1	1

Exclusive OR (XOR) Function



Truth Table

I3.2	I4.4	Q2.1
0	0	0
0	1	1
1	0	1
1	1	0

Three of the most common Boolean logic functions are the AND, OR, and Exclusive OR (XOR) functions. These functions can be easily represented in a LAD program.

For example, the top network in the accompanying graphic shows an AND function. For this network, as the associated truth table shows, the only condition that causes the output coil to turn on and the Q2.1 status bit to be set to 1 is when the I3.2 and I4.4 status bits are both 1. The AND function can be easily expanded by adding additional contacts in series.

The middle network shows an OR function. For this network, as the associated truth table shows, if the status bits for either or both of the normally open contact instructions are 1, the Q2.1 status bit is set to 1. The OR function can be easily expanded by adding more contacts in parallel.

Another common Boolean logic function is shown at the bottom. This is called the Exclusive OR (XOR) function. As the truth table for this function shows, Q2.1 is 1 if either I3.2 or I4.4 is 1. If I3.2 and I4.4 are in the same state, Q2.1 is 0.

Instructions that Set and Reset Bits



Additional networks can go in between the Set and Reset Coil networks.



The term set is often used to mean causing a 1 to be written to the operand, and the term reset is often used to mean causing a 0 to be written to the operand. There are a variety of LAD instructions that set and reset bits.

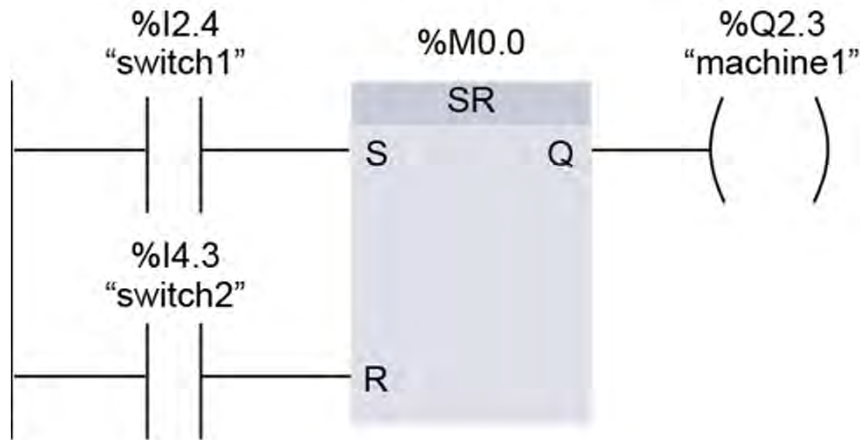
For example, the accompanying graphic shows networks that include the set and reset coil instructions. When there is power flow to the set coil instruction, the bit specified by the instruction's operand is set to a 1.

In this example, when I4.3 goes from 0 to 1, the corresponding normally open contact instruction closes and provides power flow to the set coil instruction. This causes output Q5.1 to be set to 1. Q5.1 remains a 1 even if I4.3 returns to 0.

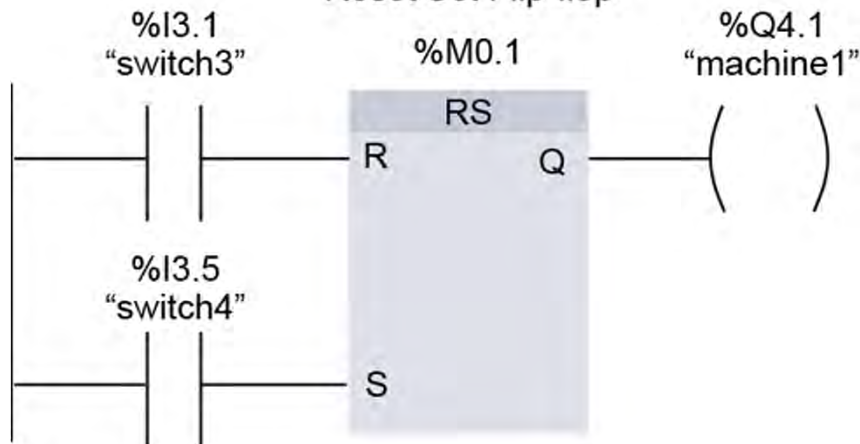
When there is power flow to the reset coil instruction, Q5.1 resets to 0. In this example, this happens when I4.5 goes to 1. There can be multiple networks in between the set and reset coil networks, and the set and reset coil networks can each have multiple contacts to establish the logical conditions for the set and reset operations.

Set Reset and Reset Set Flip-flops

Set Reset Flip-flop



Reset Set Flip-flop



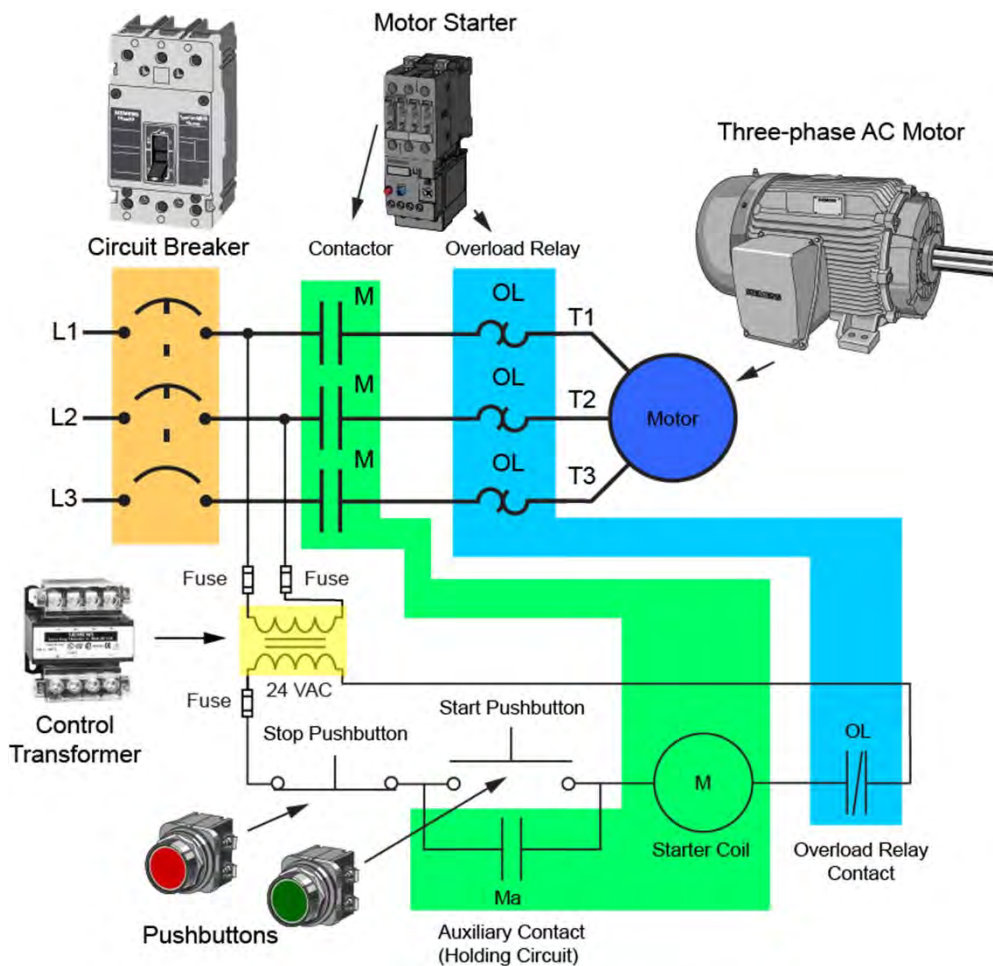
Two other instructions that set and reset bits are the set reset and the reset set flip-flops. Both networks perform a similar function. When there is power flow to the set (S) input, the output (Q) and the memory bit specified by the operand at the top of the instruction both go to 1. They remain a 1, even if there is no power flow at the set input, until there is power flow at the reset (R) input.

There is one important difference between these instructions. For the reset set flip-flop, the set input is dominant. For the set reset flip-flop, the reset input is dominant.

The significance is that, if there is power flow at both the set and reset inputs, the dominant input controls the result. In other words, with power flow at both the set and the reset inputs, the output and memory operand for the set reset flip-flop are both 0. For the reset set flip-flop, with power flow at both the set and the reset inputs, the output and memory operand are both 1.

In the examples shown, note that a bit memory (M) address is used as an operand. Using a bit memory address avoids the problem of picking an I or Q operand that could otherwise be associated with I/O devices.

Motor Start-Stop Circuit



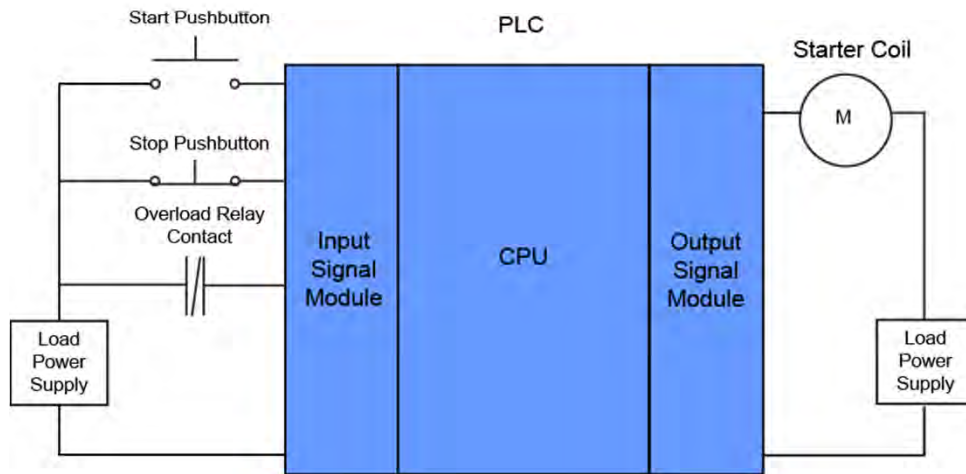
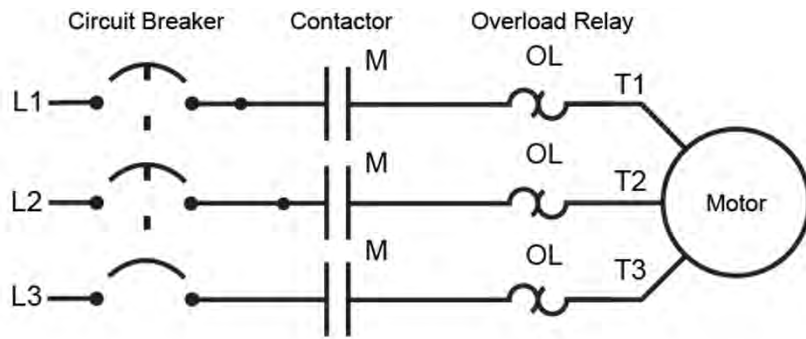
Three-phase AC motor start-stop control is a basic PLC application. Consider how AC motor control is often done using hard-wired circuits.

The upper portion of the diagram is called the power circuit, because power to the motor flows through a circuit breaker and a motor starter composed of a contactor and an overload relay.

Power to the control circuit is through a control transformer. The pushbuttons are wired to control the flow of current to the starter coil, which is part of the motor starter contactor. An auxiliary contact that is part of the contactor is also wired into this circuit as is a contact that is part of the motor starter overload relay.

Because the stop pushbutton and overload relay contacts in the control circuit are normally closed, when the start pushbutton is pressed, current flows through the starter coil, which closes both the motor starter's main contacts and the auxiliary contact in the control circuit, and the motor starts. Because the auxiliary contact is now closed, even when the start pushbutton is released, current continues to flow through the starter coil and the motor continues to run.

PLC Motor Control



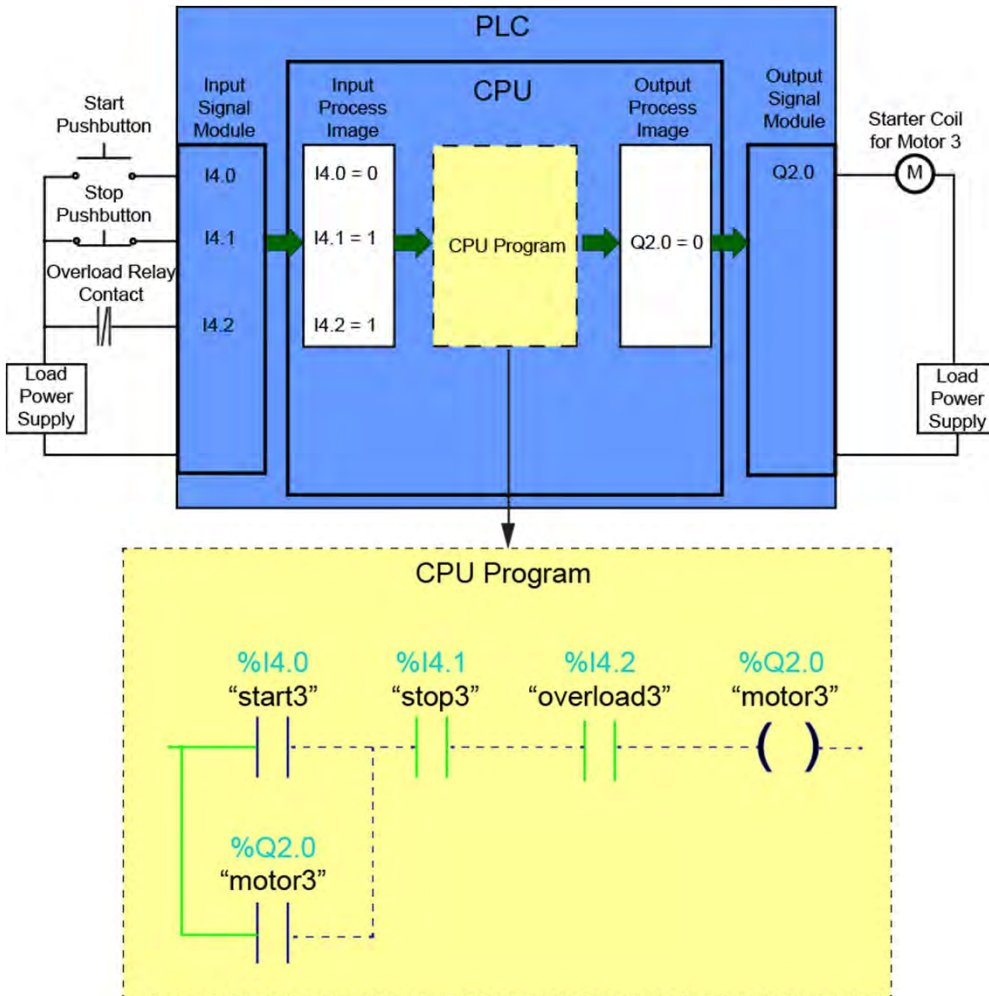
Motor start-stop control is a common application for a PLC. Note that the power circuits are not wired to the PLC and the hard-wired control circuit has been replaced by the PLC and the devices connected to it.

The input and output devices shown are not interconnected in the same way as for the hardwired circuit. It is only after the PLC has been configured and programmed that the inputs can be used to control the starter coil.

Consider how the motor start-stop control should function. When the start pushbutton is pressed, as long as the stop pushbutton is not also pressed and the normally closed overload relay contact is closed, the starter coil energizes, closing the contactor's contacts and starting the motor. The starter coil remains energized, allowing the motor to continue to run, when the start pushbutton is released.

When the stop pushbutton is pressed, the starter coil de-energizes, opening the contactor's contacts and removing power from the motor. The third input to the PLC is a normally closed contact that is part of the overload relay. When the overload relay senses an overload, this contact opens and the motor stops.

PLC Motor Control Program

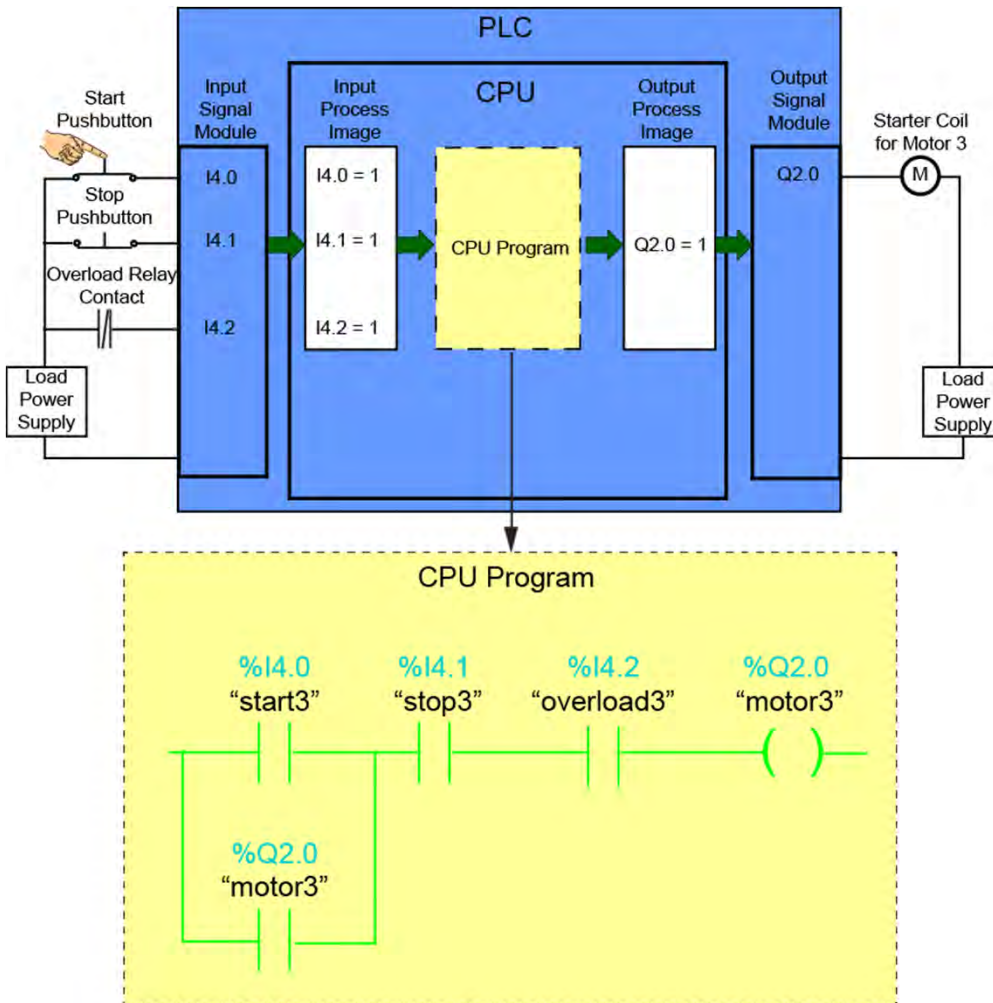


Consider how this could be programmed in a SIMATIC S7 PLC. For simplicity, this example has the three inputs wired to one input signal module that has been set up so that the input addresses used are I4.0, I4.1, and I4.2. These inputs could have been wired to separate input signal modules or to integrated digital input signal channels, if available. The output signal module has been set up so that the output address used is Q2.0.

The I/O address assignments determine which process image status bit is associated with each input and output. The operand for each instruction determines which process image status bit is associated with that instruction.

All the instructions needed are in one network. Recall that a normally open contact instruction is open when the associated process image status bit is 0 and closed when the associated process status bit is 1. This means that, for the current state of the inputs, the I4.1 and I4.2 normally open contacts are closed, and the other two normally open contacts are open. The output coil instruction controls the associated output process image status bit, causing it to be 0 when there is no power flow to the instruction, so output Q2.0 is off, and the motor is not running.

Starting the Motor



Note what happens when the start pushbutton is pressed. The I4.0 process image status bit is set to 1, causing the I4.0 normally open contact to close, and providing power flow to the output coil instruction. This sets the Q2.0 process image status bit to 1 and turns on output Q2.0. When output Q2.0 turns on, the starter coil for the motor energizes, and the motor starts.

On the next scan of the PLC, with the Q2.0 process image status bit set to 1, the Q2.0 normally open contact closes. With this contact closed, there will still be power flow to the output coil instruction even after the start pushbutton is released, and the motor will continue to run.

The motor will stop if the stop pushbutton is pressed or if a motor overload occurs, causing the normally closed overload relay contact to open.

How-to Video Library



This extensive library of short videos was created by our instructional experts to meet the real-world needs of industry, with all levels of experience in mind. By providing on-demand, how-to instruction in easy-to-understand bites, the How-to Video Library helps maintain the critical industrial and manufacturing knowledge and skills developed during instructor-led training courses. Videos are typically three-minutes long and conveniently available via any computer or mobile device with Internet access.

Learning begins once you've completed registration.

- Start your subscription at any time. Videos are available 24/7/365.
- Purchase one, three, six, or 12-month subscriptions by technology or in one complete bundle.
- Take advantage of our most-flexible option – ultimate access with a full, one-year subscription.

For additional information: www.usa.siemens.com/sitrain

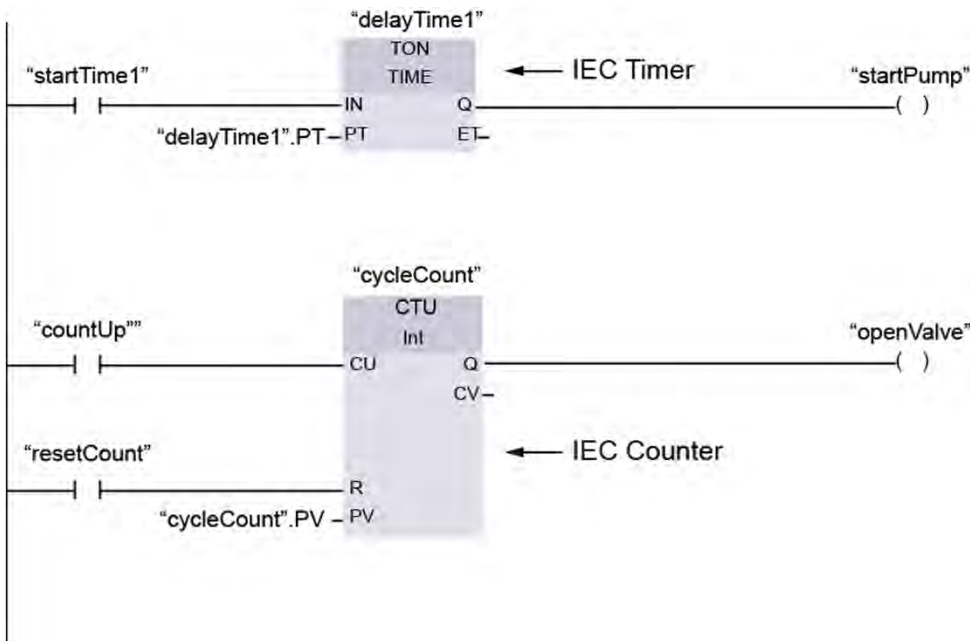
Chapter 3 - S7-1200 PLC Programming



This chapter covers the following topics:

- Programming Concepts
- LAD Programming Basics
- **LAD Timers and Counters**

Timers and Counters



Timers and counters are extremely useful PLC instructions. Timers allow PLCs to control and monitor time-dependent aspects of machines and processes. Counters allow PLCs to keep track of actions or events.

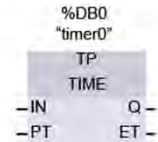
The IEC 61131 standard identifies PLC programming languages and standardizes the instructions available. Some Siemens S7 PLCs were already in use when the IEC 61131 specification was developed. This is noticeable when reviewing the SIMATIC timer and SIMATIC counter instructions available for S7-300 and S7-400 PLCs. While the tasks performed by SIMATIC timers and counters are similar to those performed by IEC timers and counters, there are noticeable differences. Keep in mind that S7-300 and S7-400 PLCs also allow IEC timers and IEC counters to be implemented through use of system function blocks.

Because S7-1200 and S7-1500 PLCs were developed well after the deployment of IEC 61131, these PLCs include IEC timers and counters as basic instructions. Therefore, to minimize the complexity of this lesson, the IEC timers and counters included are consistent with those used with S7-1200 PLCs.

IEC Timers

Box Instructions

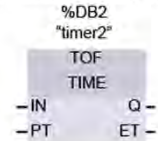
TP Timer (Pulse Timer)



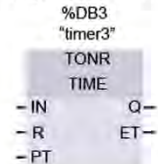
TON Timer (On-delay Timer)



TOF Timer (Off-delay Timer)

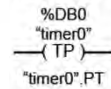


TONR Timer (On-delay Retentive Timer)

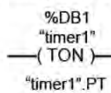


Coil Instructions

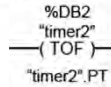
TP Timer Coil (Pulse Timer Coil)



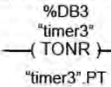
TON Timer Coil (On-delay Timer Coil)



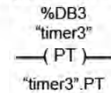
TOF Timer Coil (Off-delay Timer Coil)



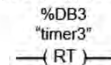
TONR Timer Coil (On-delay Retentive Timer Coil)



PT Coil (Preset Timer Coil)



RT Coil (Reset Timer Coil)



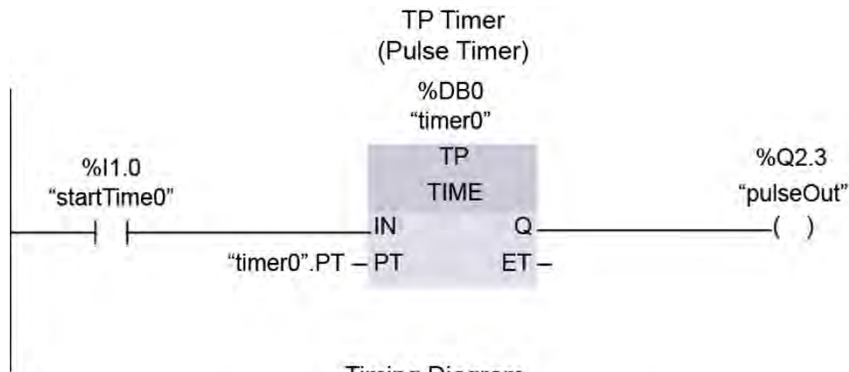
For LAD programs, S7-1200 PLCs support the IEC box and coil timer instructions shown in the accompanying graphic.

Each timer uses a data block (DB) to store timer data. For S7-1200 PLCs, the number of timers that you can use in program is limited only by CPU memory size.

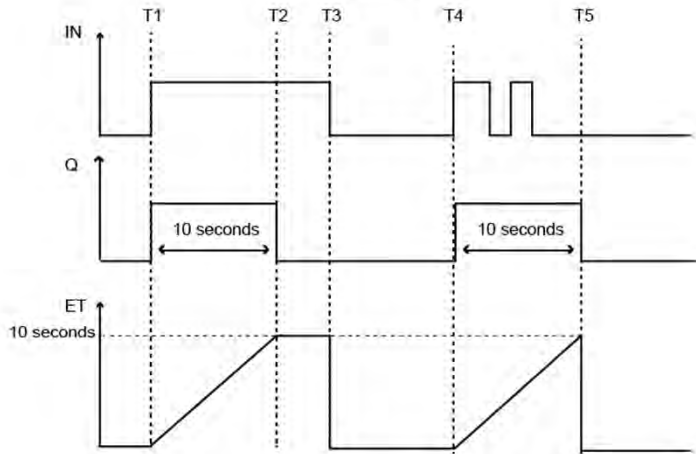
IEC timers in S7-1200 PLCs use the 32-bit Time data type for preset time (PT) and elapsed time (ET) values. Time data is stored as a signed double integer interpreted as milliseconds. However, negative time values cannot be used with timer instructions. This means that the range of values that can be represented by timer preset and elapsed times is 0 to +2,147,483,647 milliseconds, which is equal to 24 days, 20 hours, 31 minutes, 23 seconds, and 647 milliseconds.

In STEP 7, preset time is entered using this format: T#(days)d(hours)h(minutes)m(seconds)s(milliseconds)ms. For example, T#12d10h30m21s253ms is a valid time, but it is not necessary to specify all units of time. Therefore, T#5h10s and T#500h are also valid.

TP Timer Box Instruction



Timing Diagram

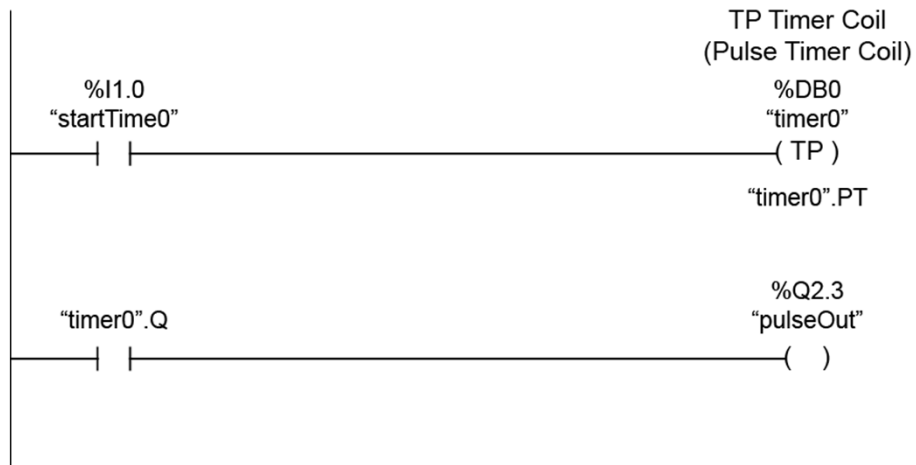


A TP timer, also called a pulse timer, is an instruction that generates a pulse at output parameter (Q) when the input parameter (IN) transitions from 0 to 1, unless the timer has already started timing. The pulse duration is defined by the preset time parameter (PT). The elapsed time (ET) increases from 0 to the preset time during the pulse duration and is retained until IN transitions from 1 to 0. The PT and ET values are stored in a data block (DB).

The accompanying graphic shows a TP timer in a network and an associated timing diagram. Note that PT is set to 10 seconds. When the I1.0 contact closes at time T1, IN transitions from 0 to 1. This starts the timer timing and causes Q and output Q2.3 to transition from 0 to 1. At time T2, when ET reaches 10 seconds, the timer stops timing, and Q and output Q2.3 are reset to 0. The 10 seconds elapsed time is retained until time T3 when the I1.0 contact opens, and IN and ET are reset to 0.

This sequence begins again at time T4 when the I1.0 contact closes. However, the TP timer continues to time and Q, and output Q2.3 remain on when IN is reset to 0 during the preset time. At T5, the elapsed time reaches 10 seconds and Q and output Q2.3 are reset to 0. Because IN is already at 0, ET is also reset to 0.

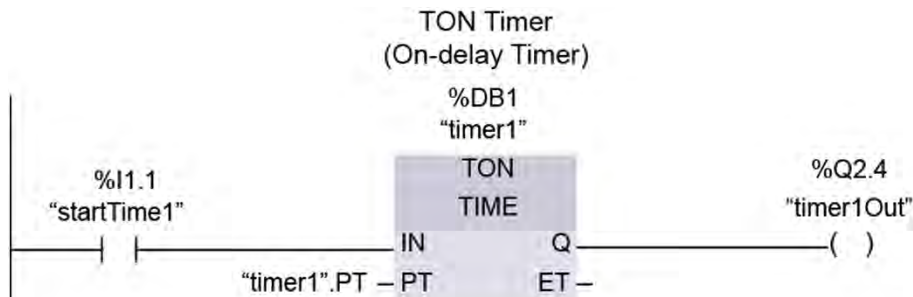
TP Timer Coil Instruction



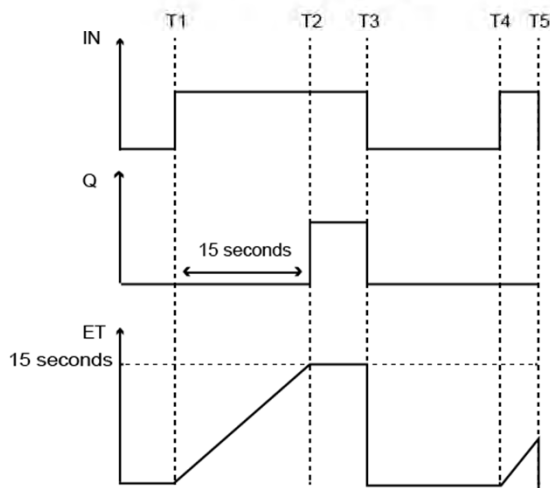
As shown in the accompanying graphic, a LAD program can also use a TP timer coil instruction to perform the same function as a TP timer box instruction.

Like the box instruction, a TP timer coil has an associated data block that stores the PT and ET values.

TON Timer Box Instruction



Timing Diagram

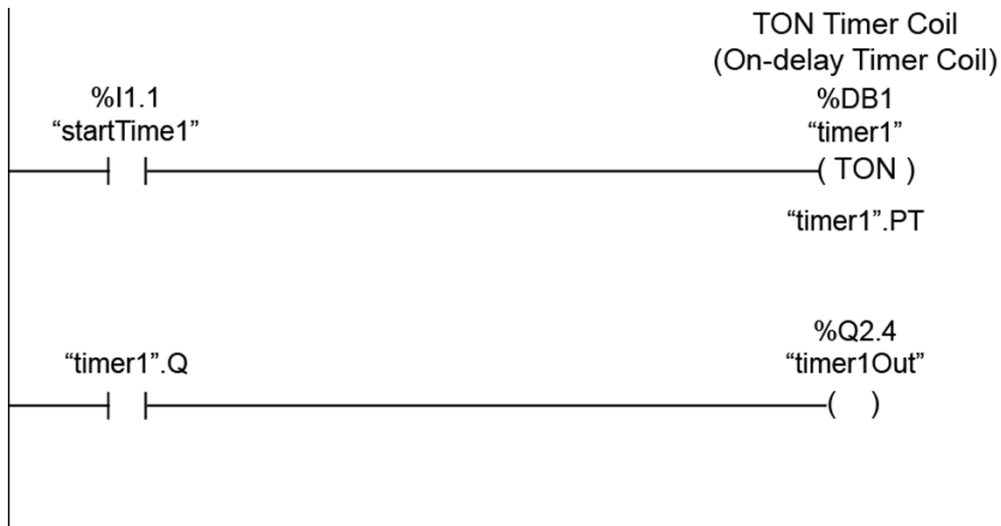


A TON timer, also called an on-delay timer, is an instruction that begins timing when the input parameter (IN) transitions from 0 to 1. When the elapsed time (ET) reaches the preset time (PT), the output parameter (Q) transitions from 0 to 1. Q remains a 1 until IN transitions from 1 to 0. If IN transitions from 1 to 0 before ET is equal to PT, the timer stops timing, ET is reset to 0, and Q remains a 0. The PT and ET values are stored in the data block (DB) associated with the instruction.

The accompanying graphic shows a TON timer in a network and an associated timing diagram. In this example, PT is equal to 15 seconds. At time T1, when the I1.1 contact closes, IN transitions from 0 to 1. This starts the timer timing. At time T2, when ET is equal to 15 seconds, the timer stops timing, and Q and output Q2.4 transition from 0 to 1. At time T3, when the I1.1 contact opens, IN, ET, Q, and output Q2.4 are reset to 0.

This sequence begins again at time T4 when the I1.1 contact closes and IN transitions from 0 to 1. However, at time T5, the I1.1 contact opens, and IN transitions from 1 to 0 before ET is equal to PT, so Q and output Q2.4 remain off, and ET is reset to 0.

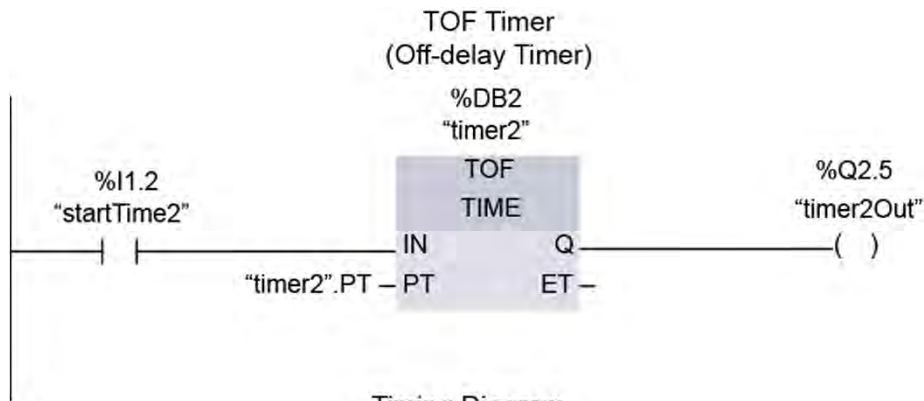
TON Timer Coil Instruction



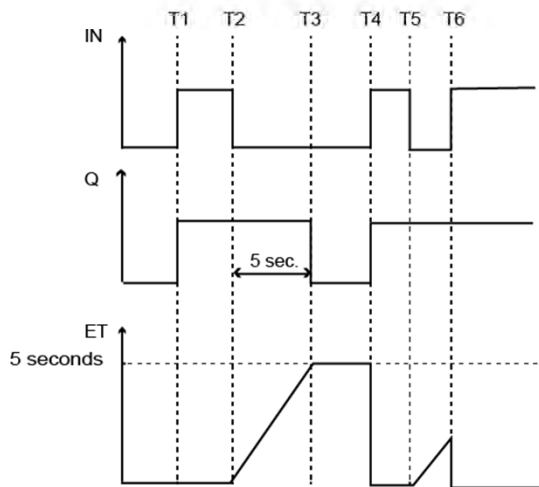
As shown in the accompanying graphic, a LAD program can also use a TON timer coil instruction to perform the same function as a TON timer box instruction.

Like the box instruction, a TON timer coil has an associated data block that stores the PT and ET values.

TOF Timer Box Instruction



Timing Diagram

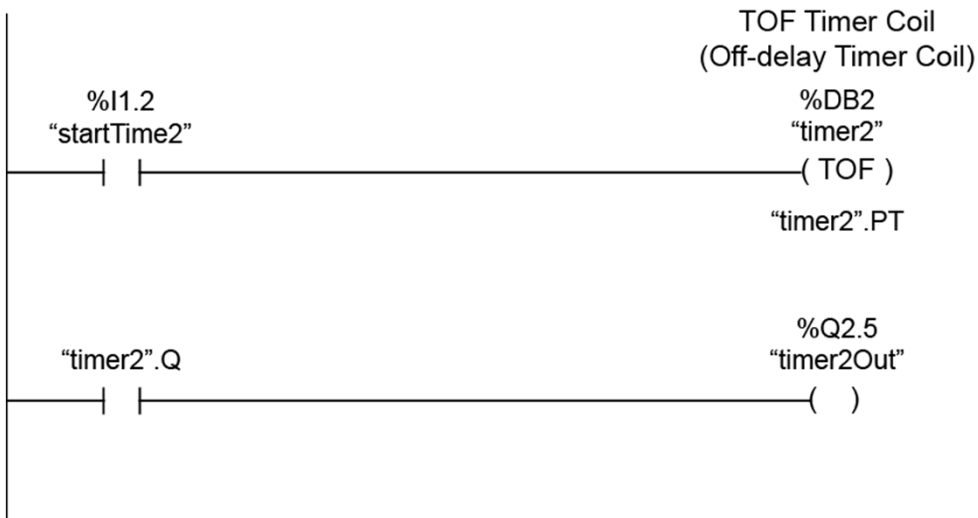


A TOF timer, also called an off-delay timer, is an instruction that sets the output parameter (Q) to 1 when input parameter (IN) transitions from 0 to 1. However, the TOF timer does not begin timing until IN transitions from 1 to 0. When the elapsed time (ET) reaches the preset time (PT), Q transitions from 1 to 0. Q remains a 0 and the elapsed time is retained until the next 0 to 1 transition of IN. The PT and ET values are stored in the data block (DB) associated with the instruction.

The accompanying graphic shows a TOF timer in a network and an associated timing diagram. In this example PT is equal to 5 seconds. When the I1.2 contact closes at time T1, IN, Q, and output Q2.5 transition from 0 to 1. At time T2, when the I1.2 contact opens, IN transitions from 1 to 0, and the timer starts timing. At time T3, when ET is equal to 5 seconds, the timer stops timing, and Q and output Q2.5 transition from 1 to 0.

At time T4, when the I1.2 contact closes and IN and Q transition from 0 to 1, ET is reset to 0. At time T5, when the I1.2 contact opens, IN transitions from 1 to 0, and the timer starts timing. However, at time T6, the I1.2 contact closes and IN transitions from 0 to 1 before ET is equal to PT, so Q and output Q2.5 remain on, and ET is reset to 0.

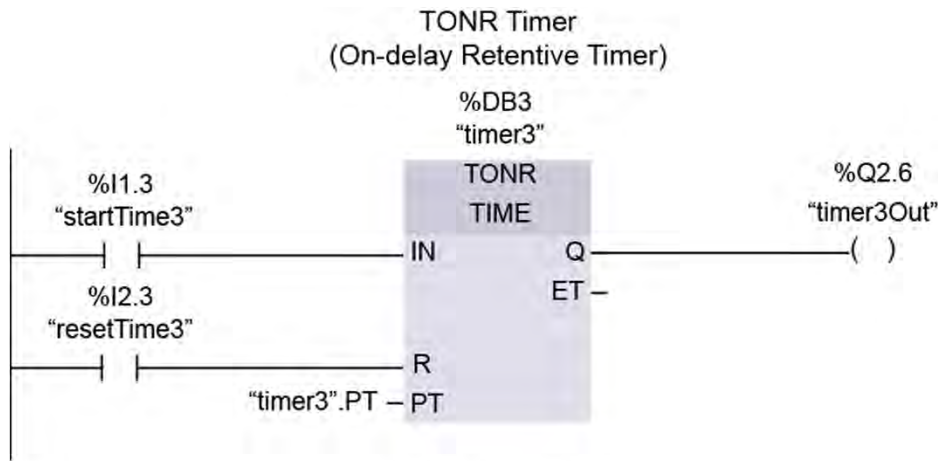
TOF Timer Coil Instruction



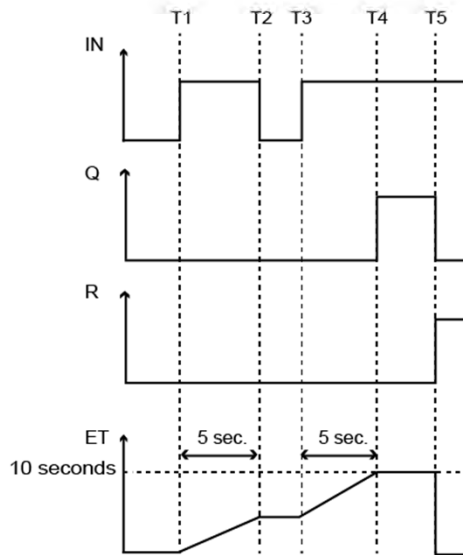
As shown in the accompanying graphic, a LAD program can also use a TOF timer coil instruction to perform the same function as a TOF timer box instruction.

Like the box instruction, a TOF timer coil has an associated data block that stores the PT and ET values.

TONR Timer Box Instruction



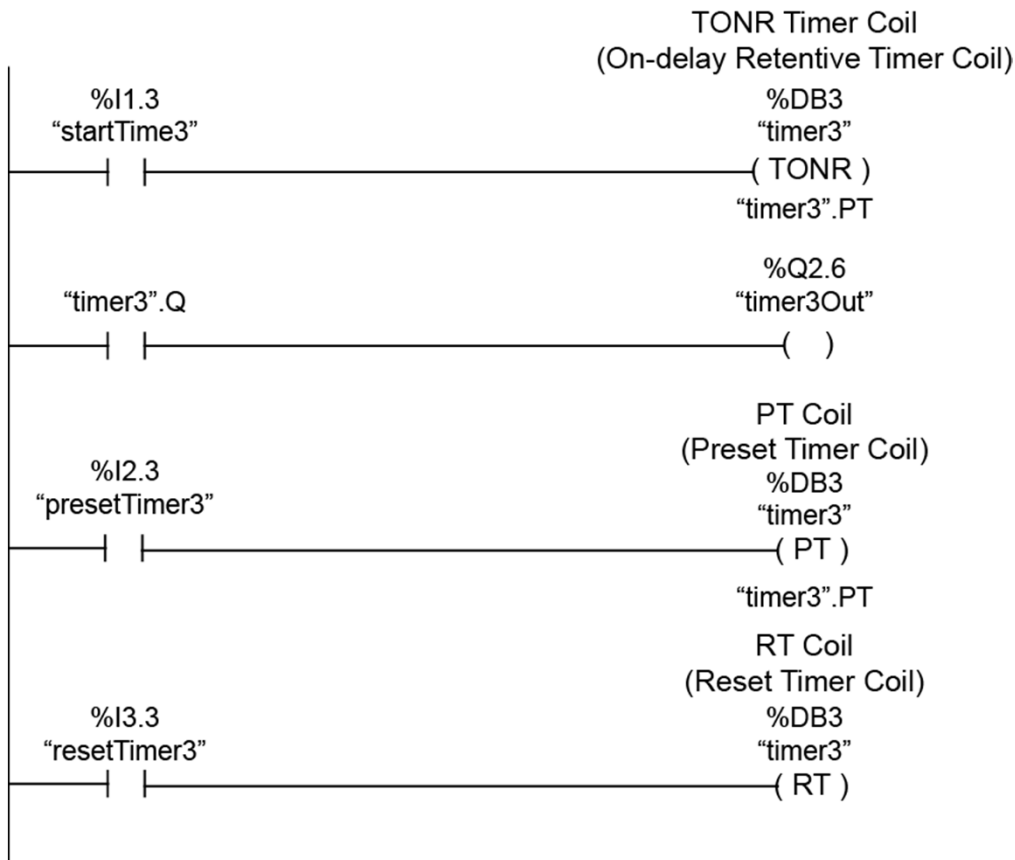
Timing Diagram



A TONR timer, also called an on-delay retentive timer, is an instruction that begins timing when the input parameter (IN) transitions from 0 to 1. If IN transitions from 1 to 0, the elapsed time (ET) is retained. When IN again transitions from 0 to 1, the timer resumes timing from the retained ET value. When ET reaches the preset time (PT), the timer stops timing, and the output parameter (Q) transitions from 0 to 1. When the reset parameter (R) transitions from 0 to 1, ET and Q are reset to 0, even if IN remains a 1. The PT and ET values are stored in the data block (DB) associated with the instruction.

The accompanying graphic shows a TONR timer in a network and an associated timing diagram. In this example PT is set to 10 seconds. When the I1.3 contact closes at time T1, IN transitions from 0 to 1. This starts the timer timing. At time T2, when the I1.3 contact opens, the timer stops timing, but retains the 5 second ET value. At time T3, when the I1.3 contact closes, IN transitions from 0 to 1, and the timer resumes timing. At time T4, when ET is equal to 10 seconds, the timer stops timing, and Q and output Q2.6 transition from 0 to 1. At time T5, when the I2.3 contact closes, R transitions from 0 to 1, and ET, Q, and output Q2.6 are reset to 0.

TONR, PT, and RT Timer Coil Instructions



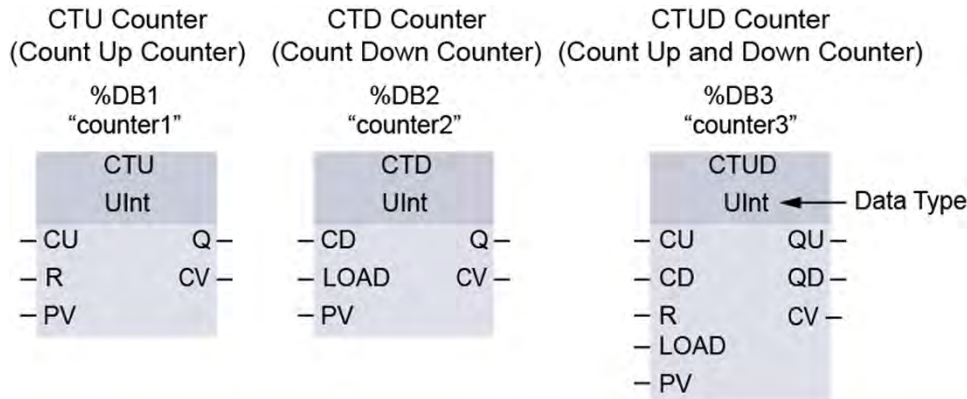
As shown in the accompanying graphic, a LAD program can also use a TONR timer coil instruction to perform the same function as a TONR timer box instruction. Like the box instruction, a TONR timer coil has an associated data block that stores the PT and ET values.

The accompanying graphic also shows a PT timer coil instruction, also called a preset timer coil instruction, and an RT timer coil instruction, also called a reset timer coil instruction.

These instructions can be used with timer box instructions or other timer coil instructions and can be placed in a mid-line or end position in a network.

When the PT timer coil instruction receives power flow, it loads a preset time to the specified timer. When the RT timer coil instruction receives power flow, it resets the specified timer.

IEC Counters



Integer Type	Size	Equivalent Decimal Range
Unsigned Short Integer (USInt)	8 Bits (1 Byte)	0 to 255
Unsigned Integer (UInt)	16 Bits (1 Word)	0 to 65,535
Unsigned Double Integer (UDInt)	32 Bits (1 Double Word)	0 to 4,294,967,295
Signed Short Integer (SInt)	8 Bits (1 Byte)	-128 to +127
Signed Integer (Int)	16 Bits (1 Word)	-32,768 to +32,767
Signed Double Integer (DInt)	32 Bits (1 Double Word)	-2,147,483,648 to +2,147,483,647

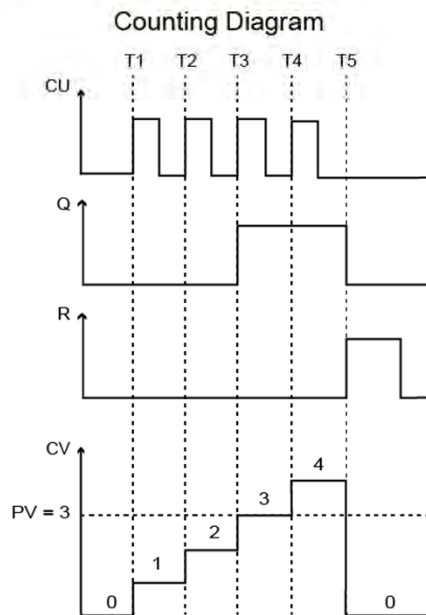
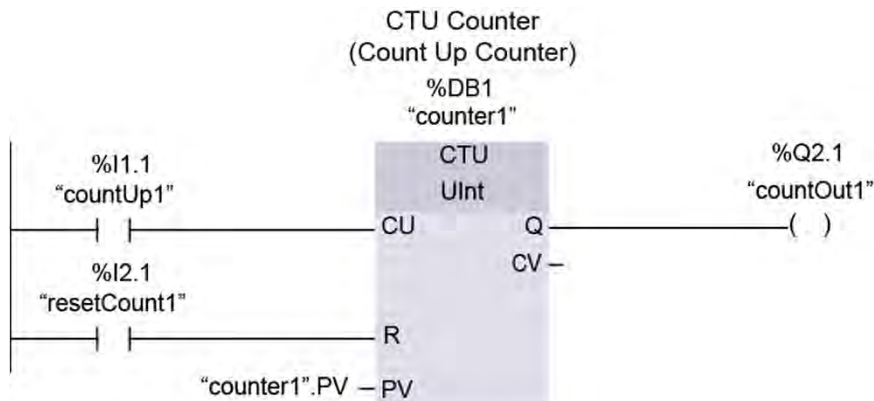
For LAD programs, S7-1200 PLCs support the IEC counter instructions shown in the accompanying graphic.

CU, CD, and CTUD counter instructions use software counters with a maximum counting rate that is limited by the execution rate of the organization block (OB) they are placed in. S7-1200 PLCs also provide high-speed counters for counting events that occur faster than can be handled by IEC counters.

Each IEC counter instruction uses a data block (DB) to maintain counter data. The number of counter instructions that you can use in a user program is limited only by the amount of memory in the CPU.

The amount of memory required for an instruction depends on the data type selected to store count values; therefore, it makes sense to choose the data type that matches the application requirements. The accompanying graphic also shows the data types available for IEC counters in S7-1200 PLCs.

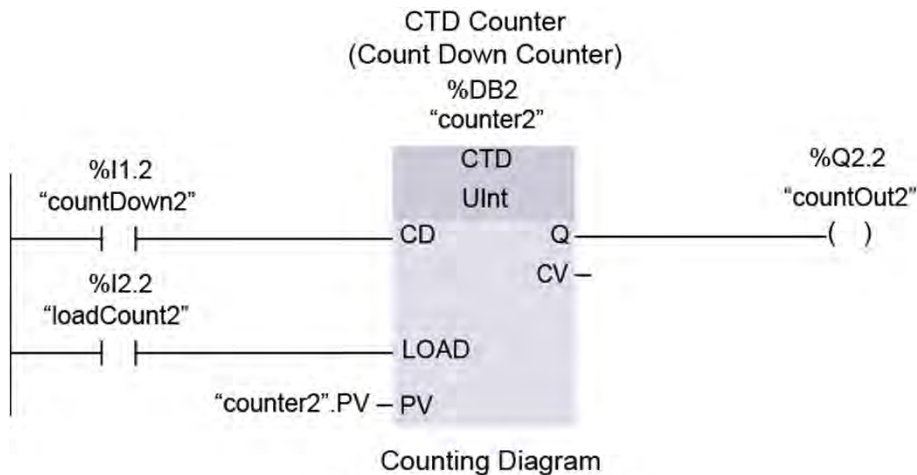
CTU Counter Instruction



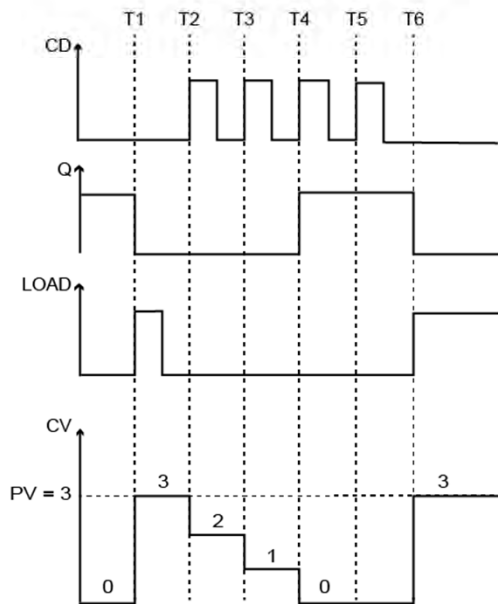
A CTU counter, also called a count up counter, is an instruction that counts up with each 0 to 1 transition of the count up parameter (CU). When the count value parameter (CV) reaches the preset value (PV), the output parameter (Q) transitions from 0 to 1. CV continues to increment with each 0 to 1 transition of CU until the high limit of the data type selected is reached. When the reset parameter (R) transitions from 0 to 1, CV and Q are reset to 0.

The accompanying graphic shows a CTU counter in a network and an associated counting diagram. In this example, PV is equal to 3. Each closure of the I1.1 contact causes a 0 to 1 transition of CU, which increments CV. At time T3, when CV equals PV, Q and output Q2.1 transition from 0 to 1. The count continues at time T4 when CV increments to 4. Because the data type selected for the counter in this example is unsigned integer (UInt), it could count up to 65,535. At time T5, when the I2.1 contact closes, R transitions from 0 to 1, and CV, Q, and output Q2.1 are reset to 0.

CTD Counter Instruction



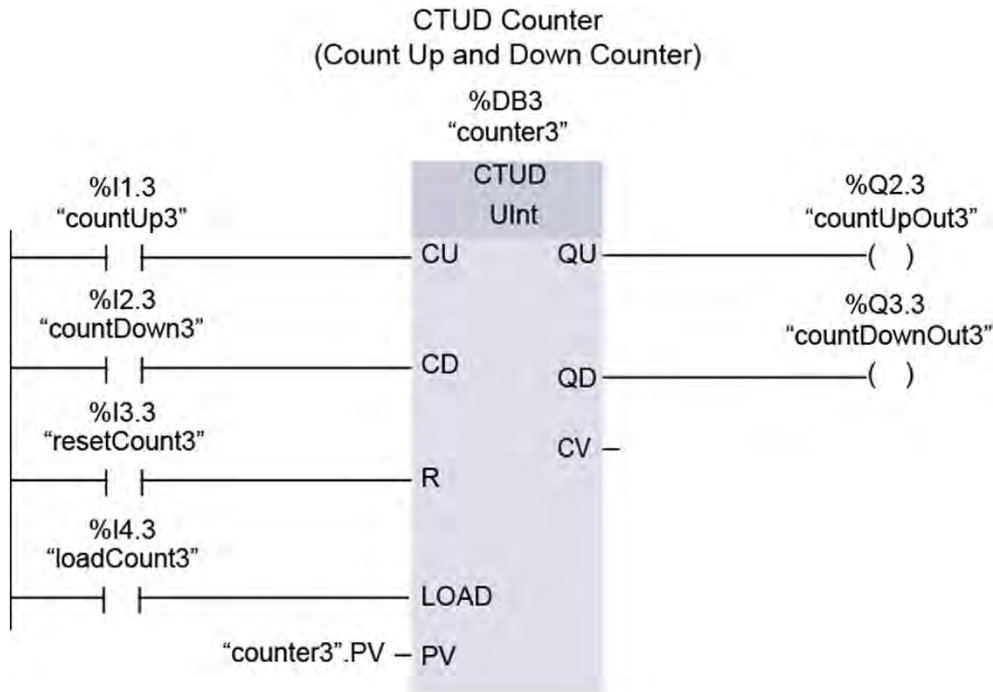
Counting Diagram



A CTD counter, also called a count down counter, is an instruction that counts down with each 0 to 1 transition of the count down parameter (CD). When the LOAD parameter transitions from 0 to 1, the count value parameter (CV) is set equal to the preset value (PV), and the output parameter (Q) is reset to 0. Each 0 to 1 transition of CD decrements CV by 1. When CV reaches 0, Q is set to 1. CV continues to decrement by 1 with each 0 to 1 transition of CD until the low limit of the data type is reached.

The accompanying graphic shows a CTD counter in a network and an associated counting diagram. In this example, PV is equal to 3. At time T1, the I2.2 contact closes causing a 0 to 1 transition of LOAD, which sets CV to 3 and resets Q and output Q2.2 to 0. Prior to time T2, the I2.2 contact opens, and LOAD is reset to 0. Thereafter, each closure of the I1.2 contact causes a 0 to 1 transition of CD, which decrements CV. At time T4, when CV equals 0, Q and output Q2.2 transition from 0 to 1. Because the data type selected for the counter in this example is unsigned integer (UInt), it stops counting down at 0. At time T6, when the I2.2 contact closes, LOAD transitions from 0 to 1, CV is set equal to 3, and Q and output Q2.2 are reset to 0.

CTUD Counter Instruction



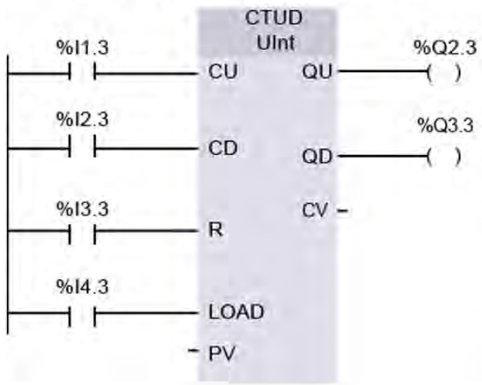
A CTUD counter, also called a count up and down counter, is an instruction that counts up with each 0 to 1 transition of the count up parameter (CU) and counts down with each 0 to 1 transition of the count down parameter (CD). If CU and CD both transition from 0 to 1 at the same time, the count does not change.

The current value of the count is available at the count value parameter (CV). The maximum and minimum values of CV and the present value parameter (PV) are determined by the data type selected for the counter.

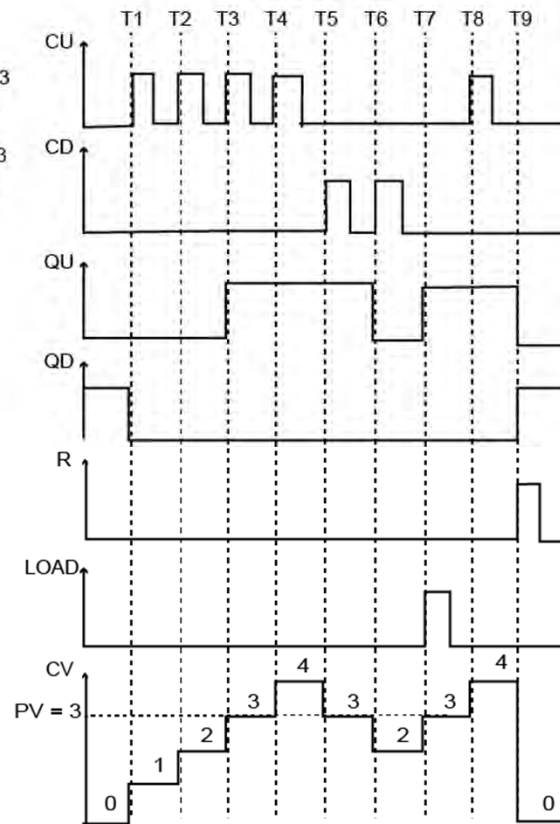
A 0 to 1 transition of the LOAD parameter sets CV equal to PV. A 0 to 1 transition of the reset parameter (R) resets CV to 0. QU is equal to 1 if CV is greater than or equal to PV. QD is equal to 1 if CV is less than or equal to 0.

CTUD Counter Instruction Example

CTUD Counter
(Count Up and Down Counter)



Counting Diagram



The accompanying graphic shows a CTUD counter in a network. For this example, PV is equal to 3. The counting diagram shown in the accompanying graphic is for the following sequence of events.

- At the start of the sequence, CV is 0, so QD and output Q3.3 are 1, and QU and output Q2.3 are 0.
- At time T1, the I1.3 contact closes, CU transitions from 0 to 1, and CV is incremented to 1. Because CV is greater than 0, QD and output Q3.3 are reset to 0.
- Prior to time T2, the I1.3 contact opens. At time T2, the I1.3 contact closes, CU transitions from 0 to 1, and CV increments to 2.
- Prior to time T3, the I1.3 contact opens. At time T3, the I1.3 contact closes, CU transitions from 0 to 1, and CV increments to 3. Because CV is equal to PV, QU and output Q2.3 are set to 1.
- Prior to time T4, the I1.3 contact opens. At time T4, contact I1.3 closes, CU transitions from 0 to 1, and CV increments to 4.
- At time T5, the I2.3 contact closes, CD transitions from 0 to 1, and CV decrements to 3.
- Prior to time T6, the I2.3 contact opens. At time T6, contact I2.3 closes, CD transitions from 0 to 1, and CV decrements to 2. Because CV is now less than PV, QU and output Q2.3 are reset to 0.
- At time T7, the I4.3 contact closes, LOAD is set to 1, CV is set equal to PV, and QU and output Q2.3 are set to 1.
- Prior to time T8, contacts I1.3 and I4.3 open. At time T8, the I1.3 contact closes, CU transitions from 0 to 1, and CV is incremented to 4.
- At time T9, the I3.3 contact closes, R, QD, and output Q3.3 are set to 1, and CV, QU, and output Q2.3 are reset to 0.

Simulators



Engineered to provide a real-world experience, Siemens simulators are fully functional, ready-to-use systems available in a variety of configurations.

System-level design makes the simulators an invaluable tool for program testing and debugging, reinforcing learning, shop floor troubleshooting, and more. With portable construction and hard-shell cases, they can be easily transported. Custom-built systems are also available.

For additional information: www.usa.siemens.com/sitrain

Chapter 4 – Additional Information



This chapter covers the following topics:

- **Additional S7-1200 PLC Capabilities**
- **Additional Siemens Products**

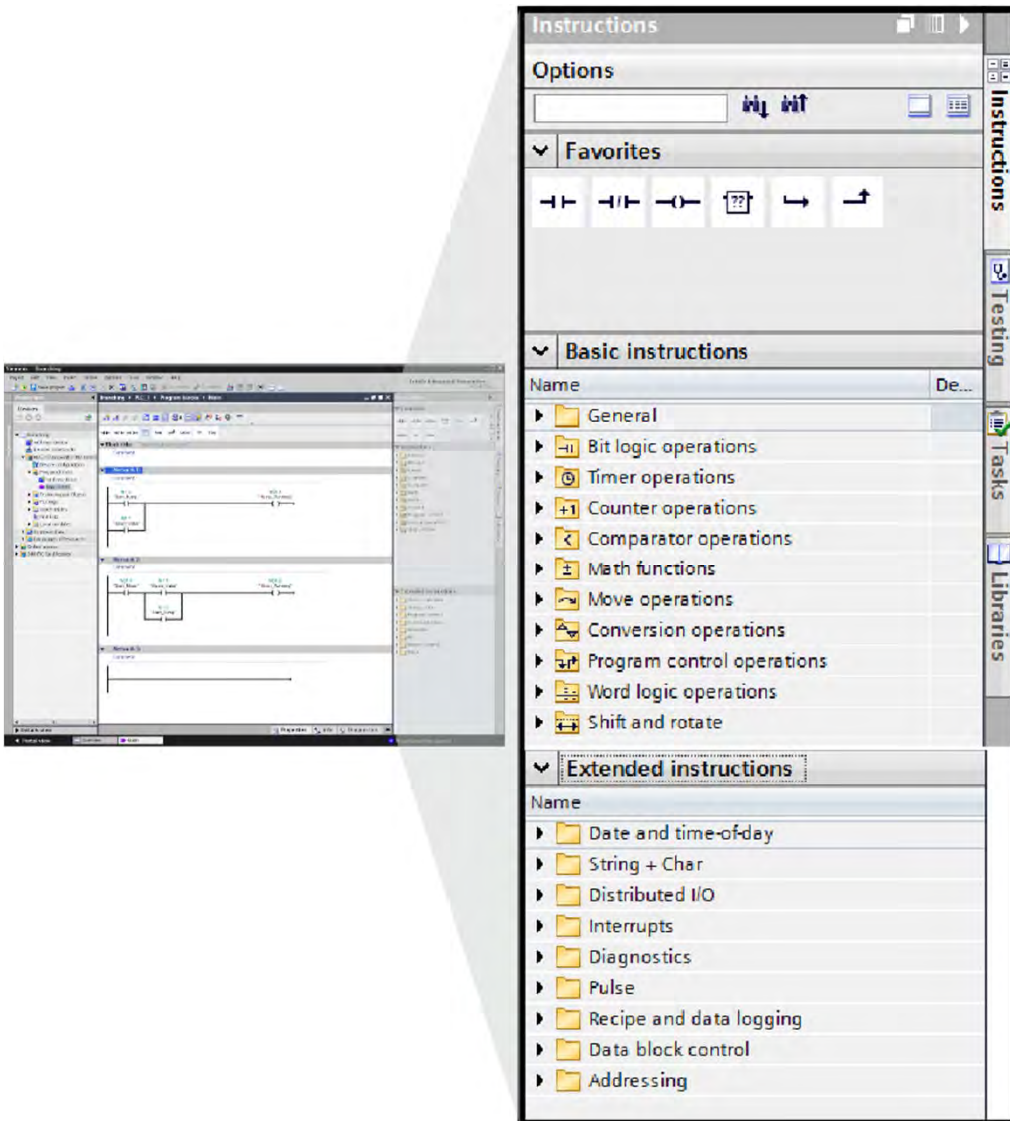
Additional LAD Instructions

The instructions shown in the previous lessons are only a few of the many types of instructions that can be included in a Siemens PLC LAD program. A LAD program can include a variety of basic and extended instructions.

The basic instructions are grouped in the following categories: bit logic operations, timer operations, counter operations, comparator operations, math functions, move operations, conversion operations, program control operations, word logic operations, and shift and rotate operations.

The extended instructions are grouped in the following categories: date and time-of-day, distributed I/O, interrupts, diagnostics, pulse, recipe and data logging, data block control, and addressing.

In combination, these instructions provide SIMATIC S7 PLCs with the full range of LAD programming capabilities to meet a broad range of applications.



Additional S7-1200 PLC Programming Languages

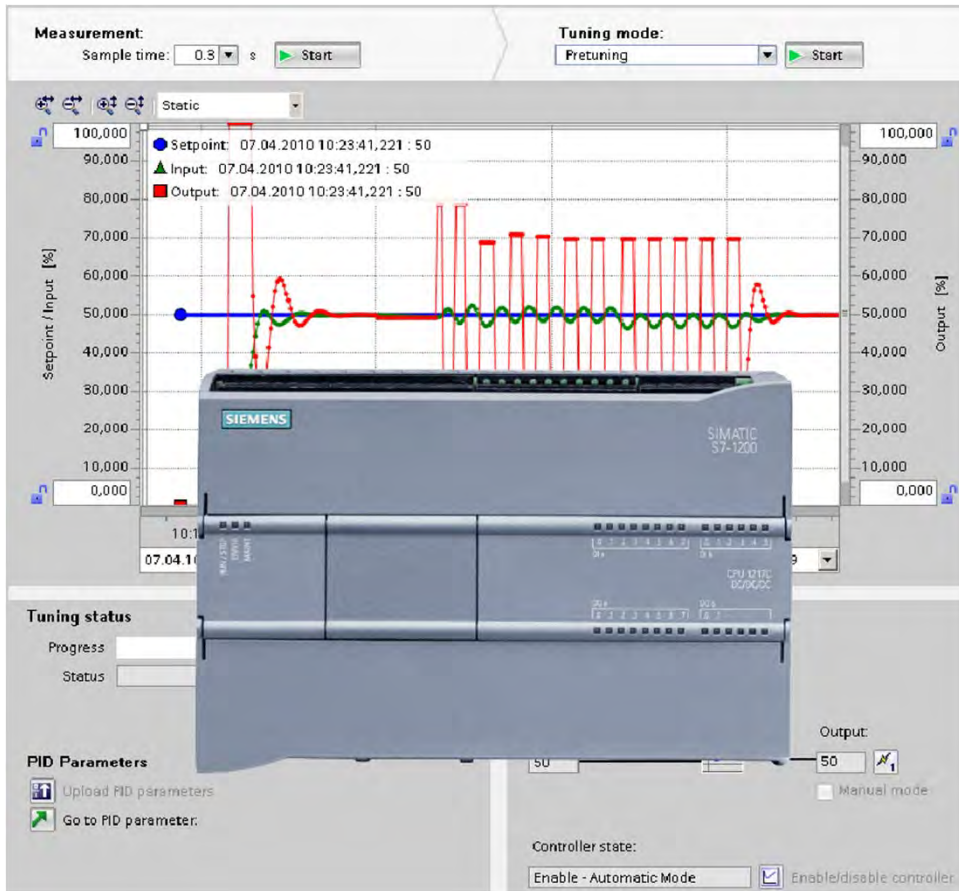


In addition to the LAD programming language, S7-1200 PLCs support the following IEC 61131 programming languages.

- Function block diagram (FBD) is a graphical programming language similar to LAD. However, rather than using contact logic, FBD uses Boolean logic instructions with similar functionality to the logic gates of a digital logic circuit.
- Structured control language (SCL), referred to in the IEC 61131 specification as structured text, is a high-level, PASCAL-based programming language. SCL is generally considered the best PLC programming language to use for complex algorithms and processing of complex data structures.

Because the choice of a programming language is done by program block, the most appropriate language can be chosen for each block based on the tasks performed by that block.

Integrated Technology



S7-1200 PLCs have built-in technology such as high-speed inputs, high-speed outputs, and PID control that reduces the need for specialty modules, reducing space and cost.

High-speed inputs

Up to six high-speed counters can be used. Three inputs at 100 kHz and three inputs at 30 kHz are seamlessly integrated for counting and measuring functionality.

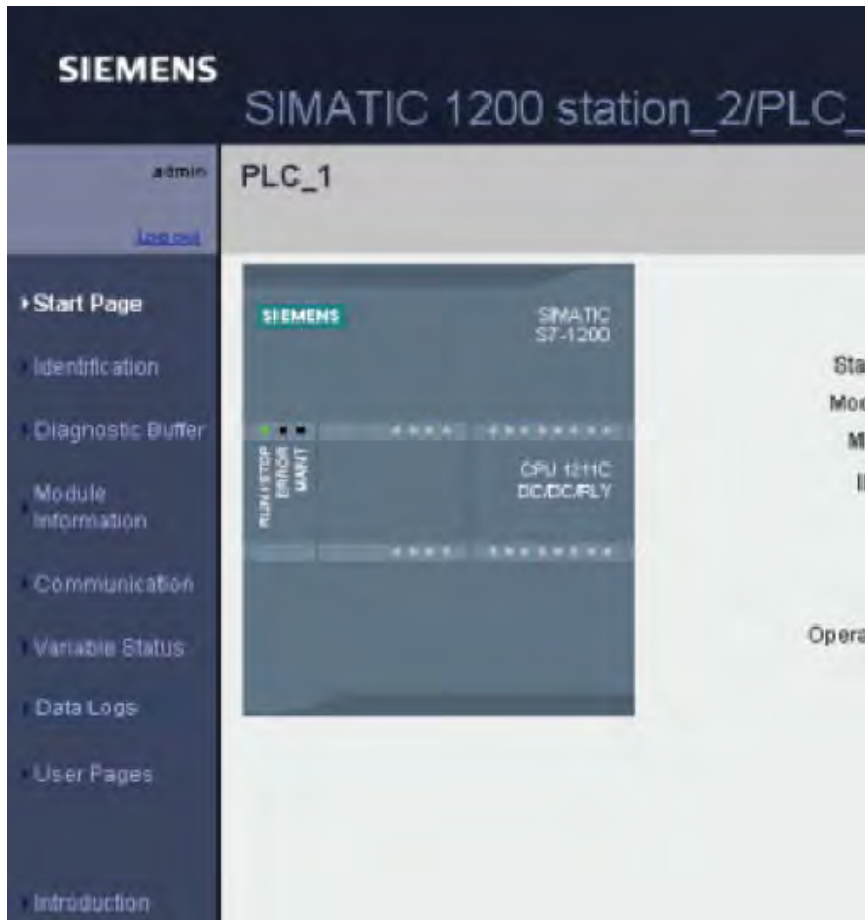
High-speed outputs

Two high-speed pulse train outputs at 100 kHz are integrated for controlling the speed and position of a stepper motor or a servo drive. They can alternatively be used as pulse width-modulated outputs for controlling the speed of a motor, positioning a valve, or controlling a heating element.

PID control

PID control loops with auto-tune allow for simple closed-loop process control applications.

Integrated Web Server



The web server for the S7-1200 provides web Page access to data about your CPU and process data. You can access S7-1200 web Pages from a PC or a mobile device. The web server displays the Pages in a format and size compatible with the device you use to access the Pages.

Examples of standard S7-1200 web Page information include:

- Start Page with general information about the CPU
- Identification Page with detailed CPU information
- Diagnostic buffer Page with diagnostic information
- Module information Page with information about communication and signal modules
- Communication Page with network and other communication information
- Variable status Page with CPU variables and related information

The S7-1200 also supports user-defined web Pages for access to CPU data.

SITRAIN® Training for Industry



Online Self-paced Learning – Programs with maximum flexibility so students can easily fit courses into their busy schedules



Virtual Instructor-led Learning - Classroom lectures delivered in the convenience of your home or office



Classroom Learning - Expert and professional instructors, proven courseware, and quality workstations combine for the most effective classroom experience possible at your facility or ours



How-to Video Library - Quick, affordable, task-based learning options for a broad range of automation topics for training or purchase



Simulators - World-class simulation systems available for training or purchase

For additional information: www.usa.siemens.com/sitrain

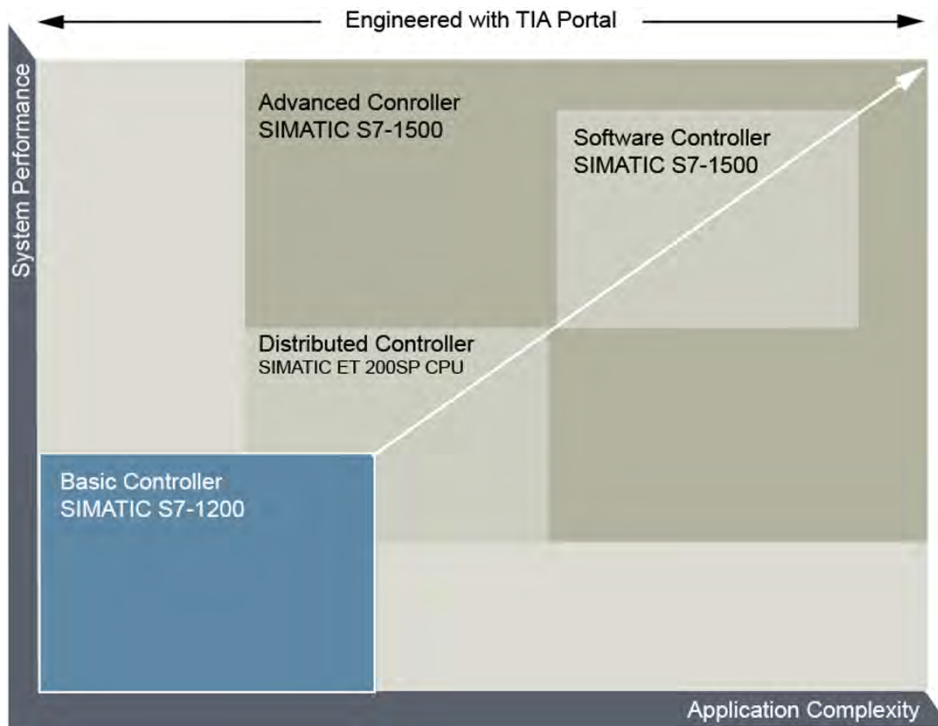
Chapter 4 – Additional Information



This chapter covers the following topics:

- Additional S7-1200 PLC Capabilities
- **Additional Siemens Products**

SIMATIC Controllers

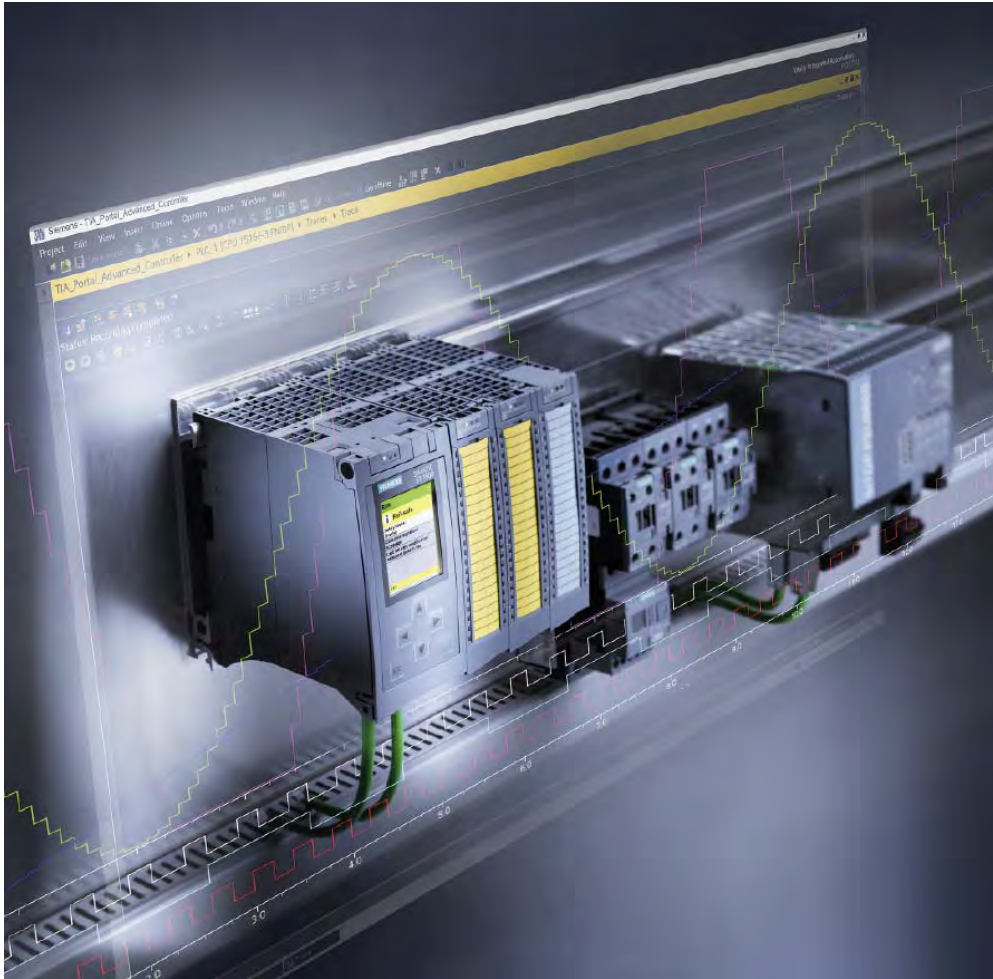


Because every machine or plant has individual requirements for system performance and application complexity, Siemens offers a full range of PLCs to provide you with the perfect control solution for each application.

This course has focused on S7-1200 basic controllers designed for small to medium applications. Additional Siemens PLCs include S7-1500 advanced controllers for medium and complex applications, S7-1500 software controllers for PC-based applications, and ET 200SP distributed controllers for decentralized applications.

All basic, advanced, and distributed controllers are also available as failsafe versions for implementation of failsafe applications.

S7-1500 Advanced Controller



The S7-1500 Advanced Controller sets the standard for system performance and usability. The seamless integration of the S7-1500 controller into the TIA Portal offers advantages such as shared data management, a uniform operating concept, and centralized services. This simplifies the use of universal functions.

S7-1500 PLCs are quick and easy to install and set up. For you, this means a shorter time-to-market and a higher rate of return on your investment.

There are various S7-1500 CPU versions available in several performance categories. The portfolio includes standard CPUs and standard CPUs with the option of integrating C/C++ code. Compact CPUs have integrated digital and analog I/O channels and allow counter inputs and pulse inputs to be directly recorded on the CPU. Failsafe CPUs can concurrently run standard and safety programs. Technology CPUs incorporated motion control functions for demanding solutions.

S7-1500 Software Controller



S7-1500 software controller for PC-based automation with SIMATIC Industrial PCs is operated independently of Windows, thus offering a high level of system availability. This is advantageous for fast controller ramp-up and makes it possible to update or reboot while the control system is running.

S7-1500 software controller features both know-how and access protection as well as important automation functions for motion control and interfaces to PROFIBUS and PROFINET. Use TIA Portal for efficient engineering.

S7-1500 software controller ODK supports the development of Windows and real-time library functions and enables the integration of Code C++ higher-level language programs.

ET 200SP Distributed Controller



ET 200SP distributed controllers feature the same functionalities as CPU 1511 and CPU 1513 for S7-1500 controllers. This includes integrated trace functions, integrated system diagnostics, integrated security, integrated web server, and symbolic programming.

ET 200SP standard and failsafe CPUs come with 3 integrated Ethernet ports, 2 Ethernet connection ports can be flexibly selected by means of bus adapters as a cost saving feature.

ET 200SP CPUs can also be configured as interface modules with the PROFINET iDevice functionality. An iDevice is a local IO controller that also functions as an IO device for a central IO controller. The central IO controller has direct access to the IO image in the local controller, so communication is simple and fast, and no complex programming or additional communication hardware are required.

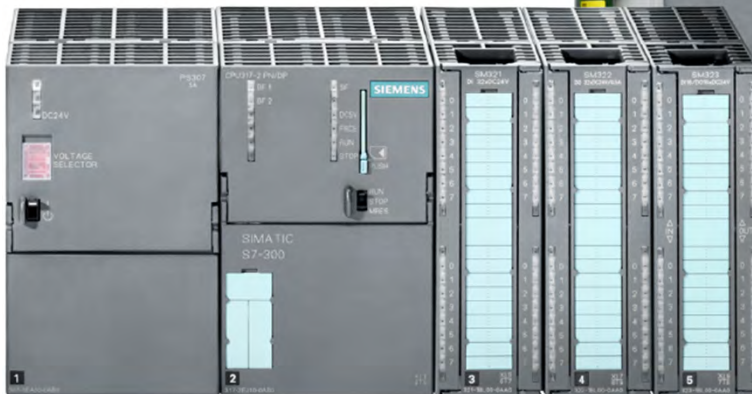
ET 200SP CPUs support the PROFINET energy profile for improved energy efficiency, PROFINET isochronous real-time (IRT) communication for motion control, and DP master communication with PROFIBUS DP devices.

S7-300 and S7-400 Controllers

S7-400 PLC



S7-300 PLC



S7-300 PLCs have a space-saving modular design and are suited to a wide range of automation tasks. SIMATIC S7-300 CPUs are available in Standard, Compact (C), Failsafe (F), and Technology (T) versions.

S7-400 PLCs have high processing speeds and are especially suitable for data-intensive tasks such as coordinating overall plants and controlling lower-level systems. S7-400 CPUs are available in Standard, Failsafe (F), and High-availability (H) versions.

S7-300 and S7-400 PLCs are configured and programmed using STEP 7 Professional (TIA Portal) or a version of STEP 7 Professional that is not in TIA Portal.

ET 200 Distributed I/O Systems



ET 200 distributed I/O systems offer standard and failsafe solutions for use in a control cabinet or without a cabinet directly on the machine.

The ET 200 modular design includes digital and analog I/O, intelligent modules with CPU functionality, safety technology, motor starters, frequency converters, and technology modules.

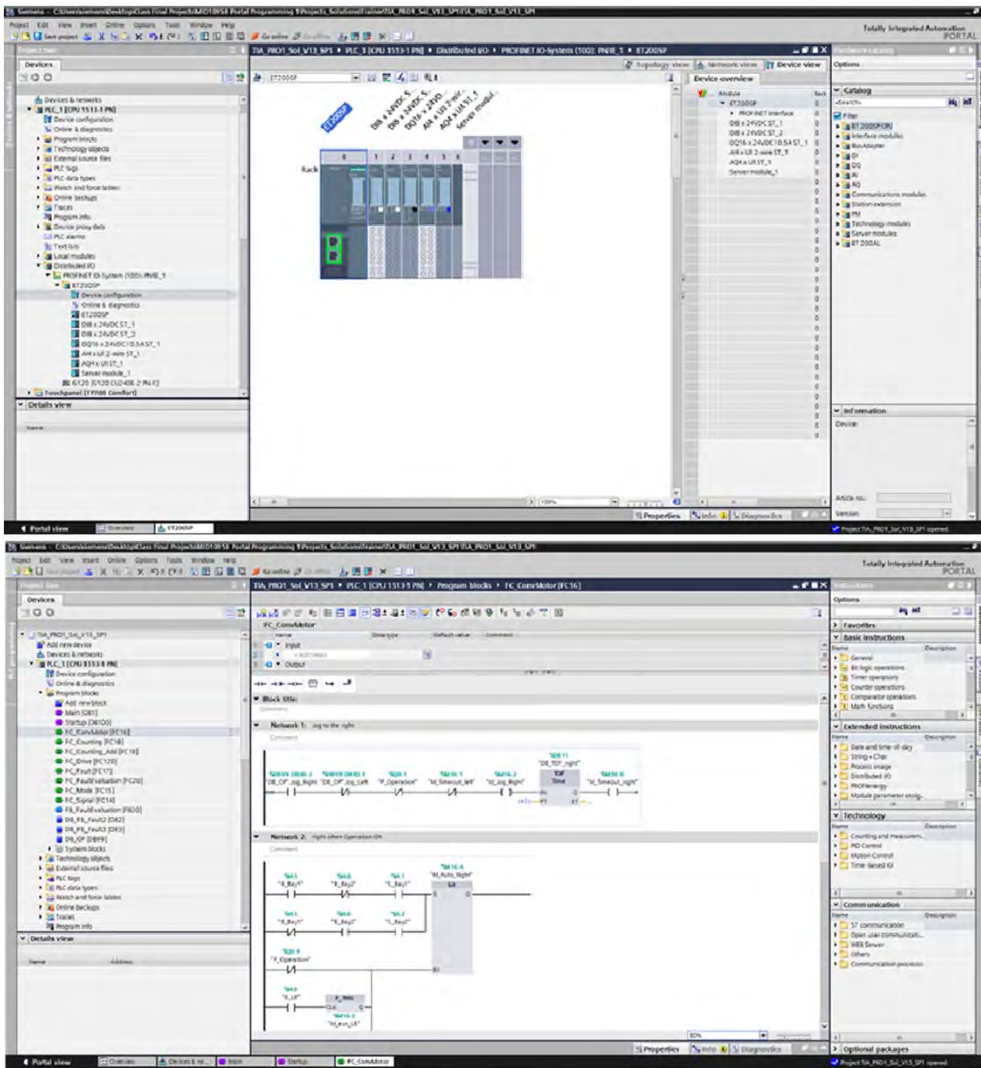
TIA Portal makes the integration of safety technology easier and more efficient.

ET 200 distributed I/O systems can be integrated into your existing automation system via PROFIBUS DP or PROFINET.

Systems designed for use in a control cabinet include: ET 200SP, ET 200MP, ET 200S, ET 200M, and ET 200iSP.

Systems designed for use directly on a machine without a control cabinet include: ET 200AL, ET 200pro, ET 200eco PN, and ET 200eco.

STEP 7 Professional (TIA Portal)



STEP 7 Professional (TIA Portal) is used for all phases of development and operation for SIMATIC controllers. It incorporates intuitive features for all tasks, enabling the efficient creation of user programs.

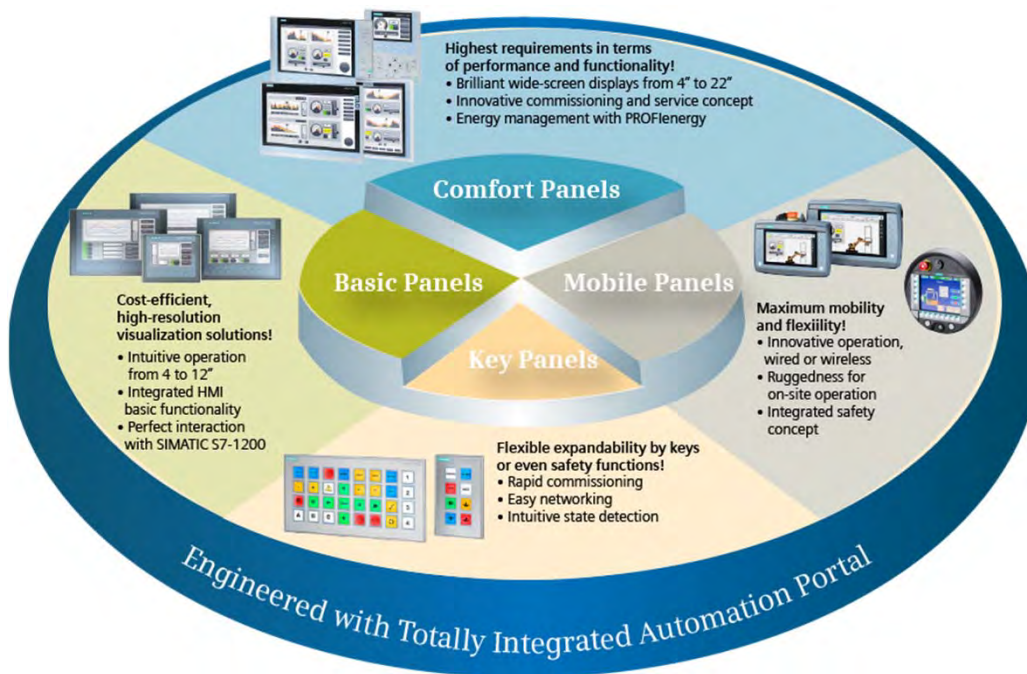
Within STEP 7 Professional (TIA Portal) the various editors are embedded in a common working environment to ensure data consistency and availability.

In addition to the LAD, FBD, and SCL programming languages included in STEP 7 Basic (TIA Portal), STEP 7 Professional (TIA Portal) also includes the following languages.

- Statement list (STL), also known as Instruction list (IL), is a text-based programming language with the same capabilities as LAD and FBD.
- GRAPH, also known as sequential function chart (SFC), is a graphical programming language used for sequential control programs.

STEP 7 Professional also includes S7-PLCsim software, which simulates a controller for functional testing of user programs on the programming device. System documentation and TeleService software are also included.

SIMATIC HMI Panels



SIMATIC HMI Panels are available for virtually any application and can be intuitively configured in TIA Portal using SIMATIC WinCC software. This results in increased engineering efficiency, especially when additional Siemens automation products are used. The interaction with STEP 7 does away with multiple entries and ensures maximum data consistency.

Panel models have Integrated functionality across all display sizes. Because the software is scalable, you can start with a small solution and expand it at any time. Innovative graphical user interfaces open up new possibilities for intuitive operation and monitoring.

SIMATIC HMI includes the following panel types.

- SIMATIC HMI Comfort Panels: Designed for applications requiring the highest performance and functionality.
- SIMATIC HMI Basic Panels: Cost-efficient, high resolution visualization solutions.
- SIMATIC HMI Mobile Panels: Maximum mobility and flexibility.
- SIMATIC HMI Key Panels: Flexible expandability by keys or even safety functions.

SITRAIN World



From the basics to advanced specialist skills, Siemens SITRAIN courses deliver extensive expertise directly from the manufacturer and encompass the entire spectrum of Siemens Industry products and systems.

Worldwide, SITRAIN courses are available in over 200 locations in over 60 countries.

For additional information including a SITRAIN world map and SITRAIN contacts worldwide:

<http://sitrain.automation.siemens.com/sitrainworld/Default.aspx>