

# SIEMENS

## SIMATIC

### S7-1200

### Getting started with S7-1200

#### Getting Started

#### Preface

---

#### Quick review

---

1

#### Installation

---

2

#### Create a simple latch circuit

---

3

#### Complete the user program

---

4

#### Use a watch table for monitoring

---

5

## Legal information

### Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

<b>⚠ DANGER</b>
indicates that death or severe personal injury <b>will</b> result if proper precautions are not taken.
<b>⚠ WARNING</b>
indicates that death or severe personal injury <b>may</b> result if proper precautions are not taken.
<b>⚠ CAUTION</b>
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.
<b>CAUTION</b>
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.
<b>NOTICE</b>
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

### Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

### Proper use of Siemens products

Note the following:

<b>⚠ WARNING</b>
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

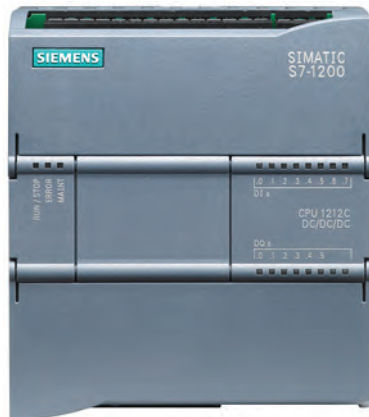
### Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

### Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Preface



As part of the SIMATIC commitment to "totally integrated automation" (TIA), the S7-1200 product family and the STEP 7 Basic programming package give you the tools for creating your automation application. The compact design, low cost, and powerful features make the S7-1200 compact controller a perfect solution for controlling small applications.

The examples in this document guide your first steps in the world of S7-1200, teaching you the basic skills within the shortest possible time.

---

## Note

**Review the *S7-1200 Programmable Controller System Manual* for safety guidelines**

This quick-start guide provides only a quick introduction to the world of S7-1200. Before performing the exercises in this quick-start guide, review the safety guidelines and other information in the *SIMATIC S7-1200 Programmable Controller System Manual*.

---

## Parts list

Description	Quantity	Order number
CPU 1212C AC/DC/RLY	1	6ES7212-1BD30-0XB0
Simulator: SIM 1274 (digital inputs)	1	6ES7274-1XF30-0XA0
Standard Ethernet CAT5 cable	1	6XV1850-2GH20
STEP 7 Basic programming software	1	6ES7822 -0AA00-0YA0

The starter kit also contains the S7-1200 documentation collection CD.

---

## Note

**You must provide a power cord for the CPU.**

The CPU 1212C AC/DC/RLY requires 120/240 VAC. You must provide a power cord and connect it to the CPU. For any other CPU, refer to the *S7-1200 Programmable Controller System Manual* for power requirements and installation procedures.

---

## Contact information

For additional information, refer to the *S7-1200 programmable controller system manual*. Use the customer support web site (<http://www.siemens.com/automation/support-request>) to search for specific information about products or to contact technical support representatives.

Contact your Siemens distributor or sales office for assistance in answering any technical questions, for training, or for ordering S7 products. Because your sales representatives are technically trained and have the most specific knowledge about your operations, process and industry, as well as about the individual Siemens products that you are using, they can provide the fastest and most efficient answers to any problems you might encounter.



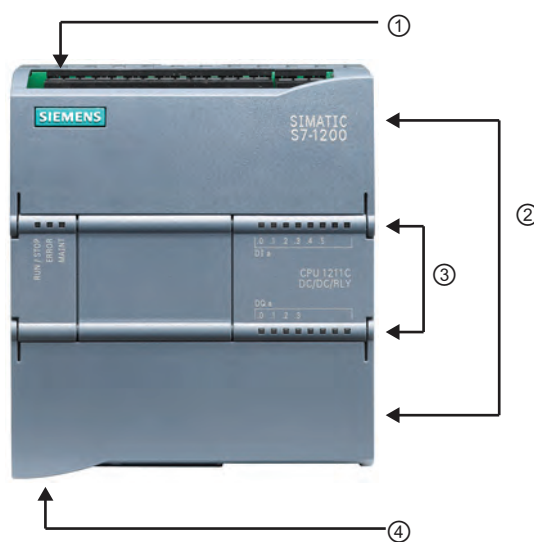
# Table of contents

	<b>Preface</b> .....	<b>3</b>
<b>1</b>	<b>Quick review</b> .....	<b>7</b>
1.1	Basic information about how the CPU works.....	8
1.1.1	Tasks performed every scan cycle .....	8
1.1.2	Operating modes of the CPU.....	9
1.1.3	Digital data stored in bits, bytes, words and more .....	10
1.1.4	Data types supported by the S7-1200 .....	11
1.1.5	Memory areas of the S7-1200 .....	12
1.2	From schematic diagram to user program.....	13
1.2.1	Converting the schematic to LAD instructions .....	14
1.2.2	Basic electrical circuits.....	16
1.3	Basic features of STEP 7 Basic.....	18
1.3.1	Help when you need it .....	19
1.3.2	Printing from the online help .....	20
1.3.3	Inserting instructions .....	21
1.3.4	Dragging and dropping between editors.....	22
1.3.5	Detecting the configuration of a CPU .....	23
1.3.6	Changing the settings for the work area .....	24
<b>2</b>	<b>Installation</b> .....	<b>25</b>
2.1	Read this first for safety information .....	25
2.2	Dimensions and guidelines .....	26
2.3	Connecting the hardware.....	27
2.4	Installing STEP 7 Basic.....	28
<b>3</b>	<b>Create a simple latch circuit</b> .....	<b>29</b>
3.1	Create a project for your user program.....	30
3.2	Create a simple network in your user program.....	31
3.3	Enter the tags and addresses for the instructions .....	33
3.4	Configure the CPU.....	35
3.5	Download the user program to the CPU.....	37
3.6	Test the operation of your sample user program.....	38
3.7	Reviewing the tasks for the first exercise .....	39
<b>4</b>	<b>Complete the user program</b> .....	<b>41</b>
4.1	Add an on-delay timer to delay the start of an output.....	42
4.2	Turn on an output for 5 seconds .....	45
4.3	Editing your user program for a different structure .....	48

<b>5</b>	<b>Use a watch table for monitoring .....</b>	<b>51</b>
5.1	Create a watch table .....	52
5.2	Monitor the data values in your CPU .....	53
5.3	Monitor the status in the LAD editor.....	54
5.4	Force an input to a specific value .....	54
5.5	Going online .....	56
5.6	Building on a good beginning.....	56
	<b>Index.....</b>	<b>57</b>

## Quick review

With an integrated power supply and a variety of on-board input and output circuits, the S7-1200 CPU is a powerful controller. After you download your program, the CPU monitors the inputs and changes the outputs according to the logic of your user program, which can include Boolean logic, counters, timers, and complex math operations.



- ① Power connector
- ② Removable user wiring connectors (behind the doors)
- ③ Status LEDs for the on-board I/O
- ④ PROFINET connector (on the bottom of the CPU)

To communicate with a programming device, the CPU provides a built-in PROFINET port. With the PROFINET network, the CPU can communicate with HMI panels or another CPU.

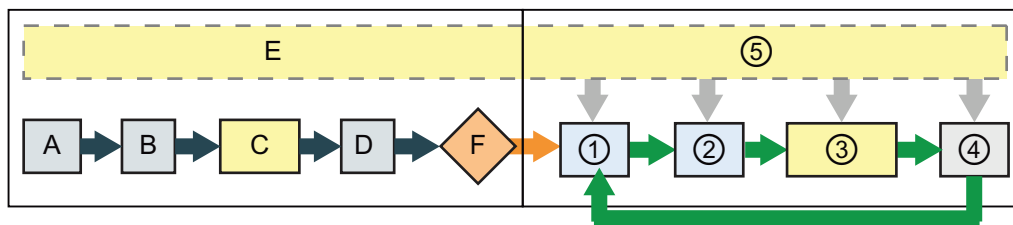
Feature	CPU 1212C
Physical size	90 mm x 100 mm x 75 mm
<ul style="list-style-type: none"> <li>• Work memory</li> <li>• Load memory</li> <li>• Retentive memory</li> </ul>	<ul style="list-style-type: none"> <li>• 25 Kbytes</li> <li>• 1 Mbyte</li> <li>• 2 Kbytes</li> </ul>
<ul style="list-style-type: none"> <li>• On-board digital I/O</li> <li>• On-board analog I/O</li> </ul>	<ul style="list-style-type: none"> <li>• 8 inputs and 6 outputs</li> <li>• 2 inputs</li> </ul>
<ul style="list-style-type: none"> <li>• Process image (Input)</li> <li>• Process image (Output)</li> <li>• Bit memory (M)</li> </ul>	<ul style="list-style-type: none"> <li>• 1024 bytes</li> <li>• 1024 bytes</li> <li>• 4096 bytes</li> </ul>

## 1.1 Basic information about how the CPU works

### 1.1.1 Tasks performed every scan cycle

The scan cycle provides consistent logic through the execution of the user program for a given cycle and prevents the flickering of physical output points that might change state multiple times in the process image output area. Each scan cycle includes writing the outputs, reading the inputs, executing the user program instructions, and performing system maintenance or background processing.

Under default conditions, all digital and analog I/O points use an internal memory area called the "process image" to update the I/O synchronously with the scan cycle. The process image contains a snapshot of the physical inputs ("I memory") and outputs ("Q memory") on the CPU, signal board, and signal modules.



- |              |  |
|--------------|--|
| STARTUP mode | <ul style="list-style-type: none"> <li>A The input area of the process image (I memory) is cleared.</li> <li>B The outputs are initialized with the last value.</li> <li>C Any start-up logic (contained in special code blocks) is executed.</li> <li>D The state of the physical inputs is copied to I memory.</li> <li>E Any interrupt events are queued for processing during RUN mode.</li> <li>F The writing of the output area of the process image (Q memory) to the physical outputs is enabled.</li> </ul> |
| RUN mode     | <ul style="list-style-type: none"> <li>① Q memory is written to the physical outputs.</li> <li>② The state of the physical inputs is copied to I memory.</li> <li>③ The logic of your user program is executed.</li> <li>④ The self-test diagnostics are performed.</li> <li>⑤ Interrupts and communications are processed during any part of the scan cycle.</li> </ul>   |



## 1.1.2 Operating modes of the CPU

The CPU has three modes of operation: STOP mode, STARTUP mode, and RUN mode. Status LEDs on the front of the CPU indicate the current mode of operation.

- In STOP mode, the CPU is not executing the program, and you can download a project.
- In STARTUP mode, the CPU executes any startup logic (if present). Interrupt events are not processed during the startup mode.
- In RUN mode, the scan cycle is executed repeatedly. Interrupt events can occur and be processed at any point within the program cycle phase.

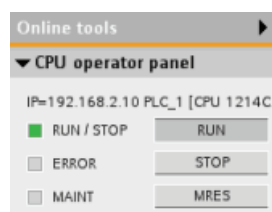
---

### Note

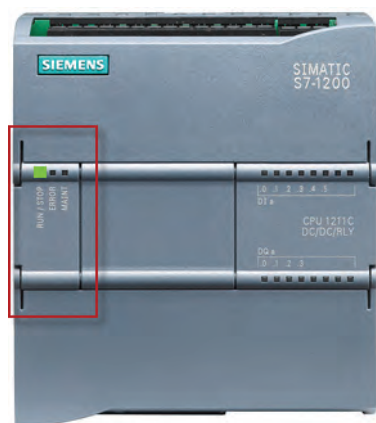
You cannot download a project while the CPU is in RUN mode. You can download your project **only** when the CPU is in STOP mode.

---

The CPU does not have a physical switch for changing the operating modes (STOP or RUN). When you configure the CPU in the device configuration, you configure the start-up behavior in the properties of the CPU. The STEP 7 Basic provides an operator panel for changing the operating mode of the online CPU (Page 56).



Use the button on the operator panel to change the operating mode (STOP or RUN). The operator panel also provides an MRES button for resetting the memory.



The color of the RUN/STOP indicator shows the current operating mode of the CPU:

- Yellow indicates STOP mode.
- Green indicates RUN mode.
- Flashing indicates STARTUP mode.

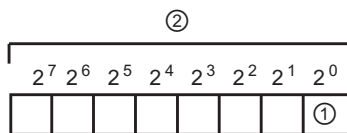
### 1.1.3 Digital data stored in bits, bytes, words and more



$2^0$  The smallest unit of information in a digital system is known as a "bit" (for "binary digit"). A bit stores one of two possible states, either a "0" (false or not true) or "1" (true).

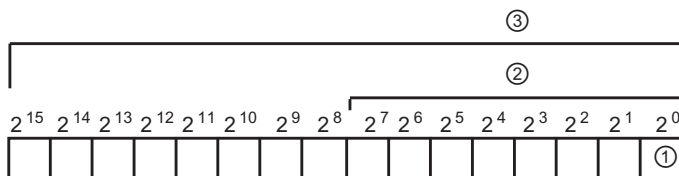
A light switch is an example of a "binary" system with only two states. The light switch determines either a "light on" or a "light off" state, and this "value" can be stored in one bit. The digital value of the light switch answers the question: "Is the light on?" If the light is switched on ("true"), then the value is 1. If the light is switched off ("false"), then the value is 0.

The CPU organizes bits of data into groups. A group of 8 bits ① is called a byte ②.



Each bit in the group is defined exactly by a separate position with its own address. Each bit has a byte address and bit addresses 0 to 7.

A group of 2 bytes is called a "word". A group of 4 bytes is called a "double word".



- ③ Word
- ② Byte
- ① Bit

You use the binary number system (base 2) for counting the numbers. A word can represent an integer number from -32768 to +32767. The bit with the value  $2^{15}$  is used to denote a negative number (when position  $2^{15}$  has the value "1" the number is negative).

**Note**

The CPU also supports a grouping of 8 bytes that form a "long real" data type (LReal) for storing very large or very precise values. The range for the LReal is:  $\pm 2.23 \times 10^{-308}$  to  $\pm 1.79 \times 10^{308}$

### 1.1.4 Data types supported by the S7-1200

The data type specifies not only the size of a data element, but also the structure of the bits within the data.

Data type	Size	Range	Constant Entry Examples
Bool (Boolean)	1 bit	0 to 1	TRUE, FALSE, 0, 1
Byte (byte)	8 bits (1 byte)	16#00 to 16#FF	16#12, 16#AB
Word (word)	16 bits (2 bytes)	16#0000 to 16#FFFF	16#ABCD, 16#0001
DWord (double word)	32 bits (4 bytes)	16#00000000 to 16#FFFFFFFF	16#02468ACE
Char (character)	8 bits (1 byte)	16#00 to 16#FF	'A', 't', '@'
SInt (short integer)	8 bits (1 byte)	-128 to 127	123, -123
USInt (unsigned short integer)	8 bits (1 byte)	0 to 255	123
Int (integer)	16 bits (2 bytes)	-32,768 to 32,767	123, -123
UInt (unsigned integer)	16 bits (2 bytes)	0 to 65,535	123
DInt (double integer)	32 bits (4 bytes)	-2,147,483,648 to 2,147,483,647	123, -123
UDInt (unsigned double integer)	32 bits (4 bytes)	0 to 4,294,967,295	123
Real (real or floating point)	32 bits (4 bytes)	+/-1.18 x 10 <sup>-38</sup> to +/-3.40 x 10 <sup>38</sup>	123.456, -3.4, -1.2E+12, 3.4E-3
LReal (long real)	64 bits (8 bytes)	+/-2.23 x 10 <sup>-308</sup> to +/-1.79 x 10 <sup>308</sup>	12345.123456789 -1.2E+40
Time (time)	32 bits (4 bytes)	T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Stored as: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String (character string)	Variable	0 to 254 byte-size characters	'ABC'
DTL <sup>1</sup> (date and time long)	12 bytes	Minimum: DTL#1970-01-01-00:00:00.0 Maximum: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

<sup>1</sup> The DTL data type is a structure of 12 bytes that saves information on date and time in a predefined structure. You can define a DTL in either the Temp memory of the block or in a DB.

### 1.1.5 Memory areas of the S7-1200

**Note**

**Using symbolic addresses instead of absolute addresses**

STEP 7 Basic facilitates symbolic programming. You create symbolic names or "tags" for the addresses of the data, whether as PLC tags relating to memory addresses and I/O points or as local variables used within a code block. To use these tags in your user program, simply enter the tag name for the instruction parameter. For a better understanding of how the CPU structures and addresses the memory areas, the following paragraphs explain the "absolute" addressing that is referenced by the PLC tags.

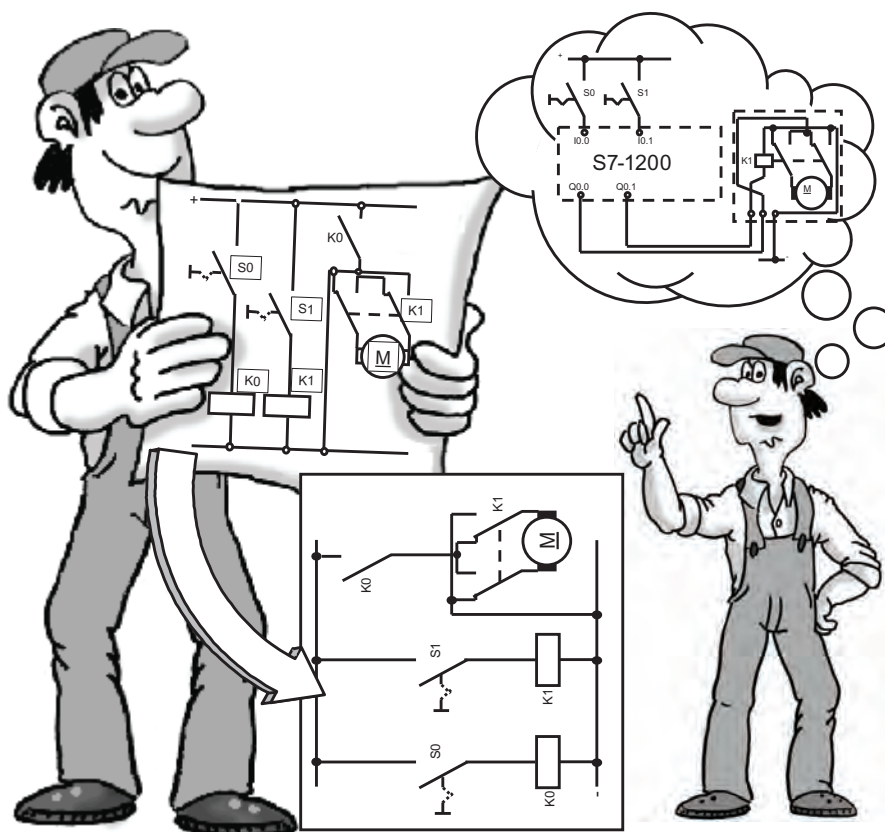
The CPU provides a variety of specialized memory areas, including inputs (I), outputs (Q), bit memory (M), data block (DB), and local or temporary memory (L). Your user program accesses (reads from and writes to) the data stored in these memory areas. Each different memory location has a unique address. Your user program uses these addresses to access the information in the memory location.

Memory area	Description
I Process image input	The CPU copies the state of the physical inputs to I memory at the beginning of the scan cycle. To immediately access or to force the physical inputs, append a ":P" to the address or tag (such as "Start:P" or I0.3:P).
Q Process image output	The CPU copies the state of Q memory to the physical outputs at the beginning of the scan cycle. To immediately access or to force the physical outputs, append a ":P" to the address or tag (such as "Stop:P" or Q0.3:P).
M Bit memory	The user program reads and writes the data stored in M memory. Any code block can access the M memory. You can configure addresses within M memory to retain the values of the data after a power cycle.
L "Temp" memory	Whenever a code block is called, the CPU allocates the temporary, or local, memory (L) to be used during the execution of the block. When the execution of the code block finishes, the CPU reallocates the local memory for the execution of other code blocks.
DB Data block	Use the DB memory for storing various types of data, including intermediate status of an operation or other control information parameters for FBs, and data structures required for many instructions such as timers and counters. You can specify a data block to be either read/write or read only. You can access data block memory in bits, bytes, words, or double words. Both read and write access is permitted for read/write data blocks. Only read access is permitted for read-only data blocks.

Whether you use a tag (such as "Start" or "Stop") or an absolute address (such as "I0.3" or "Q1.7"), a reference to the input (I) or output (Q) memory areas accesses the process image and not the physical output. To immediately access or force the physical (peripheral) input or output in your user program, append the reference with ":P" (such as "Stop:P" or "Q0.3:P").

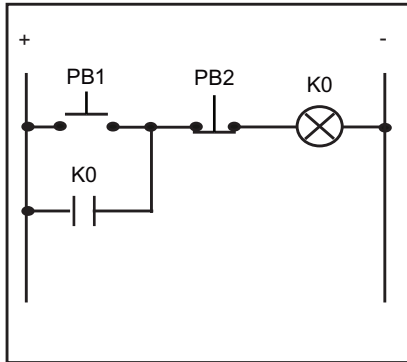
## 1.2 From schematic diagram to user program

To transform a circuit diagram into a PLC program, simply rotate your circuit diagram 90° to the left. Your power rail then appears on the left, with the grounding rail on the right. The switching elements of your circuit appear in the middle.



The circuit logic of a machine (such as time relays or flip-flops) which used to be implemented by wiring together switches, auxiliary contactors or control contactors is now handled by the PLC. The control elements on the input side (such as input switches or selector switches) and the power contactors on the output side (such as motor contactors, polarity reversers, or valves) cannot be replaced by the PLC.

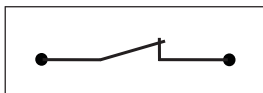
### 1.2.1 Converting the schematic to LAD instructions



Let's have a closer look at the structure of the PLC program in a ladder diagram (LAD).

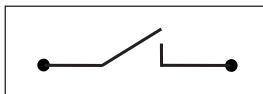
This type of representation closely resembles a circuit diagram.

Digital control logic uses only 0 and 1 for the data states. The "0" state equates to a "false" condition, and the "1" state equates to a "true" condition. In an electrical circuit, current flow determines the binary state: Current flow equals 1, and no current flow equals 0. At the beginning of every scan, the CPU stores the states of the inputs as either 0 (false) or 1 (true).



A closed switch allows current flow. In this case, the state of the circuit equals "1".

"1" = "True" = Current flow



An open switch does not allow current flow. In this case, the state of the circuit equals "0".

"0" = "False" = No current flow

Specific LAD instructions correspond to basic electrical circuits of the schematic diagram.



**Positive logic** for the input sensor scan ("normally open contact"):

- 24V = high-level = "1"
- 0V = low-level = "0"



Is current flowing? If yes, then the result of the scan is "true" (or "1").



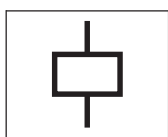
**Negative logic** for the input sensor scan ("normally closed contact"):

- 0V = low-level = "1"
- 24V = high-level = "0"

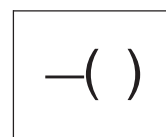


Is current flowing? When there is **no** current flowing through the physical contact, then the result of the scan is "true" (or "1"). Power flow then passes through the LAD contact.

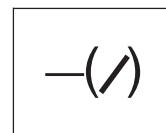
Remember that there is **no** current when the contact is closed because this is the "normal" condition. When the contact is actuated, it then opens and signals that the contact is active



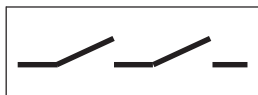
**Output coil:** If the value "true" (current or "power flow") is passed to the coil, the CPU activates the coil by turning the coil on (set to "1").



You can insert an output coil in the middle of a network, not only at the end. In addition, the CPU supports a negated coil that reflects the inverse of the power flow to that coil.



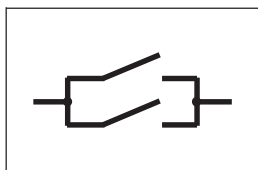
You combine electrical circuits to perform logic operations. For example, a serial connection of two contacts creates an AND operation for two inputs.



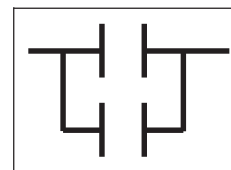
The first switch AND the second switch must be closed in order to pass current. You represent the AND operation in LAD by combining two instructions in sequence.



For OR logic, you use a parallel connection of two contacts.



Either the first switch OR the second switch must be closed in order to pass current. To create a parallel circuit connection (OR logic), you create a branch in the LAD network.

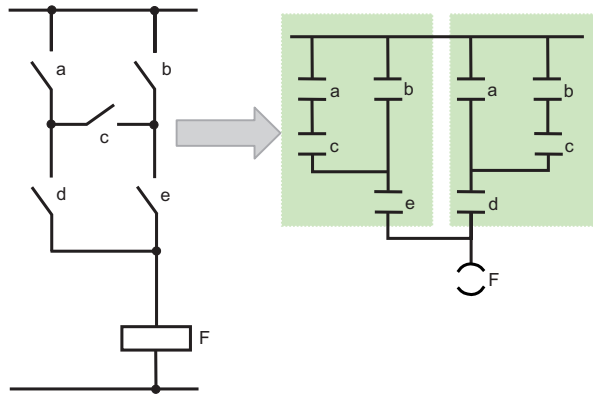
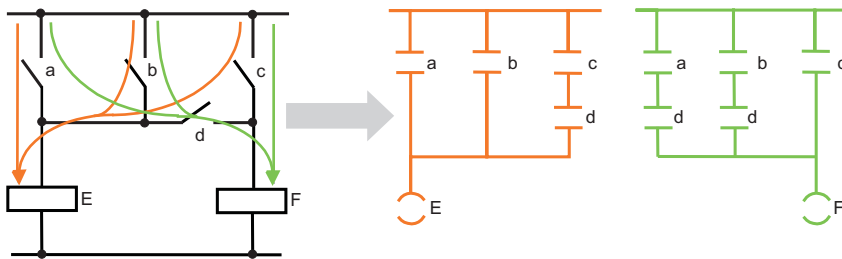


### 1.2.2 Basic electrical circuits



If you are changing over from contactor technology to PLC technology you will probably encounter switch combinations that cannot be converted directly into ladder diagram representation. (To aid in the comparison, the LAD networks have been arranged vertically.)

**Bridge circuits:** The following figure shows a simple bridge circuit (on the left). You use two LAD networks to implement the simple bridge circuit, as shown on the right. Each network handles one of the two possible current paths of the bridge circuit.



As shown here, you can also have a complex bridge circuit.

The first network corresponds to the "a-c-parallel-b" path. The second network corresponds to the "b-c-parallel-a" path.

**Note**

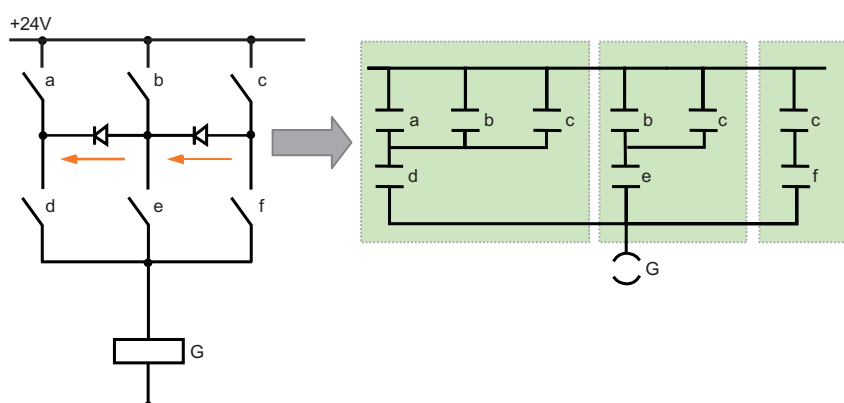
In new projects, avoid using the bridge circuit in the circuit diagram where possible! Think "in LAD" right from the start!





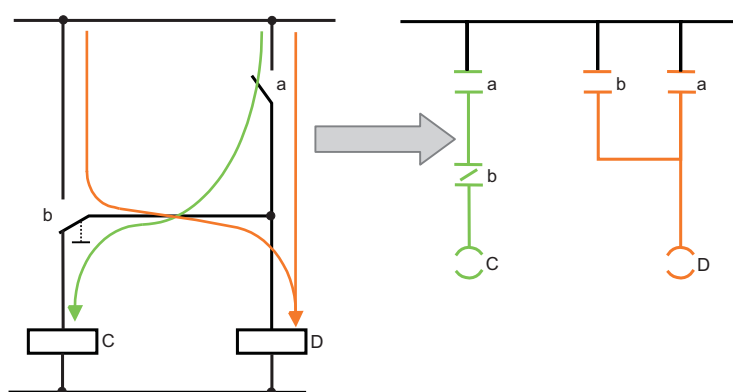
**Diode circuits:** When diodes have been used in "old" circuit diagrams, converting them into ladder diagram terms is not an altogether simple matter. Since diodes represent connection lines in principle but only conduct current in one direction, a similar solution is adopted here as with the bridge circuit. For ease of comparison with the circuit diagram, the ladder diagram is arranged vertically again.

For example, the following circuit provides three current paths: over switch "d", over switch "e", or over switch "f".



The current running through the diodes can flow only from "b" to "d" or from "c" to "e". When representing this circuit in LAD, the three current paths result in three framed sub-networks. With switches "d", "e" and "f" on the same rail as output "G", these three sub-networks are also linked to form one network.

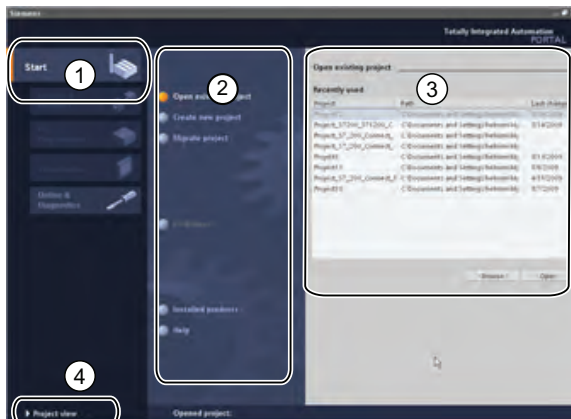
**Change-over switches:** Changeover switches should likewise not cause you any problem when you convert a circuit diagram into LAD. In principle, you can use a normally closed contact and a normally open contact with the same input address to convert a changeover switch. (For the following example, the current paths are highlighted.)



Changeover switch b is divided into a normally closed contact that is switched in series and contributes to the effect at output "C", or a normally open contact that takes effect in parallel with "a" and switches "D".

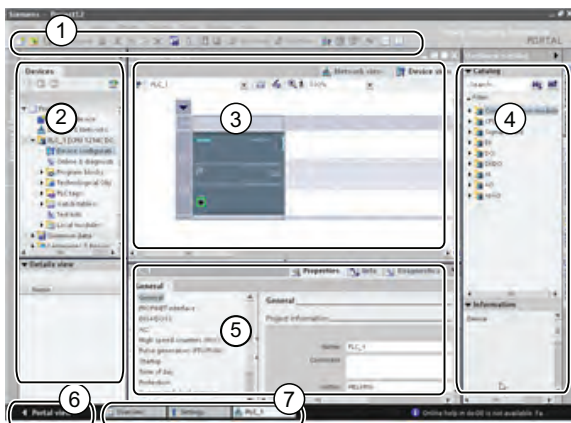
### 1.3 Basic features of STEP 7 Basic

STEP 7 Basic provides a user-friendly environment to develop controller logic, configure HMI visualization, and setup network communication. To help increase your productivity, STEP 7 Basic provides two different views of the project: a task-oriented set of portals that are organized on the functionality of the tools (Portal view), or a project-oriented view of the elements within the project (Project view). Choose which view helps you work most efficiently. With a single click, you can toggle between the Portal view and the Project view.



The Portal view provides a functional view of the project tasks and organizes the tools according to the tasks to be accomplished. You can easily determine how to proceed and which task to choose.

- ① Portals for the different tasks
- ② Tasks for the selected portal
- ③ Selection panel for the selected action
- ④ Changes to the Project view



The Project view provides access to all of the components within a project.

- ① Menus and toolbar
- ② Project navigator
- ③ Work area
- ④ Task cards
- ⑤ Inspector window
- ⑥ Changes to the Portal view
- ⑦ Editor bar

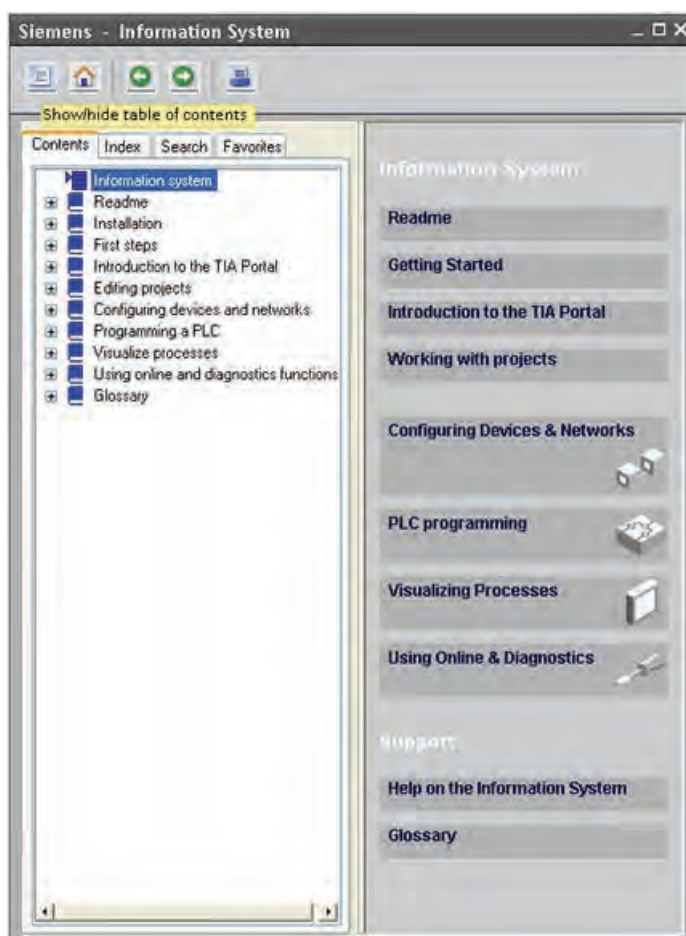
With all of these components in one place, you have easy access to every aspect of your project. For example, the inspector window shows the properties and information for the object that you have selected in the work area. As you select different objects, the inspector window displays the properties that you can configure. The inspector window includes tabs that allow you to see diagnostic information and other messages.

By showing all of the editors that are open, the editor bar helps you work more quickly and efficiently. To toggle between the open editors, simply click the different editor. You can also arrange two editors to appear together, arranged either vertically or horizontally. This feature allows you to drag and drop between editors.

### 1.3.1 Help when you need it

To help you to find more information or to resolve issues quickly and efficiently, STEP 7 Basic provides intelligent point-of-need assistance. Hovering over an element of the software interface displays the tool tip. Some of the tool tips in the interface "cascade" to provide additional information and even include a link to a specific topic in the online information system. A black triangle alongside the tool tip signifies that more information is available.

Help is always just a click away! From the Portal view, select the Start portal and click the "Help" command. From the Project view, select the "Show help" command in the "Help" menu.



The information system opens in a window that does not obscure the work areas.

To undock the help window and display the contents, click the "Show/Hide contents" button. You can then resize the help window.

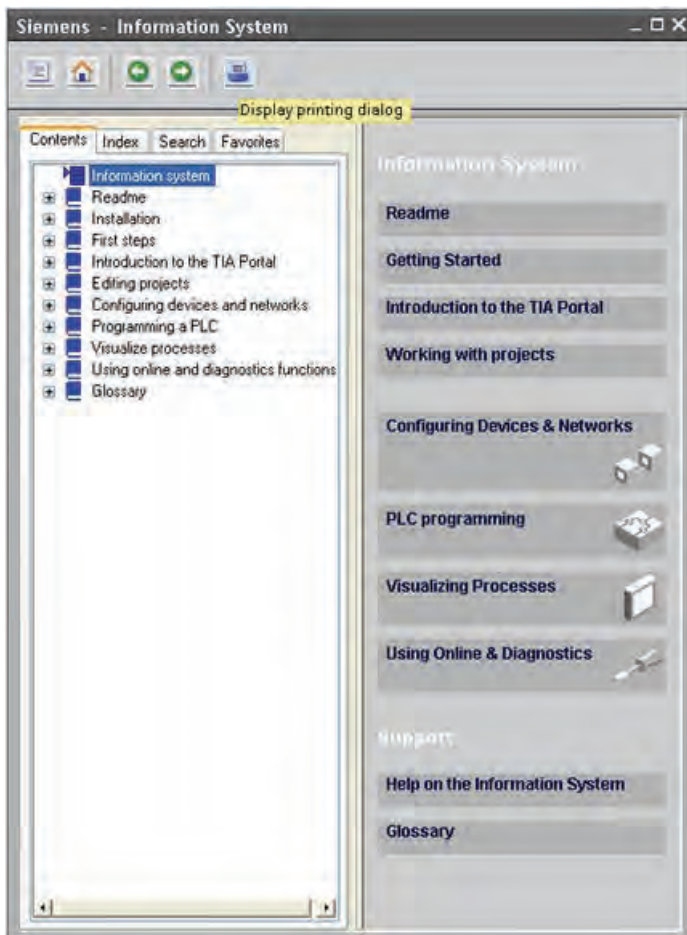
If STEP 7 Basic is maximized, clicking the "Show/Hide contents" button does not undock the help window.

Click the "Restore down" button to undock the help window.

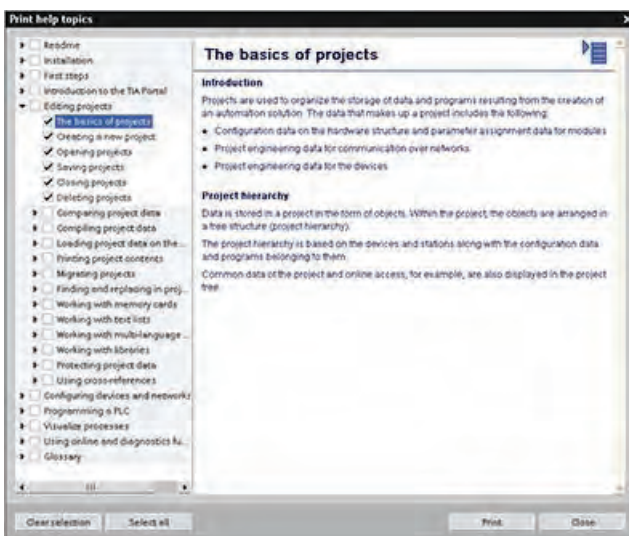


You can then move and resize the help window.

### 1.3.2 Printing from the online help



To print from the information system, click the "Print" button on the help window.

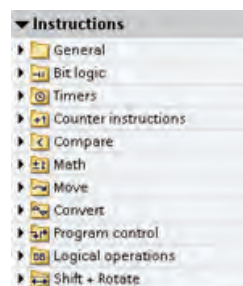


The "Print" dialog allows you to select the topics to print. Make certain that the panel displays a topic. You can then select any other topic to print. Click the "Print" button to send the selected topics to your printer.

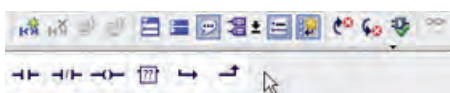
### 1.3.3 Inserting instructions

#### Dragging and dropping instructions from the instruction tree

STEP 7 Basic provides task cards that contain the instructions for your program. The instructions are grouped according to function. To create your program, you drag instructions from the task card onto a network.



#### Using the "Favorites" tool bar for inserting commonly used instructions



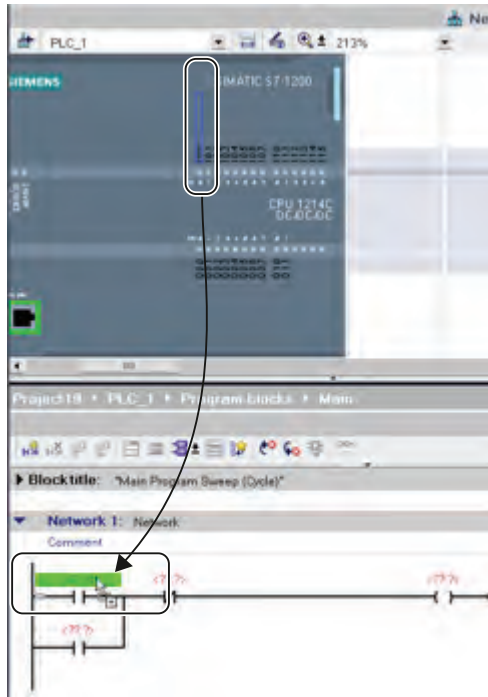
STEP 7 Basic provides a "Favorites" toolbar to give you quick access to the instructions that you frequently use.

STEP 7 Basic displays the "Favorites" in both the LAD editor and the "Instructions" task card. Simply click the icon for the instruction to insert it into your network! (Or double-click, if you use the "Favorites" located in the "Instructions" task card.)

You can easily customize the "Favorites" by adding a new instruction. Simply drag and drop an instruction to the "Favorites". The instruction is now just a click away!



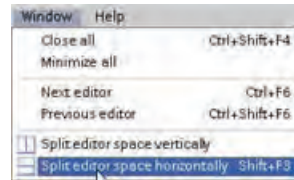
### 1.3.4 Dragging and dropping between editors



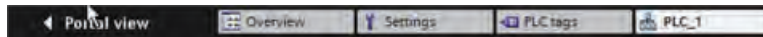
To help you perform tasks quickly and easily, STEP 7 Basic allows you to drag and drop elements from one editor to another. For example, you can drag an input from the CPU to the address of an instruction in your user program. (You must zoom in at least 200% to select the I/O of the CPU.)

Notice that the tag names are displayed not only in the PLC tag table, but also are displayed on the CPU.

To display two editors at one time, use the "Split editor" menu commands or buttons in the toolbar.



To toggle between the editors that have been opened, click the icons in the editor bar.



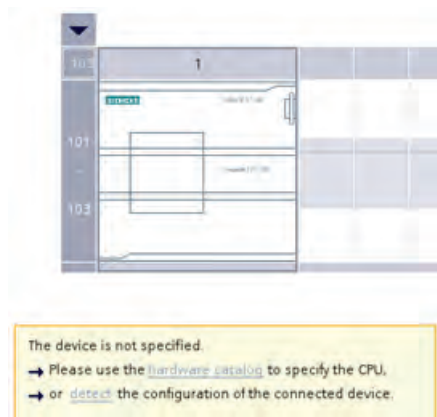
### 1.3.5 Detecting the configuration of a CPU



If you have a physical CPU that you can connect to the programming device, it is easy to upload the configuration of the hardware. You must first connect the CPU to your programming device, and you must create a new project. After creating the unspecified CPU, you can upload the hardware configuration for the online CPU.

From the program editor, select the "Hardware detection" command from the "Online" menu.

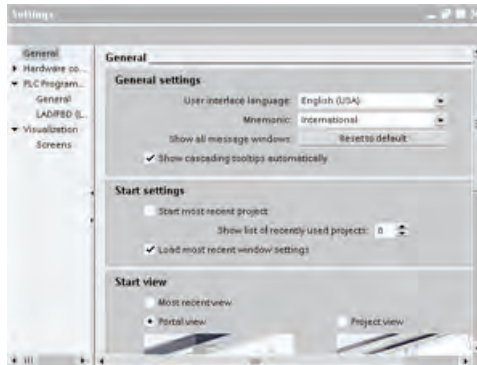
From the device configuration editor, select the option for detecting the configuration of the connected device.



#### Note

After you select the CPU from the online dialog, STEP 7 Basic uploads the configuration from the CPU, including any modules (SM, SB, or CM). If the CPU has already been assigned an IP address, the IP address is uploaded. However, other settings (such as the properties for the analog I/O) are **not** uploaded. You must go to "Device configuration" to manually configure the settings for the CPU and modules.

### 1.3.6 Changing the settings for the work area



You can select a variety of settings, such as the appearance of the interface, language, or the directory for saving your work.

Select the "Settings" command from the "Options" menu to change these settings.



# Installation

## 2.1 Read this first for safety information

This *Getting Started with S7-1200* provides a quick introduction to the world of S7-1200, but it is not a substitute for the *S7-1200 Programmable Controller System Manual*. Before installing your CPU 1212C AC/DC/RLY, please review the safety guidelines and other information described in the *S7-1200 Programmable Controller System Manual*.

---

### Note

The information in this manual describes the CPU 1212C AC/DC/RLY delivered with the S7-1200 Starter Kit. For information about any other S7-1200 CPU, refer to the *S7-1200 Programmable Controller System Manual*.

---

As a general guideline before installing or removing any electrical device, ensure that the power to that equipment has been turned off. Also, ensure that the power to any related equipment has been turned off.

 <b>WARNING</b>
--

<b>Installation or removal of S7-1200 or related equipment with the power applied could cause electric shock or unexpected operation of equipment.</b>
--

Failure to disable all power to the S7-1200 and related equipment during installation or removal procedures could result in death, severe personal injury and/or property damage due to electric shock or unexpected equipment operation.
---

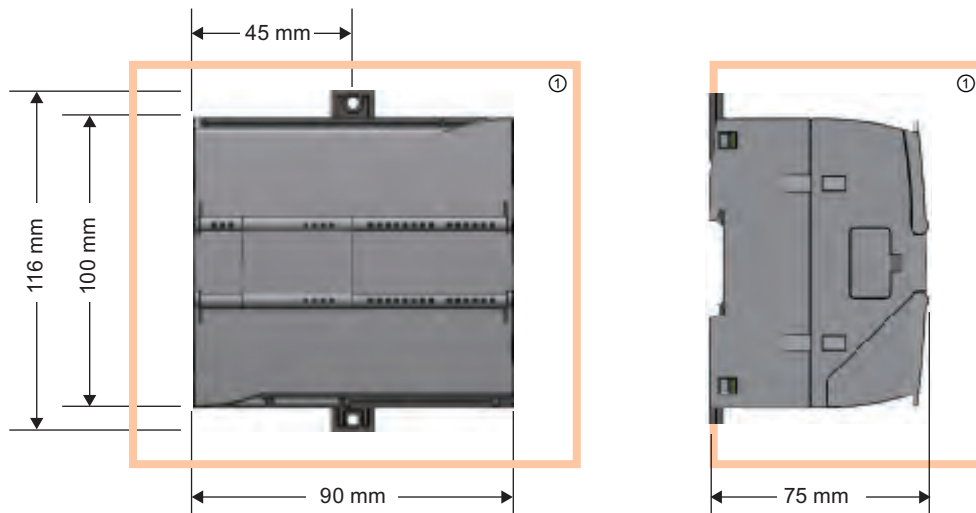
Always follow appropriate safety precautions and ensure that power to the S7-1200 is disabled before attempting to install or remove S7-1200 CPUs or related equipment.
---

 <b>WARNING</b>
--

<b>Do not use the input simulators in a hazardous location (Class I DIV 2 or Class I Zone 2).</b>
---

The input simulators are not approved for use in Class I DIV 2 or Class I Zone 2 hazardous locations. The switches present a potential spark hazard/explosion hazard if used in a Class I DIV 2 or Class I Zone 2 location.
---

## 2.2 Dimensions and guidelines



① Thermal clearance: 25 mm



Always consider the following guidelines when planning your installation:

- Separate the devices from heat, high voltage, and electrical noise.
- Provide adequate clearance for cooling and wiring. A 25 mm thermal zone must be provided above and below the unit for free air circulation.

Refer to the *S7-1200 Programmable Controller System Manual* for specific requirements and guidelines for installation.

**⚠ WARNING**

**Installation or removal of S7-1200 or related equipment with the power applied could cause electric shock or unexpected operation of equipment.**

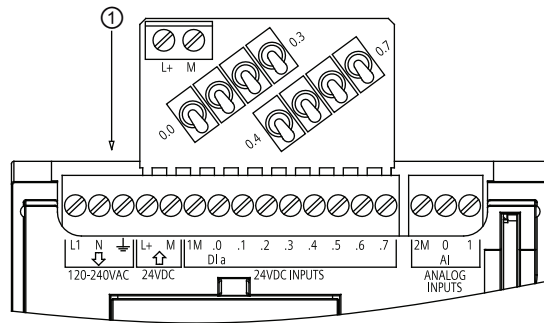
Failure to disable all power to the S7-1200 and related equipment during installation or removal procedures could result in death, severe personal injury and/or property damage due to electric shock or unexpected equipment operation.

Always follow appropriate safety precautions and ensure that power to the S7-1200 is disabled before attempting to install or remove S7-1200 CPUs or related equipment.

## 2.3 Connecting the hardware

### Insert the input simulator

The simulator connects to the 24 VDC and the input terminals. Ensure that all of the switches on the simulator are set to the "off" position.

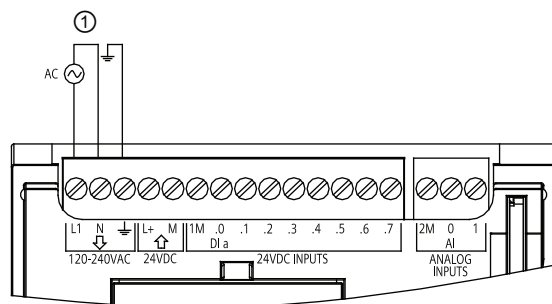


- ① Insert the simulator.
1. Loosen the screws of the terminal.
2. Insert the simulator.
3. Tighten all of the screws on the terminal.

### Connect the power to the CPU

**CAUTION**

The S7-1200 Starter Kit provides a CPU 1212C AC/DC/RLY, which uses 120/240 VAC. To use a different CPU, refer to the *S7-1200 Programmable Controller System Manual* for instructions on how to connect the power to your specific CPU.



- ① Connect power to the CPU.
- The CPU requires 120 to 240 VAC. Insert the power lines of the cord into the L1 and N terminals, and insert the ground line in the ground terminal. After tightening the terminal screws, you can plug the cord into the outlet.

### Connect the PROFINET cable

The PROFINET cable is a standard CAT5 Ethernet cable with RJ45 connectors that connects the CPU with your computer or programming device.



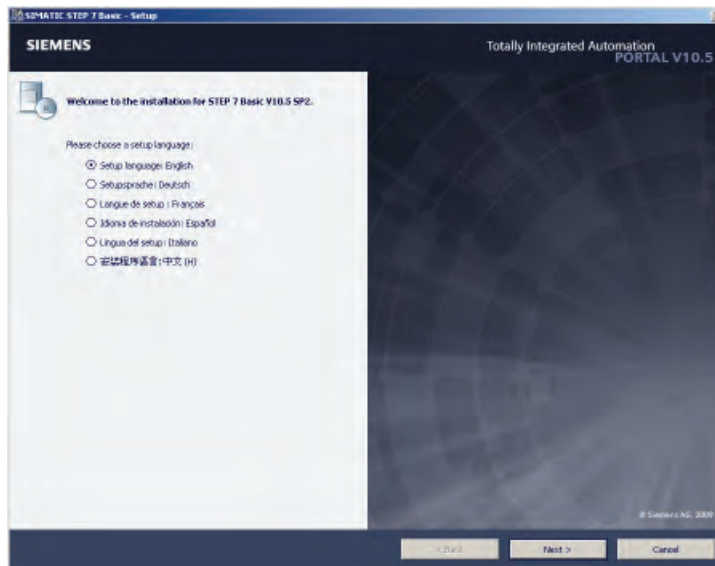
① Insert one end of the PROFINET cable to the CPU.

Insert the other end of the cable into the Ethernet port of your computer or programming device.

## 2.4 Installing STEP 7 Basic

The installation disk for STEP 7 Basic contains the following SIMATIC software:

- STEP 7 Basic for your S7-1200 CPU
- WinCC Flexible Basic for the SIMATIC HMI Basic Panels
- License Manager for authorizing your SIMATIC software products

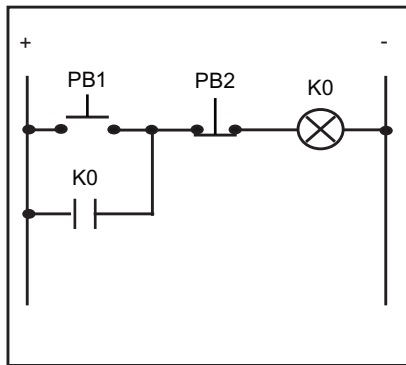


Insert the disk into the drive of your computer. The installation program should start automatically. (If the program does not start automatically, refer to the instructions for the installation disk.)

Choose the language for the installation dialogs, and then simply follow the directions.

You will be given the option for installing the software for multiple languages. Typically, install only the language that you will be using. You can, however, install any or all of the other languages. Be aware that installing more than one language requires more disk space.

## Create a simple latch circuit



You should be familiar with the standard latching function:

- The output K0 is activated (ON) as soon as the push-button input PB1 closes (ON).
- Because the latching circuit utilizes the state of K0, K0 remains active (ON) after PB1 releases (OFF).
- Pressing the push-button input PB2 deactivates K0 (OFF).
- K0 remains OFF until push-button input PB1 closes (ON) again.

Using K0 as a parallel input to PB1 ensures that the circuit will be "latched" on until K0 turns off.

Our first exercise shows how to create a latching circuit in LAD by performing the following tasks:

- Creating a project
- Inserting LAD instructions to create a small user program
- Associating the LAD instructions with the I/O of the CPU
- Configuring the CPU
- Downloading your user program to the CPU
- Testing the operation of the user program

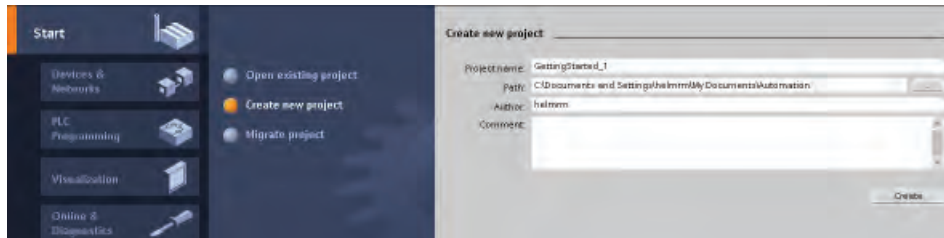


### 3.1 Create a project for your user program



Double-click the desktop icon to start STEP 7 Basic.

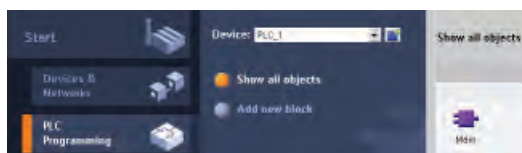
After STEP 7 Basic opens, click "Create new project" in the Start portal. Enter the project name and click "Create".



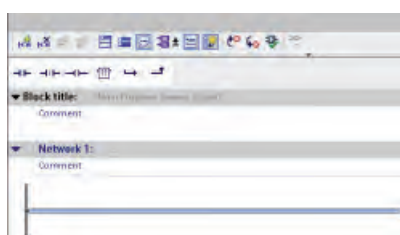
STEP 7 Basic displays the "First steps".

Create a new user program by clicking "Create a PLC program".

STEP 7 Basic creates the "Main" code block for your user program and opens the "PLC Programming" portal.



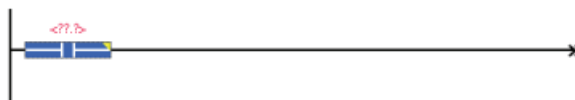
Open the program editor by double-clicking the "Main" block



STEP 7 Basic changes to the Project view and displays the network of the Main block.

### 3.2 Create a simple network in your user program

To enable the latching circuit, use a normally open contact. The normally open contact provides power flow (current) when the switch is turned on. Click the normally open contact in the "Favorites" to insert the contact onto the network.



To disable the latching circuit, use a normally closed contact. The normally closed contact provides power flow (current) until the switch is turned on. Turning on a normally closed contact interrupts power flow.



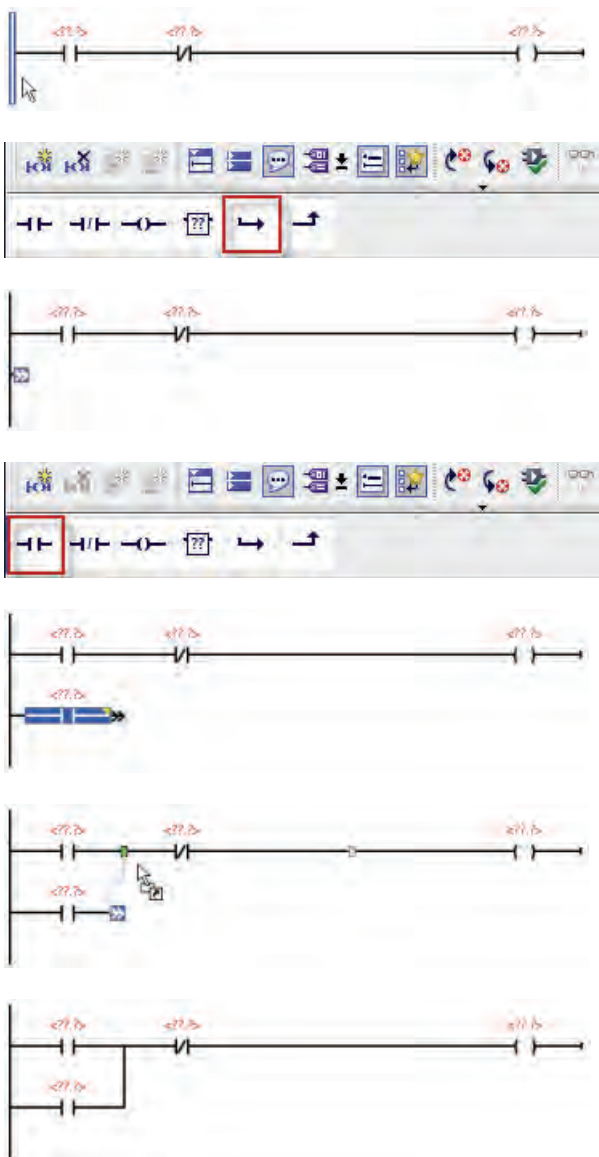
Click the normally closed contact in the "Favorites" to insert the contact onto the network.

Power flows through the two contacts to energize the coil.



Click the coil to insert a coil onto the network.

3.2 Create a simple network in your user program



To "latch" the coil on after the "On" switch releases, you create a parallel branch.

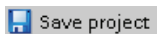
1. Select the rail of the network.
2. Click the "Open branch" in the "Favorites" to open a branch from the rail.

Click the normally open contact in the "Favorites" to insert the contact onto the branch.

Close the branch by dragging the end to the network.

Connecting the branch between the two contacts on the network ensures the following conditions:

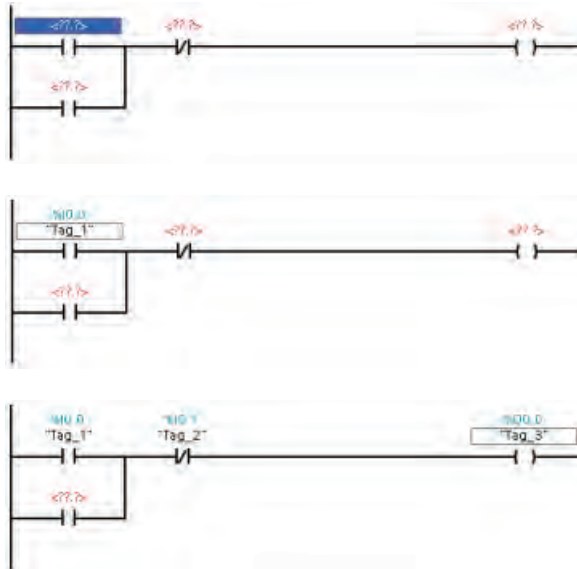
- The power to the coil can flow to the coil after the first switch releases (turns off).
- The normally closed contact can break the circuit and turn the coil off.



Click the "Save project" button on the toolbar to save your work. You can now create "tags" that associate the instructions of the user program with the inputs and outputs of the user program.



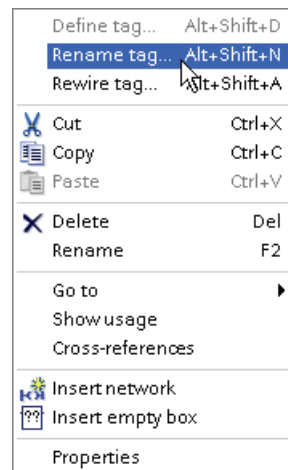
### 3.3 Enter the tags and addresses for the instructions



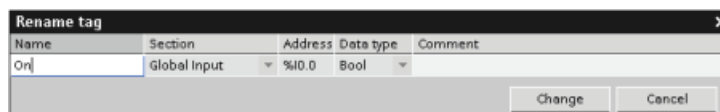
The next step is to associate the contacts and coils to the inputs and outputs of the CPU. You create "PLC tags" for these addresses.

1. Select the first contact and double-click the operand ("**<???.?>**").
2. Enter the address "I0.0" to create a default tag for this input.
3. Enter the address "I0.1" for the normally closed contact.
4. Enter an address of an output ("**Q0.0**") for the coil.

You can easily rename the default tag name created by STEP 7 Basic.



Simply right-click the instruction (contact or coil) and select the "Rename tag" command from the context menu.

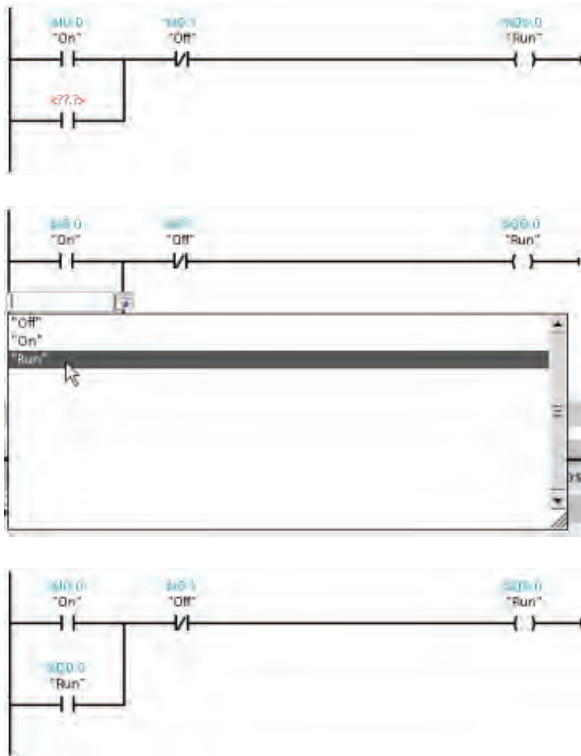


Enter the following names for the three instructions:

- Change "Tag\_1" (I0.0) to "On".
- Change "Tag\_2" (I0.1) to "Off".
- Change "Tag\_3" (Q0.0) to "Run".

Create a simple latch circuit

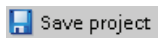
3.3 Enter the tags and addresses for the instructions



STEP 7 Basic saves the tags in a tag table. You can enter the tag address directly from the tag table to the instruction.

1. Select the contact on the branch.
2. Either click on the icon next to the field or enter an "r" or an "o" to display the entries in the tag table.
3. Select "Run" from the list of tags in the tag table.

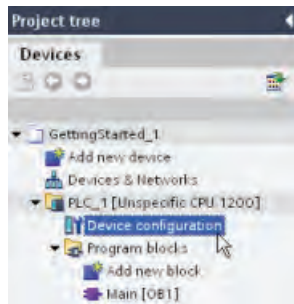
The latching circuit is now complete.



Click the "Save project" button on the toolbar to save your work. Your program is now ready to be downloaded to the CPU. Before you can download the user program, however, you must configure a CPU in the device configuration for the project.

### 3.4 Configure the CPU

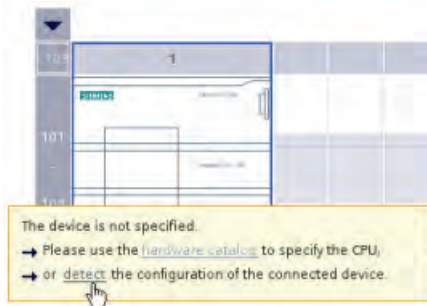
#### Uploading the configuration of the CPU



You can use the "Detect CPU" feature of the Device Configuration to upload the hardware configuration of the CPU.

1. In the Project tree, expand the "PLC" container.
  2. Double-click "Device configuration" to display the CPU.
- STEP 7 Basic had created an "unspecified" CPU when you opened the LAD editor.

You can now click the "Detect" link on the unspecified CPU to connect to the online CPU.



STEP 7 Basic "detects" any CPU connected to the computer. Select the CPU and click the "Load" button to upload the CPU configuration to your project.



STEP 7 Basic displays the CPU in the device configuration.

### Configuring the properties for the CPU

You use the properties to configure the operating parameters for the CPU. For example, the default setting for startup after a power cycle is to have the CPU go to STOP mode.

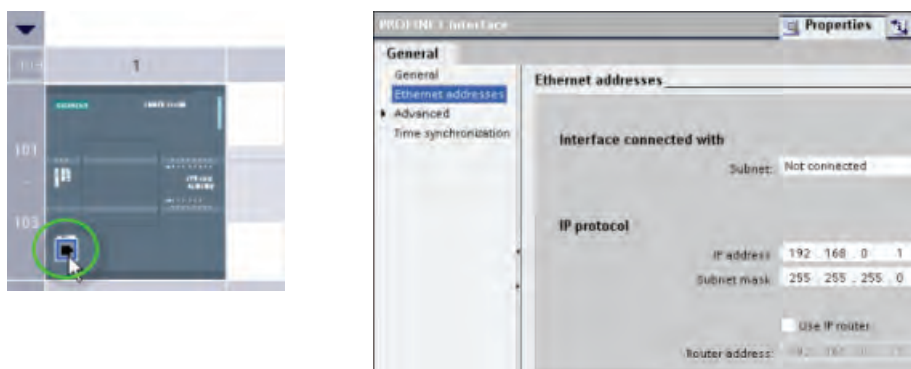


For this exercise, change the startup property:

1. Select the CPU to display the properties in the inspector window.
2. In the inspector window, select the "Properties" tab and select "Startup" to display the options for the startup mode of the CPU.
3. Select "Warm restart - RUN".

You have configured the CPU to always perform a warm restart and go to RUN mode following a power cycle.

As you noticed when you uploaded the CPU configuration, the CPU does not have a pre-assigned IP address. You assign the IP address for each CPU.



1. Select the PROFINET port on the CPU to display properties for just the PROFINET interface. (You can also select "PROFINET interface" in the "General" properties of the CPU.)
2. Select "Ethernet addresses" in the inspector window.

The "IP protocol" section displays the default IP address created by STEP 7 Basic.

This example uses the default IP address. Based on the needs of your application and installation, you might need to configure a specific network address for the CPU. Refer to the system manual for more information, and consult your local network specialist.

## Downloading the configuration to the CPU

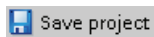
You can now do download the configuration:

-  1. Select the CPU.
- 2. Click the "Download" button on the toolbar.

After connecting to the CPU, STEP 7 Basic displays the "Load preview" dialog. Click "Load" to download the device configuration to the CPU. After the download is complete, STEP 7 Basic displays the "Load results" dialog. Click "Finish".

The CPU is now configured to use the default IP address and to go to RUN mode following a power cycle. You can now download the user program.

## Saving your work in the project

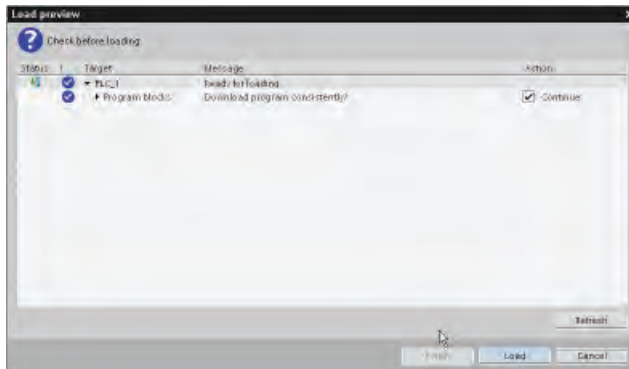


You can now save your work. Simply click the "Save project" button on the toolbar.

## 3.5 Download the user program to the CPU



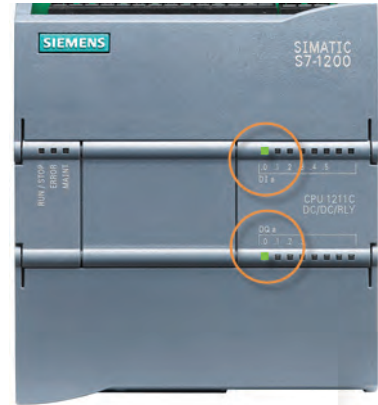
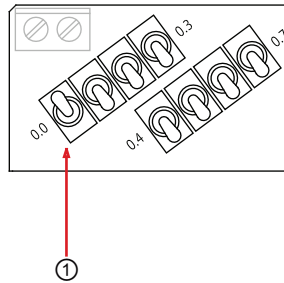
Now you can download your user program. Open the program editor and simply click the "Download" button.



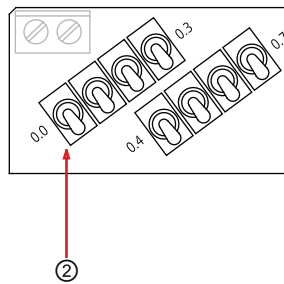
After connecting to the CPU, STEP 7 Basic displays the "Load preview" dialog. Click "Load" to download the user program to the CPU. Before clicking "Finish", select "Start all" to ensure that the CPU goes to RUN mode.

### 3.6 Test the operation of your sample user program

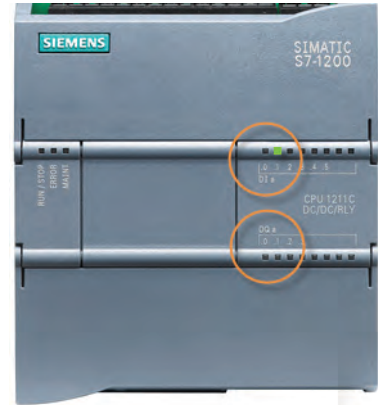
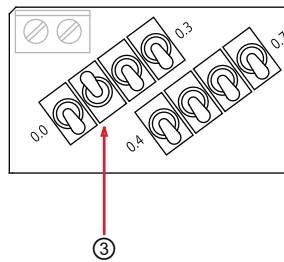
- ① Turn the "On" switch (I0.0) on.  
The status LEDs for "Start" (I0.0) and "Run" (Q0.0) turn on.



- ② Turn the "On" switch (I0.0) off.  
The status LED for "On" (I0.0) turns off, but the status LED for "Run" (Q0.0) stays on.



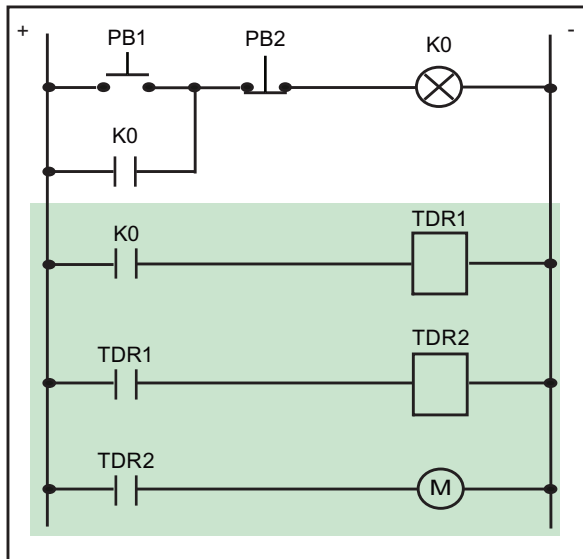
- ③ Turn the "Off" switch (I0.1) on.  
The status LED for "Off" (I0.1) turns on, and the status LED for "Run" (Q0.0) turns off.



## 3.7 Reviewing the tasks for the first exercise

Congratulations! You have transformed a simple electrical circuit into LAD instructions by performing the following tasks.

- Using the current path to create a logical flow for the instructions
- Inserting contacts and coils to create a latching circuit
- Creating tags to link the instructions to the inputs and outputs of the circuit
- Uploading the configuration of the CPU to your project
- Downloading and testing your user program



In a following exercise, you will add other networks for the simple user program.



*Create a simple latch circuit*

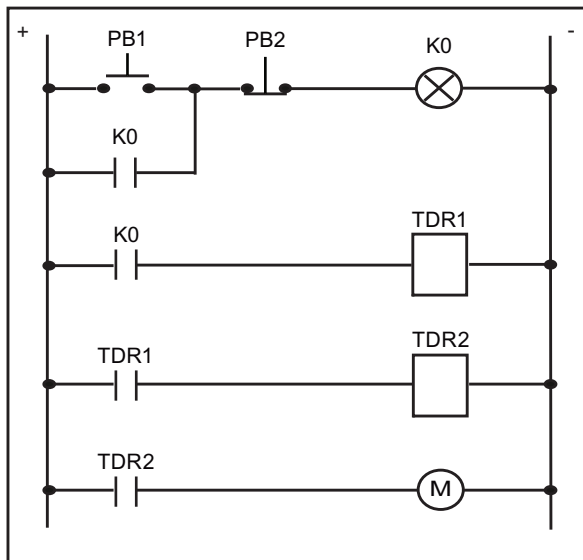
*3.7 Reviewing the tasks for the first exercise*

---



# 4

## Complete the user program



In the next exercise, you will complete the user program to include the rest of the circuit by inserting an on-delay timer and an off-delay timer.



---

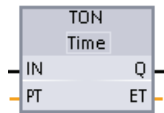
### Note

S7-1200 does not limit the number of timers or counters in your user program. The data for each timer is stored in a data block, so you are limited only by the size of your user program in relation to the amount of load memory for your CPU.

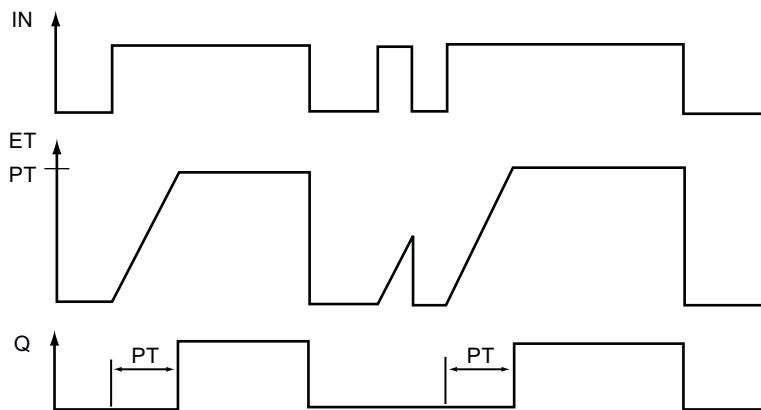
---

### 4.1 Add an on-delay timer to delay the start of an output

In this exercise, you will expand on the latch circuit by inserting an on-delay timer to turn on an output after a preset amount of time.



You use the TON instruction to turn on an output (Q) after a preset delay. The timer uses a structure stored in a data block to maintain the timer data. You assign this data block when the timer instruction is placed in the editor.



When the timer turns on (IN = 1), the on-delay timer waits a preset amount of timer (PT) until turning its output on (Q = 1). The output stays on (Q = 1) as long as the input stays on (IN = 1).

The on-delay timer uses a data block (DB) to maintain the timer data. You assign this data block when you insert the TON instruction onto the network.

For this exercise, you will enter a constant for the for preset time parameter (PT).

For your own application, you could also enter a Time (4-byte signed value) memory address that stores the preset value. This would allow your user program to change the preset value according to operating conditions, if required. You can also store the elapsed time (ET) in a Time (4-byte signed value) memory address that can be accessed by the other elements of your user program.

## Insert an on-delay timer

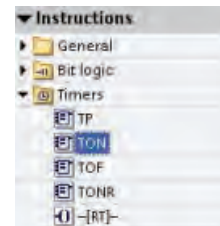
The second network of your user program will use a TON instruction that turns on 5 seconds after the "Run" output of the latching circuit turns on.



First, enter the contact that will activate the timer.

1. Select the second network of your user program.
2. Just like you did for the latch circuit (Page 31), click the normally open contact in the "Favorites" to insert the instruction.
3. For the address of the instruction, select the "Run" tag. (As for the previous exercise (Page 33), you can type an "r" or else click the tag icon to display the list of tags.)

Expand the "Timers" folder in the Instruction task card, and then drag the TON timer to the network.



Dropping the TON instruction onto the network automatically creates a single-instance data block (DB) for storing the timer data.

Click "OK" to create the DB.

Now create a 5-second delay.



1. Double-click the preset time (PT) parameter.
2. Enter a constant value of "5000" (for 5000 ms, or 5 seconds).



You can also enter "5s" for 5 seconds. ("5h" enters 5 hours, and "5m" enters 5 minute.)

STEP 7 Basic formats the constant as "T#5000ms"

## Complete the user program

### 4.1 Add an on-delay timer to delay the start of an output

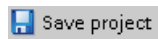
Now insert a coil that will turn on after 5 seconds (the preset value for the TON instruction). For this exercise, enter "M0.0" for the address. This stores the value in the bit memory (M) area. Rename the tag to "Delay\_5sec".



You have created an on-delay timer that turns on the "Delay\_5sec" bit after 5 seconds.



## Saving your work in the project



You can now save your work. Simply click the "Save project" button on the toolbar.

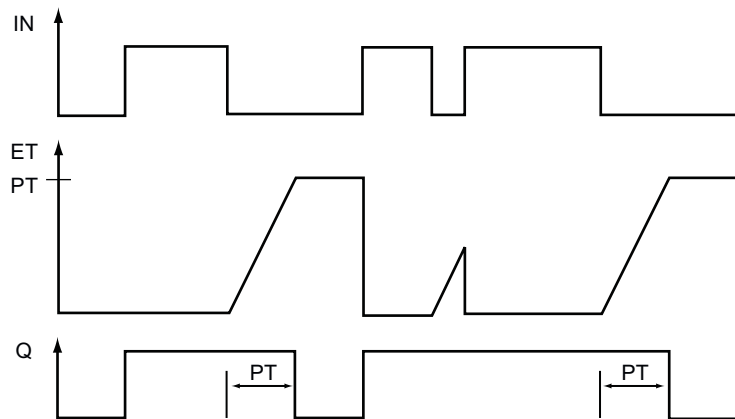
## 4.2 Turn on an output for 5 seconds

Next, you add an off-delay timer (TOF) to the third network of your user program.



You use a TOF instruction to hold an output (Q) on for a preset amount of time after the input (IN) turns off. When enabled by the rising edge of IN, the TOF instruction turns Q on. The TOF timer starts on the falling edge of IN. After the preset time (PT) elapses, the TOF instruction turns Q off.

When the timer turns on (IN = 1), the output of the off-delay timer turns on (Q = 1). After a preset amount of time (PT), the output of the off-delay timer turns off.



Like the on-delay timer, the off-delay timer uses a data block (DB) to maintain the timer data. You assign this data block when you insert the TOF instruction onto the network.

As with the on-delay timer, you will enter a constant for the for preset time parameter (PT).

You could also enter a Time (4-byte signed value) memory address that stores the preset value. This would allow your user program to change the preset value according to operating conditions, if required. You can also store the elapsed time (ET) as a Time (4-byte signed value) memory address that can be accessed by the other elements of your user program.

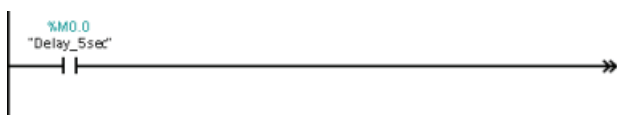
Use an off-delay timer to turn on an output for 5 seconds

The third network of your user program will use a TOF instruction that turns on and stays on for 5 seconds after the TON timer (Page 42) turns on.



For the off-delay timer, you use the "Delay\_5sec" tag to activate the timer.

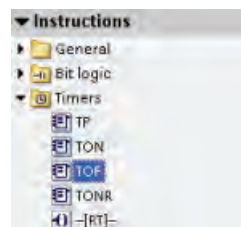
1. Select the third network of your user program.
2. Click the normally open contact in the "Favorites" to insert the instruction.
3. Select the "Delay\_5sec" tag. (You can either type a "d" or else click the tag icon to display the list of tags.)



The "Delay\_5sec" bit turns on 5 seconds after the "Run" output turns on.

To insert the TOF timer, double-click the TOF timer in the Instruction task card. Just like with the TON timer, STEP 7 Basic creates a DB for timer data. (Click "OK" to create the timer and insert the TOF instruction.)

For the preset value of the TOF instruction, enter 5000 (for 5000 ms, or 5 seconds) in the PT parameter.



Now create a coil to turn on when the TOF instruction is activated. Enter "M0.1" for the address. Rename the tag to "On\_for\_5sec".



When the TON timer turns "Delay\_5sec" on, the TOF timer turns the "On\_for\_5sec" bit on for 5 seconds.

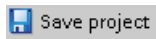
To complete the user program, create a network with two elements:

- Insert a normally open contact. For the address, select the "On\_for\_5sec" tag.
- Insert a coil. Enter "Q0.1" for the address. Rename the tag to "MotorStart".



When the TOF timer is activated, "On\_for\_5sec" turns the "MotorStart" output on for 5 seconds.

## Saving your work in the project

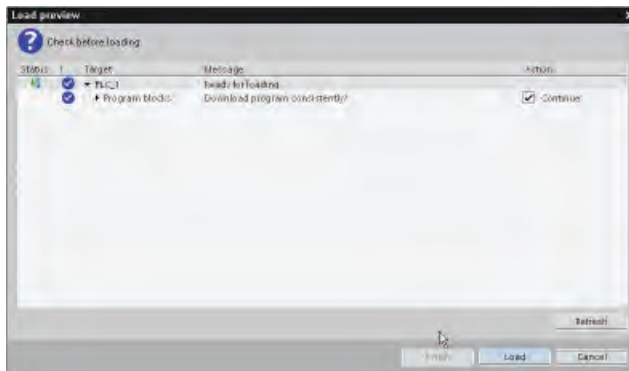


You can now save your work. Simply click the "Save project" button on the toolbar.

## Download and test the Timer instructions



Now you can download your user program. Open the program editor and simply click the "Download" button.



After connecting to the CPU, STEP 7 Basic displays the "Load preview" dialog. Click "Load" to download the user program to the CPU. Before clicking "Finish", select "Start all" to ensure that the CPU goes to RUN mode.

You are now ready to test your program:

1. Ensure that all of the inputs are off.
2. Turn input I0.0 on.

After 5 seconds, the status LED for output Q0.1 turns on.

3. Turn input I0.0 off and turn input I0.1 on.

After 5 seconds, the status LED for output Q0.1 turns off.

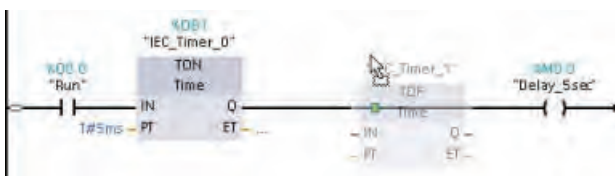
### 4.3 Editing your user program for a different structure



The networks that you created match the circuit diagram. This helps other people to understand the logic of the user program, such as for troubleshooting or for future modification.

However, you could also combine the timers and the "MotorStart" output onto one network. Our next exercise shows how easy it is to edit your LAD program.

First, drag the TOF timer from network 3 and drop it onto network 2.



The output of the TON timer now connects directly to the input of the TOF timer.



Tag list for %M0.0:

- "Delay\_5sec"
- "IEC\_Timer\_0" >>
- "IEC\_Timer\_1" >>
- "MotorStart"
- "Off"
- "On"
- "On\_for\_5sec"
- "Run"

The "Delay\_5sec" output (a bit in M memory) is no longer required for starting the TOF timer. Instead of deleting the "Delay\_5sec" output, simply rename the output.

1. Double-click the "Delay\_5sec" tag and click the "Tag" icon to display the list of tags.
2. Select the "MotorStart" tag from the list of tags.

You have now combined the logic of networks 2, 3 and 4 onto one network.



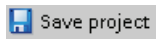
Network 3:

Now delete networks 3 and 4:

1. Select and right-click the network title to display the context menu.
2. Select the "Delete" command from the context menu.



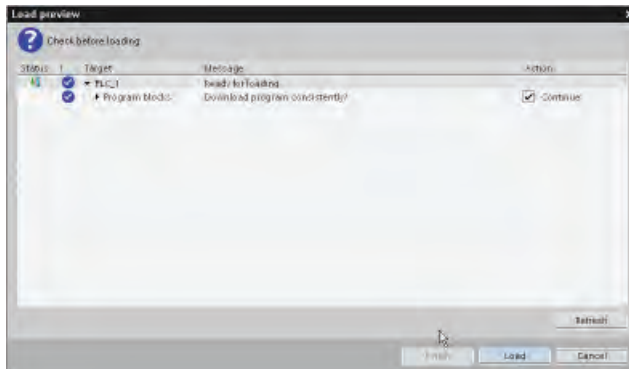
## Saving your work in the project



You can now save your work. Simply click the "Save project" button on the toolbar.



After saving your modified user program, click the "Download" button.



In the "Load preview" dialog, click "Load" to download the user program to the CPU. Select "Start all" to ensure that the CPU goes to RUN mode, and then click "Finish".

## Review of the timer exercise

In this exercise, you completed your user program by adding three networks. You also learned how to use drag-and-drop to edit the elements.



In the next exercise, you will learn how to use a watch table for monitoring the operation of your user program.

*Complete the user program*

*4.3 Editing your user program for a different structure*

---

## Use a watch table for monitoring



When you tested the latching circuit (Page 38), you used the physical switch on the simulator and watched the lights turn on and off on the front of the CPU.

For this exercise, you will use the online functionality of STEP 7 Basic to monitor the operation of your user program.

### What is a "watch table"?

A watch table allows you to monitor or modify the values of tags while the CPU executes your user program.

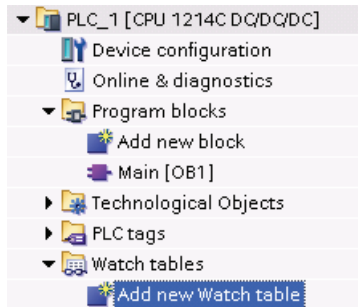
	Name	Address	Display format	Monitor value	Modify value
1	"On"	%I0.0	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>
2	"Off"	%I0.1	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>
3	"Run"	%Q0.0	Bool	<input type="checkbox"/> FALSE	<input type="checkbox"/>



The "Modify" function allows you to change the value of a tag. However, the "Modify" function does not work with inputs (I) or outputs (Q) because the CPU updates the I/O, overwriting any modified value before reading the modified value.

The watch table provides a "Force" function that allows you to modify the values of the I/O. For this exercise, you will learn how to force the input in the latch circuit on.

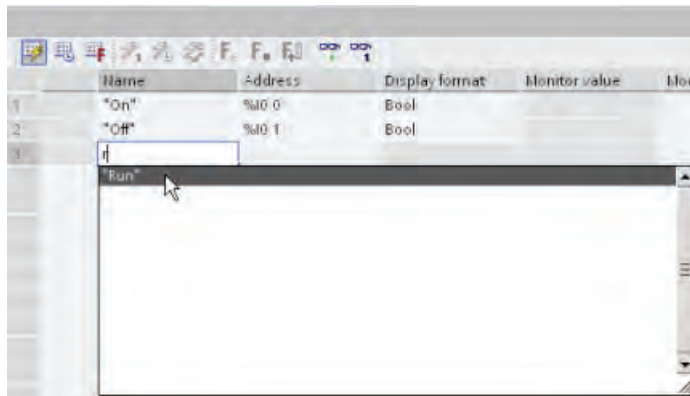
## 5.1 Create a watch table



To create a watch table:

1. Expand the "Watch tables" container.
2. Double-click "Add new watch table" to open a new watch table.

Enter the tags by clicking in the "Name" field. Type a character and select the tag from the list.



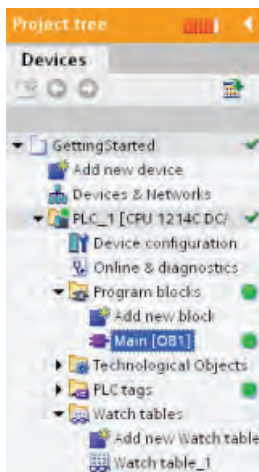
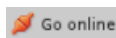
After you create your watch table, you can go online to monitor the execution of your user program.



## 5.2 Monitor the data values in your CPU



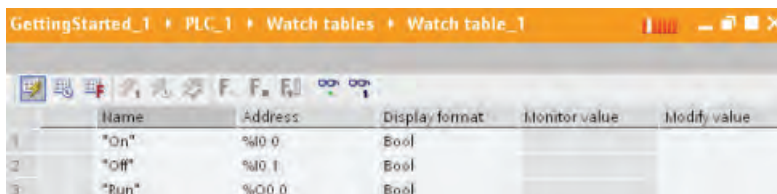
To monitor the tags, you must have an online connection to the CPU. Simply click the "Go online" button in the toolbar.



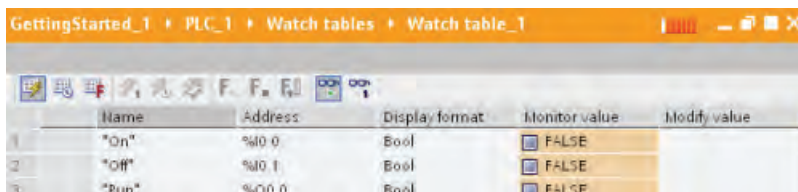
When you have connected to the CPU, STEP 7 Basic turns the headers of the work areas orange.

The project tree displays a comparison of the offline project and the online CPU. A green circle means that the CPU and the project are synchronized, meaning that both have the same configuration and user program.

The watch table shows the tags.

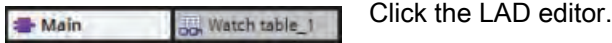


To monitor the execution of the user program and to display the values of the tags, click the "Monitor all" button in the toolbar. The "Monitor value" field shows the value for each tag.

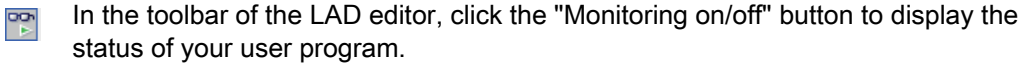


### 5.3 Monitor the status in the LAD editor

You can also monitor the status of the tags in the LAD editor. Use the editor bar to display the LAD editor. The editor bar allows you to change the view between the open editors without having to open or close the editors.



Click the LAD editor.



In the toolbar of the LAD editor, click the "Monitoring on/off" button to display the status of your user program.



The LAD editor displays power flow in green.

With all of the switches on the simulator turned off, notice that the "On" input is not green because it is off (or "false"). Notice also that there is no power flow going to the "Off" contact. However, the normally closed contact "Off" itself is green. "Off" being green does not in itself generate power flow, but means that if power were to flow into the "Off" contact, then the power flow would be passed on to the "Run" coil.

Using the simulator, turn the switch for I0.0 on and watch the power flow across the network. Now, turn I.0.0 off and see how the latching circuit works!



Now turn the switch for I0.1 off to remove power flow from the "Run" coil (Q0.0).

### 5.4 Force an input to a specific value

A watch table provides a "force" function that overwrites the value for an input or output point to a specified value for the peripheral input or peripheral output address. The CPU applies this forced value to the input process image prior to the execution of the user program and to the output process image before the outputs are written to the modules.



Use the editor bar to open the watch table.

Because you cannot force an input (or "I" address), you must change the address for "On" input to access the peripheral input. In the "Address" or "Name" cell for "On", append a ":P" to the address ("On:P").

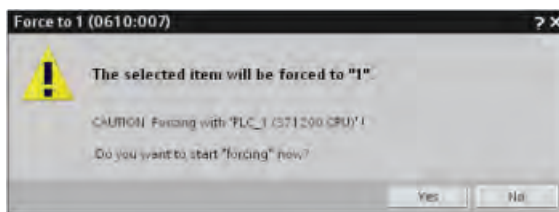
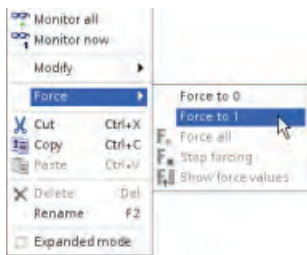
	Name	Address	Display format	Monitor value
1	"On" P	%I 0 P	Bool	<input type="checkbox"/> FALSE
2	"Off"	%I 1	Bool	<input type="checkbox"/> FALSE
3	"Run"	%Q 0	Bool	<input type="checkbox"/> FALSE



Now click the "Show/hide force columns" button to display the "Force value" column.

	Name	Address	Display format	Monitor value	Force value
1	*On:P	%I0.0 P	Bool	FALSE	
2	*Off	%I0.1	Bool	FALSE	
3	*Run	%Q0.0	Bool	FALSE	

Right-click the "Force value" cell to display the context menu. Select the "Force to 1" command to set "On:P" (I0.0) to 1 or "true".



Because the force values are stored in the CPU and not in the watch table, STEP 7 Basic asks you to confirm setting the force value to 1. Click "Yes" to confirm.



Use the "Start or replace forcing" button to force the value of the "On:P" input to 1 (or "true"). Click the "Stop forcing" button to reset the value of "On:P" to 0 (or "false").

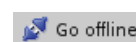
	Name	Address	Display format	Monitor value	Force value
1	*On:P	%I0.0 P	Bool	TRUE	TRUE
2	*Off	%I0.1	Bool	FALSE	
3	*Run	%Q0.0	Bool	TRUE	

With all of the switches on the simulator turned off, notice that when "On:P" is 0 ("false"), the "Run" output remains "true" (1 or on).

### Note

When an input or output is forced in a watch table, the force actions become part of the user program. If you close STEP 7 Basic, the forced elements remain active for the user program being executed by the CPU program until they are cleared. To clear these forced elements, you must use STEP 7 Basic to connect with the online CPU and use the watch table to turn off or stop the force function for those elements.

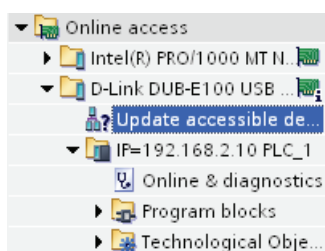
Now click the "Go offline" button to disconnect from the CPU.



## 5.5 Going online

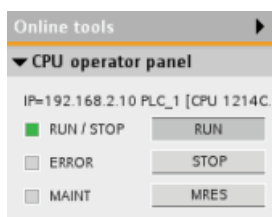
One last bit of information deals with the CPU operator panel. Remember that the CPU does not provide a physical switch for changing modes from STOP to RUN. You use STEP 7 Basic to change the operating mode for the CPU.

First, you access the online CPU.



1. Expand the "Online access" folder and expand the network adapter that connects to the CPU.
2. Double-click "Update accessible nodes" to find your CPU.
3. After STEP 7 Basic displays the CPU, expand the CPU.
4. Double-click "Online & diagnostics" to display the online tools.

The "Online tools" task card provides a variety of tools for monitoring the performance of the CPU, such as the diagnostic buffer, measurements of scan time and memory usage, and the CPU operating panel.



Use the CPU operator panel to change the operating mode. Click the "RUN" or "STOP" buttons to change the operating mode of the CPU.

Be careful! Clicking the "MRES" button performs a memory reset and deletes the work memory of the CPU! (While the MRES does not delete the user program, the non-retentive memory areas are set to the default configuration for the CPU.)

## 5.6 Building on a good beginning

Congratulations on having gotten started with your S7-1200 CPU and the STEP 7 Basic programming software!



There is more to learn, but you should now have a foundation in the skills you need to start creating your own projects.

Refer to the *S7-1200 Easy Book* and the *S7-1200 Programmable Controller System Manual*. These manuals contain useful information that explains the capabilities of the S7-1200.

Don't forget to check out the customer support web site (<http://www.siemens.com/automation/support-request>) for FAQs and other types of useful documentation.



# Index

## A

AND logic, 14, 31  
Appearance of STEP 7, 24

## B

Basic electrical circuits  
    bridge circuit, 16  
    change-over switch, 16  
    diode circuit, 16  
Bit memory (M), 7  
Branch for a network, 31

## C

Changing settings for STEP 7, 24  
Changing the tag address, 47  
Circuit diagrams  
    bridge circuit, 16  
    change-over switch, 16  
    convert to LAD, 13, 14  
    diode circuit, 16

## CPU

bit memory (M), 7  
connecting power, 27  
data types, 11  
force, 54  
inputs and outputs, 7  
load memory, 7  
M memory, 7  
monitoring, 53  
MRES, 56  
number of timers and counters supported, 41  
online, 53  
operating modes, 9, 56  
operator panel, 9, 56  
overview, 7  
physical size, 7, 26  
process image, 7  
program execution, 8  
retentive memory, 7  
safety guidelines, 25, 26  
saving, 32, 34, 37, 44, 46, 48  
unspecified CPU, 23, 35  
work memory, 7

## D

Data types, 11  
Deleting networks, 47  
Detect CPU, 23, 35  
Dimensions, 26  
Discover CPU, 23, 35  
Displaying the contents and index (online help), 19  
Documentation, 19  
Drag and drop  
    between editors, 22  
    between LAD networks, 47  
    from Instruction task card, 42  
    instructions, 21

## E

Electrical circuits  
    bridge circuit, 16  
    change-over switch, 16  
    diode circuit, 16  
Ethernet communication, 28

## F

Favorites toolbar, 21, 31  
Force, 54  
    I memory, 54  
    peripheral inputs, 54

## G

Guidelines for installation, 25, 26

## H

Help, 19  
    Displaying the contents and index, 19  
    Expanding, 19  
    Printing, 20  
    Undocking, 19

- I
- I memory, 12
  - address peripheral inputs, 54
  - changing tags, 47
  - enter addresses, 33
  - force, 54
  - monitor, 53
  - tags, 33
  - watch table, 53, 54
- Information system, 19
  - Displaying the contents and index, 19
  - Printing, 20
  - Undocking, 19
- Input simulator
  - installing, 27
  - testing with, 38
- Inputs and outputs, 7, 12
  - forcing, 54
  - monitoring, 53
  - scan cycle, 8
- Installation, 26, 27, 28
  - safety guidelines, 25, 26
- Instructions
  - changing the address of an instruction, 47
  - coil, 31
  - contacts, 31
  - drag and drop between editors, 22
  - editing networks, 47
  - favorites, 21, 31
  - force, 54
  - inserting, 21, 31
  - Instruction task card, 45
  - monitor, 53
  - new branch, 31
  - status, 53, 54
  - TOF (off-delay timer), 44
  - TON (on-delay timer), 41
- L
- LAD
  - convert from a circuit diagram, 13, 14
  - edit instructions, 47
  - insert instructions, 31, 42, 45
  - monitor, 53
  - new branch, 31
  - status, 53, 54
  - tags, 33
- Latch circuit, 31
- LEDs, 7
- Load memory, 7
- MRES, 56
- M
- M memory, 7
  - changing tags, 47
  - enter addresses, 42
  - tags, 42
- Memory areas, 7
  - address peripheral inputs, 54
  - I memory, 12
  - M memory, 12
  - peripheral I/O, 12
  - process image, 12
  - Q memory, 12
- Memory reset (MRES), 9, 56
- Monitoring
  - create a watch table, 52
  - force, 54
  - LAD status, 53
  - watch table, 52, 53, 54
- N
- Network
  - add branch, 31
  - deleting networks, 47
- Number of timers and counters supported by S7-1200, 41
- O
- Off-delay (TOF) instruction, 44
- On-delay (TON) instruction, 41
- Online
  - connecting, 56
  - force, 54
  - MRES, 56
  - operator panel, 9, 56
  - watch table, 53
- Online help, 19
  - Displaying the contents and index, 19
  - Expanding the help window, 19
  - Printing, 20
  - Undocking, 19
- Operating mode, 8, 9, 56
- Operator panel, 9, 56
- OR logic, 14, 31

**P**

- Physical size, 26
- PLC
  - bit memory (M), 7
  - data types, 11
  - force, 54
  - inputs and outputs, 7
  - load memory, 7
  - M memory, 7
  - monitoring, 53
  - MRES, 56
  - number of timers and counters supported, 41
  - overview, 7
  - physical size, 7, 26
  - process image, 7
  - program execution, 8
  - retentive memory, 7
  - saving, 32, 34, 37, 44, 46, 48
  - unspecified CPU, 23, 35
  - work memory, 7
- Portal view, 18
- Printing the help topics, 20
- Process image, 7, 12
  - force, 54
  - monitor, 53
  - status, 53, 54
- PROFINET, 28
- Programming
  - deleting networks, 47
  - drag and drop between editors, 22
  - editing, 47
  - favorites, 21
  - program execution, 8
- Project view, 18

**Q**

- Q memory, 12
  - changing tags, 47
  - tags, 33

**R**

- Retentive memory, 7
- RUN mode, 9, 56
  - force, 54
  - program execution, 8

**S**

- S7-1200
  - force, 54
  - monitoring, 53
  - MRES, 56
  - operator panel, 56
  - physical size, 26
  - safety guidelines, 25, 26
  - saving, 32, 34, 37, 44, 46, 48
- Safety guidelines, 25, 26
- Saving, 32, 34, 37, 44, 46, 48
- Saving the project, 31
- Scan cycle, 8
  - force, 54
- Settings for STEP 7, 24
- Simulator
  - installing, 27
  - testing with, 38
- Starting STEP 7 Basic, 28
- STARTUP mode
  - force, 54
  - program execution, 8
- Status LEDs, 7
- STEP 7
  - data types, 11
  - detect CPU, 23, 35
  - force, 54
  - installation, 28
  - languages, 24
  - monitoring, 53
  - MRES, 56
  - operator panel, 9, 56
  - Portal view, 18
  - Project view, 18
  - saving, 32, 34, 37, 44, 46, 48
  - settings, 24
  - starting, 28
  - watch table, 52
- STOP mode, 9, 56
  - force, 54

**T**

- Tags
  - changing the address, 47
  - force, 54
  - I memory, 33
  - M memory, 42
  - monitor, 53
  - status, 53, 54
  - tag table, 33, 47

TIA Portal  
Portal view, 18  
Project view, 18

Timers  
number of timers and counters supported, 41  
TOF (on-delay), 44  
TON (on-delay), 41  
Transform a circuit diagram, 13, 14

## U

Unspecified CPU, 23, 35  
Upload, 35

User interface  
Portal view, 18  
Project view, 18

## W

Watch table  
create, 52  
force, 51, 54  
monitor, 51, 53  
Work memory, 7