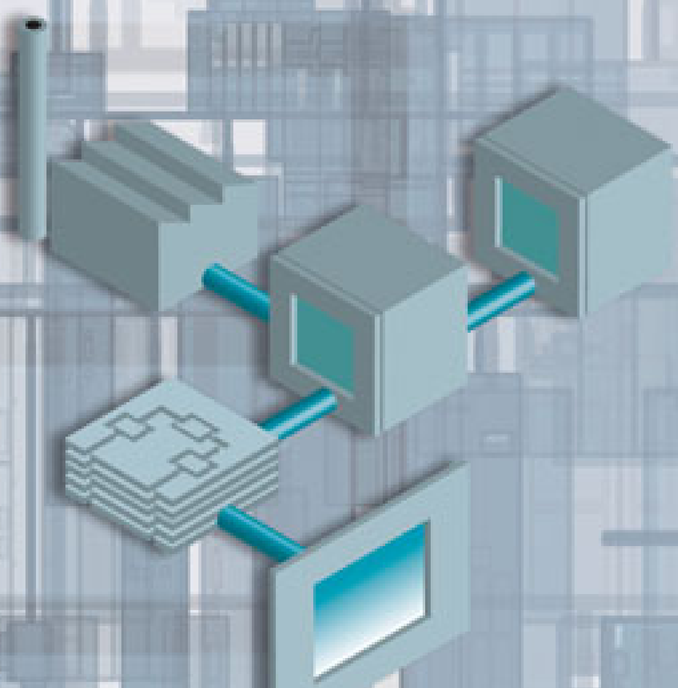
Hans Berger

Automating with SIMATIC

Controllers, Software, Programming, Data Communication, Operator Control and Process Monitoring

SIEMENS



Fifth Edition

Berger Automating with SIMATIC

Automating with SIMATIC

Controllers, Software, Programming, Data Communication, Operator Control and Process Monitoring

by Hans Berger

5th revised and enlarged edition, 2013

Publicis Publishing

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

www.publicis-books.de

Print ISBN: 978-3-89578-387-6
ePDF ISBN: 978-3-89578-676-1

5th edition, 2013

Editor: Siemens Aktiengesellschaft, Berlin and Munich
Translation: Siemens I IA Control Components and Systems Engineering, Erlangen
Publisher: Publicis Publishing, Erlangen

Printed in Germany

Foreword

The automation of industrial plants results in a growing demand for components which are increasingly different and more complex. Therefore a new challenge nowadays is not the further development of highly specialized devices but the optimization of their interaction.

The *Totally Integrated Automation* concept permits uniform handling of all automation components using a single system platform and tools with uniform operator interfaces. These requirement is fulfilled by the new SIMATIC, which provides uniformity for configuration, programming, data management, and communication.

The STEP 7 engineering software is used for the complete configuration and programming of all components. Optional packages for expanding functionalities can also be introduced seamlessly in STEP 7 if they have the same operating philosophy. The SIMATIC Manager of STEP 7 V5.5 and the TIA Portal of STEP 7 V11 coordinate all tools and centrally manage any automation data. All tools have access to this central data management so that duplicate entries are avoided and coordination problems are prevented right from the start.

Integrated communication between all automation components is a prerequisite for “distributed automation”. Communication mechanisms that are tuned to one another permit the smooth interaction of controllers, visualization systems, and distributed I/O without additional overhead. This puts the seminal concept of “distributed intelligence” within reach. Communication with SIMATIC is not only uniform in itself, it is also open to the outside. This means that SIMATIC applies widely-used standards such as PROFIBUS for field devices and Industrial Ethernet and TCP/IP protocol for the best possible connections to the office world and thus to the management level.

The 5th edition of this book provides an overview of the structure and principle of operation of a modern automation system with its state-of-the-art controllers and HMI devices, and describes the expanded facilities of distribution with PROFIBUS and PROFINET. Using the SIMATIC S7 programmable controllers as example, this book provides an insight into the hardware and software configuration of the controller, presents the programming level with its various languages, explains the exchange of data over networks, and describes the numerous possibilities for operator control and monitoring of the process.

Erlangen, July 2012 Hans Berger

Contents

1 Introduction

1.1 Components of the SIMATIC Automation System

The SIMATIC automation system consists of many components that are matched to each other through the concept of “Totally Integrated Automation” (TIA). Totally Integrated Automation means automation with integrated configuration, programming, data storage, and data transfer.

As programmable logic controllers (PLCs), the **SIMATIC S7 controllers** form the basis of the automation system. SIMATIC S7-200 and S7-1200 are micro systems for the low-end performance range – as a stand-alone solution or in a bus network. The SIMATIC S7-300 with standard CPUs and compact CPUs is the system solution with a focus on the manufacturing industry. And as the top-level device with the highest performance power of the SIMATIC controllers, the SIMATIC S7-400 enables system solutions for the manufacturing and process industries.

SIMATIC WinAC (Windows Automation Center) combines the functions of open-loop control, technology, data processing, visualization, and communication on one personal computer (PC) and is the first choice if you have to handle PC applications in addition to classic PLC tasks.

SIMATIC WinCC (Windows Control Center) is the engineering and runtime software for PC-based devices. HMI devices are configured with WinCC as engineering software, while WinCC as runtime software turns personal computers into HMI devices for industrial plants and processes.

SIMATIC HMI stands for Human Machine Interface and is the interface between operators and the machine. From the simplest text display to the operator station with powerful graphics, the human-machine interface provides all the facilities you need for operating and monitoring a machine or plant. Powerful software indicates the state of the plant with event and fault messages, manages recipes and measured value archives, and supports plant operators with troubleshooting, servicing, and maintenance.

SIMATIC NET links all SIMATIC stations and ensures trouble-free data communication. Various bus systems with graded performance also allow third-party devices to be connected, whether field devices in the plant or process PCs connected at the control level. Data traffic can go beyond the limits of various subnets, such as the transfer of automation data such as measured values and alarms or the commissioning and troubleshooting of a central location in the network group.

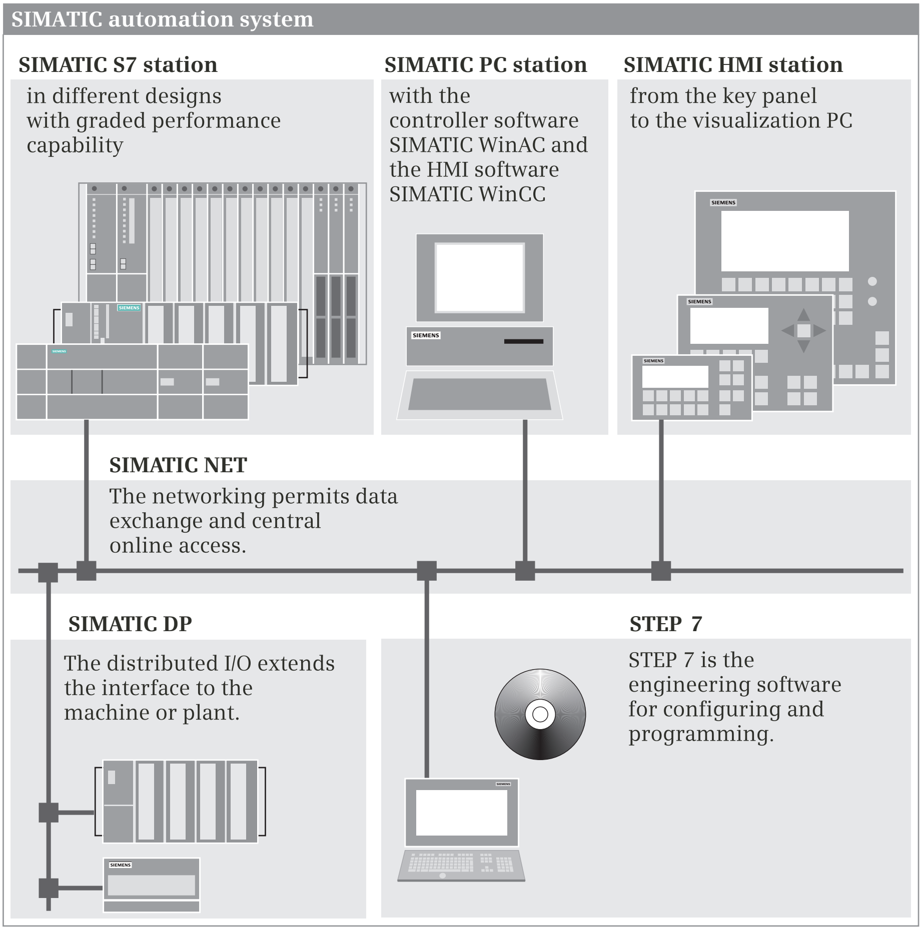


Fig. 1.1 Components of the SIMATIC automation system

SIMATIC DP stands for distributed I/O. It expands the interface between the central controller and machine or plant with I/O modules directly on site. This distributed I/O, which is spatially separate from the controller, is connected to the central control via the PROFIBUS DP and PROFINET IO bus systems – thus reducing wiring overhead. SIMATIC controllers, ET 200 I/O system modules, and third-party devices can be used as distributed I/O.

The **STEP 7** engineering software is used to configure, parameterize, and program SIMATIC components. The SIMATIC Manager in the “classic” STEP 7 version and the TIA Portal in the updated version are the central tools for managing automation data and related software editors in the form of a hierarchically organized project.

The main activities performed with STEP 7 are:

- ▷ Configuring the hardware
  (arranging modules in racks and parameterizing the module properties)
- ▷ Configuring the communication connections
  (defining communication partners and connection properties)
- ▷ Programming the user program
  (creating the control software and testing the program)

The user program can be created in the programming languages ladder logic (LAD), function block diagram (FBD), statement list (STL), and structured control language (SCL) as “logic control”, or in the programming language GRAPH as “sequence control”

1.2 From the Automation Task to the Finished Program

When you start to solve an automation problem, you have to ask yourself what type of PLC you are going to use. If the machine to be controlled is a small one, will an S7-200 be big enough or do you need an S7-300? Is it better to control the plant with an S7-400 or with a pair of S7-300s? Compact central I/Os in the control cabinet or distributed I/Os in the plant?

The following is a general outline of the steps that lead from the automation task to the finished program. In individual cases, specific requirements must be met.

Choosing the hardware

There are many criteria for selecting the type of controller. For “small” controls the main criteria are the number of inputs and outputs and the size of the user program. For larger plants you need to ask yourself whether the response time is short enough, and whether the user memory is big enough for the volume of data to be managed (recipes, archives). To be able to estimate the resources you need from the requirements alone, you need a lot of experience of previous automation solutions; there is no rule of thumb.

A production machine will probably be controlled by a single station. In this case, the number of inputs/outputs, the size of the user memory and, possibly, the speed (response time) will enable you to decide between the S7-200, S7-1200, S7-300, or the S7-400. How is the machine controlled? What HMI devices will be used?

Spatially distributed systems raise the question of what is overall the better value: the use of centralized or distributed I/O. In many cases distributed I/O not only reduces the wiring overhead needed, but also the response time and the engineering costs. This is possible due to the use of “intelligent” I/O devices with their own user program for preprocessing of signals “on site”.

Distributed automation solutions have advantages: The user programs for the different plant units are smaller with faster response times, and can often be commissioned independently of the rest of the plant. The necessary exchange of data with a “central controller” is particularly easy within the SIMATIC system using standardized bus systems.

Which programming language?

The choice of programming language depends on the task. If it mainly consists of binary signal processing, the graphical programming languages LAD (Ladder Logic) and FBD (Function Block Diagram) are ideal. For more difficult tasks requiring complex variable handling and indirect addressing, you can use the STL (Statement List) programming language, which has an assembly language format. SCL (Structured Control Language) is the best choice for people who are familiar with a high-level programming language and who mainly want to write programs for processing large quantities of data.

If an automation task consists of sequential processes, GRAPH can be used. GRAPH creates sequencers with steps and step enabling conditions that are processed sequentially. All programming languages – including GRAPH – can be used together in a user program. Every program section and “block” can be created with the suitable programming language, depending on requirements.

Creating a project

All the data for your automation solution is collected together in a “project”. You create a project using STEP 7. A project is a (software) folder in which all the data is stored in a hierarchic structure. The next level down from the project are the “stations”, which in turn contain one or more CPUs with a user program. All these objects are folders which can contain other folders or objects that represent the automation data on the screen. You use menu commands to insert new objects, open these objects, and automatically start the tool required to work with them.

An example: The user program consists of blocks, which are individual program sections with limited functions. All programmed blocks are listed in the block folder. Depending on the programming language used, double-clicking on a block starts the suitable program editor, with which you can alter or expand the program in the block, guided by menus and supported by online help.

Configuring hardware

A project must contain at least one station, either a programmable logic control (PLC) station or a human-machine interface (HMI) station. A PLC station is required to control a machine or plant. After the station is opened, a rack is shown onscreen, to which you can add the desired modules. To do this, drag the required modules from the hardware catalog to the relevant slot. If needed, change the default module properties to meet your requirements.

A project can contain additional stations that you configure in the same manner as the first station. The data transfer between the stations takes place via a subnet.

Using network configuration, you connect the bus interfaces of the “communication modules” to the subnet and thus create the network group.

Writing, debugging and saving the user program

The user program is the totality of all instructions and declarations programmed by the user for signal processing by means of which the machine or plant to be controlled is influenced in accordance with the control task. Large, complex tasks are easier to solve if they are divided into small, manageable units, which can be programmed in “blocks” (subroutines). The division can be process-oriented or function-oriented. In the first case, each program unit corresponds to a part of the machine or plant (mixer, conveyor belt, drilling assembly). In the second case, the program is divided up according to control functions, for example signaling, communication, operating modes. In practice, mixed forms of the two structuring concepts are generally used.

In the user program, signal states and variable values are used that you should preferably address with a name (symbolic addressing). A name is assigned to a memory location in the symbol table or in the PLC variable table. You can then use the name in the program. After you have entered the user program you “compile” it so that it can process the relevant control processor. The user program is created “offline”, without a connection to a controller, and is saved to the hard disk of the programming device.

You can test smaller programs, as well as individual parts of larger programs, offline with the PLCSIM simulation software and thus find and correct any possible errors before the user program is used in the machine or plant.

For commissioning, connect the programming device with the CPU, transfer the program to the CPU user memory, and test it using the STEP 7 testing functions. You can monitor and change the variable values and monitor the processing of the program by the control processor. Comprehensive diagnostic functions allow quick identification of error location and cause.

After commissioning is concluded, you document the project, e. g. in circuit manual format, by means of DOCPRO. With the “classic” version of STEP 7, you can archive an entire project, including documentation, in compressed form.

1.3 How Does a Programmable Logic Controller Work?

In conventional control engineering, a control task is solved by wiring up contactors and relays individually, i.e. depending on the task. They are therefore referred to as contactor and relay controllers, and electronic controllers assembled from individual components are referred to as *hard-wired programmed* controllers. The “program” is in the wiring. *Programmable logic* controllers, on the other hand, are made up of standard components that implement the desired control function by means of a userprogram.

SIMATIC S7 is an automation system that is based on programmable logic controllers. The solution of the control task is stored in the user memory on the CPU in the form of program instructions. The control processor reads the individual instructions sequentially, interprets their content, and executes the programmed function.

The CPU module contains an additional program: the operating system. It ensures the execution of the device-internal operating functions, such as communication with the programming device and backing up data in the event of power failure. The operating system also initiates the processing of the user program, either recurring cyclically or dependent on a trigger event such as an alarm.

Cyclic program processing

The prevalent processing type of the user program for programmable logic controllers is cyclic program processing: After the user program has been completely processed once, it is then processed again immediately from the beginning. The user program is also executed if no actions are requested “from outside”, such as if the controlled machine is not running. This provides advantages when programming: For example, you program the ladder logic as if you were drawing a circuit diagram, or program the function block diagram as if you were connecting electronic components. Roughly speaking, a programmable logic controller has characteristics like those of a contactor or relay control: The many programmed operations are effective quasi simultaneously “in parallel”.

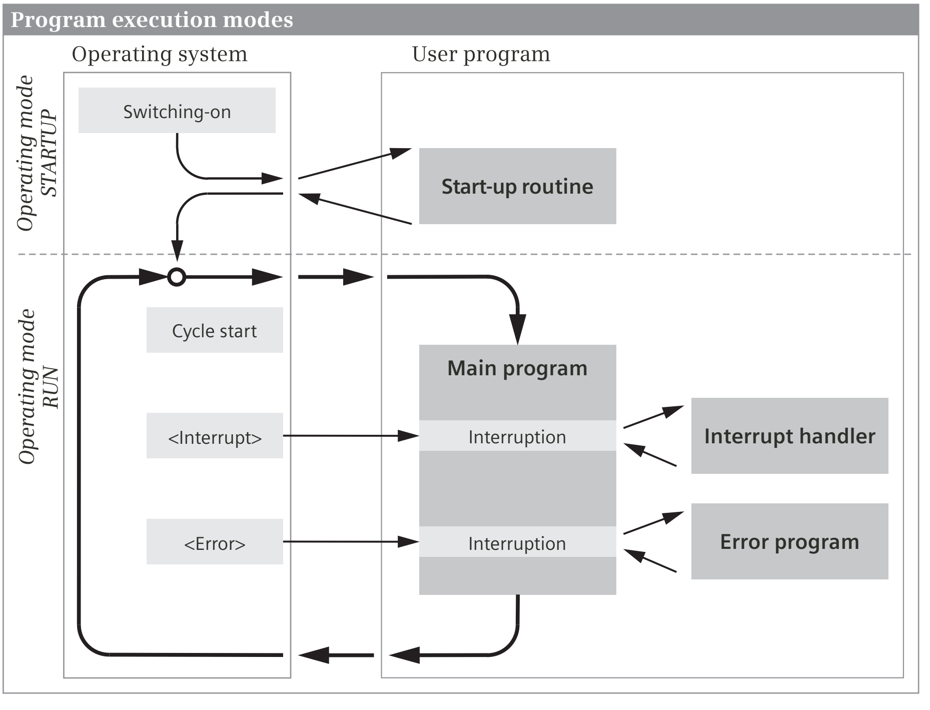


Fig. 1.2 Execution of the user program in a SIMATIC controller

After the power is switched on and the operating function test runs, the operating system starts an (optional) start-up routine once (Fig. 1.2). The main program is next in the sequence. If it has been processed to the end, processing begins again immediately at the start of the program. The main program can be interrupted by alarm or error events. The operating system then starts an interrupt handler or error program. If the interruption-controlled program has been completely processed, the program processing continues from the point of interruption in the main program. A priority scheduler controls the program execution order if several interrupt events occur simultaneously.

The user program is made up of blocks. There are several block types. Organization blocks represent the interface to the operating system. After a start event occurs (power up, cycle start, alarm, error), the operating system calls the associated organization block. It contains the appropriate user program for the event. An organization block only has to be programmed if the automation solution requires it. The program in an organization block can, if needed, be structured by function blocks (blocks with static local data) and functions (blocks without static local data). Data blocks in the user memory or the bit memory address area in the system memory are available to store user data.

1.4 The path of a binary signal from the sensor to the program

In order to do its job, the control processor in the controller needs to be connected to the machine or plant it is controlling. I/O modules that are wired to the sensors and actuators create this connection.

Connection to the programmable logic controller, module address

When wiring the machine or plant, you define which signals are connected to the programmable logic controller, and where. An input signal, e.g. the signal from pushbutton +HP01-S10 with the significance “Switch on motor”, is connected to a specific terminal on an input module (Fig. 1.3).

Each module is located in a particular slot on the rack, the number of which is the slot address. In addition, each I/O module has a so-called “logical” address. The user program uses this address to address a signal of the module. In the logical address, the binary signals are aggregated into bytes (bundles of eight bits). Bytes are numbered starting from zero – even with gaps. The bit address is counted from 0 to 7 for each byte.

You determine the slot address by plugging the module into a certain place on the rack. STEP 7 assigns the logical address consecutively, which you can change in the configuration table. The first byte of the module receives the module start address, which is the lowest address of the module. If a module has more bytes, the byte addresses are automatically incremented. In the example, the module has the module start address four – either set by STEP 7 as default or set by you – and the

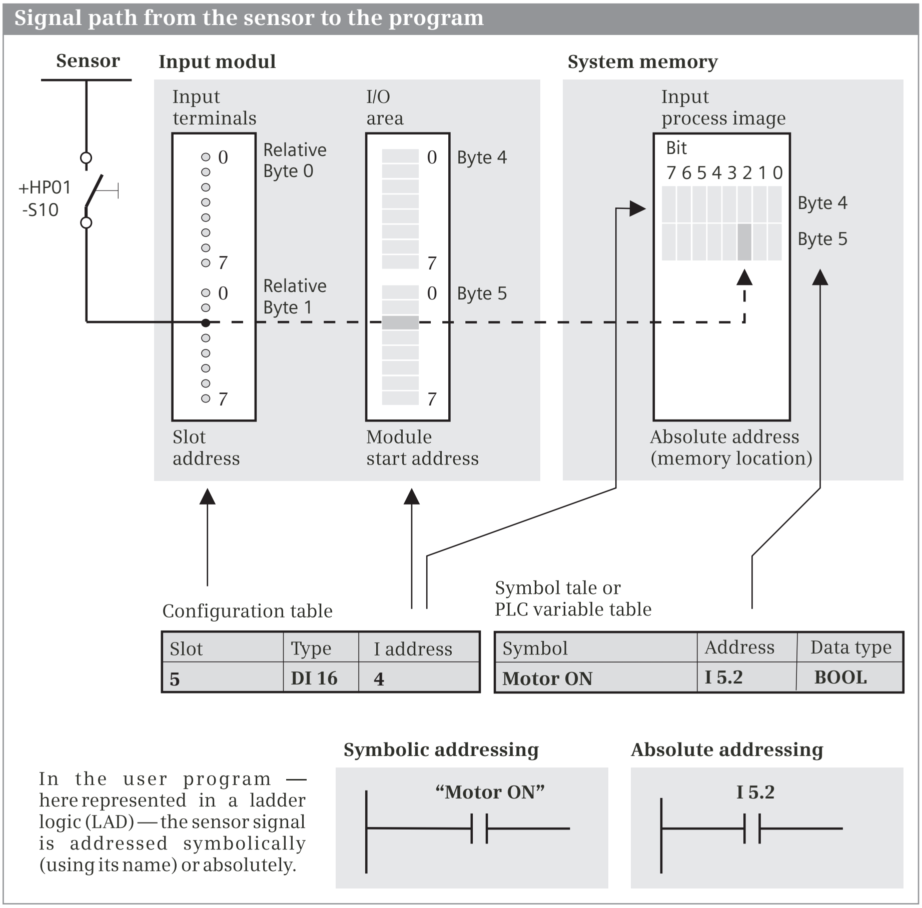


Fig. 1.3 Path of a signal from the sensor to its use in the program

next byte thus automatically has the number five. The "switch on motor" signal is connected to terminal two of the second byte (relative byte 1). By specifying the module address as 4, you are given the address of the signal: "Input in byte 5 to bit 2" or in short: I 5.2.

Symbolic address

The address "I 5.2" denotes the memory location and is the absolute address. It is much easier if you can address this signal in the program with a name that matches the meaning of the signal, for example "switch on motor". This is the symbolic address. You can find the assignment of absolute addresses to symbolic addresses in the symbol table or the PLC variable table. In this table, the "global" symbols are defined; these are the symbols that are valid in the entire user program. You specify the symbols that are valid for only one block ("local" symbols) when programming the block.

Process images

When you use the signal "switch on motor" or I 5.2 in the program, you do not address the signal memory in the module but a storage area within the CPU. This storage area is referred to as the "process image". It is also available for the outputs, which in principle are treated in the same way as the inputs.

The CPU operating system automatically transfers the signal states between the modules and the process image in each program cycle. It is also possible to address the signals directly on the modules from the user program. However, the use of a process image has advantages compared to direct access, including much faster access to the signal states and the steady signal state of an input signal during a program cycle (data consistency). The disadvantage is the increased response time, which is also dependent on the program execution time.

1.5 Data management in the SIMATIC automation system

The automation data is present in various memory locations in the automation system. First of all, there is the programming device. All automation data of a STEP 7 project is saved on its hard disk. Configuration and programming of the project data with STEP 7 are carried out in the main memory of the programming device (Fig. 1.4).

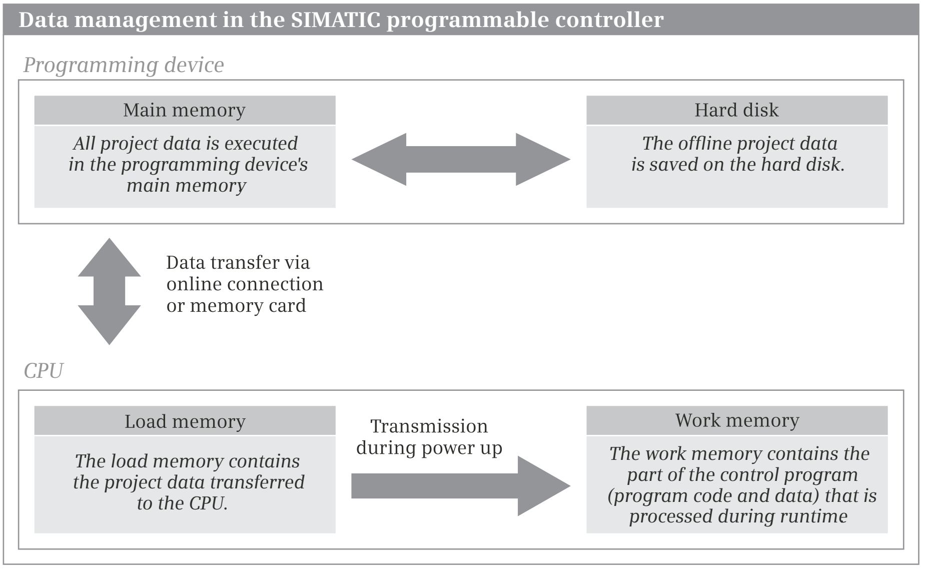


Fig. 1.4 Data management in the SIMATIC automation system

The automation data on the hard disk is also referred to as the *offline project data*. Once STEP 7 has appropriately compiled the automation data, this can be downloaded to a connected programmable logic controller. The data downloaded into the user memory of the CPU is known as the *online project data*.

The user memory on the CPU is divided into two components: The *load memory* contains the complete user program, including the configuration data, and the *work memory* contains the executable user program with the current control data. The load memory of the CPU 1200 can be expanded with a plug-in memory card. For the CPU 300, the load memory is on the memory card, which therefore must always be inserted in the CPU for use. The memory card for the CPU 1200 and CPU 300 is an SD Card, for failsafe storage of the automation data. On the CPU 400, the memory card expands the load memory; here, the memory card is a RAM card (so that the user program can be changed during testing), or a FEPROM card (for failsafe storage of the user program).

2 SIMATIC Controllers – the Hardware Platform

SIMATIC controllers – the core of the automation systems – control production machines, manufacturing plants, or industrial processes. The following description mainly refers to programmable logic controllers (PLC). Siemens also offers PC-based SIMATIC controllers.

SIMATIC S7 are programmable logic controllers (PLCs), which are available in four designs:

▷ SIMATIC S7-200, the compact micro PLC

▷ SIMATIC S7-1200, the modular „micro PLC“,

▷ SIMATIC S7-300, the modular PLC of the mid performance range

▷ SIMATIC S7-400, the modular PLC of the upper performance range

The S7-200 station consists of a basic unit and can be expanded with additional modules. In S7-300/400 stations, the power supply unit, the CPU module and the I/O modules are installed in the same mounting rack. This centralized configuration can be extended with expansion racks for the installation of additional I/O modules. The expansion rack may be a remote installation, that is it can be placed at a distance from the central rack. You use the STEP 7 Micro programming language to program the SIMATIC S7-200 controller. STEP 7 with its different programming languages is provided for the controllers of the SIMATIC S7-300/400 series.

PC-based automation is the umbrella term for PLCs based on a personal computer (PC):

▷ The industrial PC is available as a Rack PC or Box PC.

▷ The SIMATIC Panel PC is a combination of HMI device and controller.

▷ **SIMATIC WinAC** is the generic term for the program packages of SIMATIC PC-based Automation. WinAC runs on a standard PC under a Windows operating system. Distributed I/O modules form the link to the process.

  With SIMATIC PC-based Control, the controller can take the form of a purely software solution (Software PLC) or a plug-in card (Slot PLC).

SIMATIC DP are modules installed on site at the machine or in the plant and are connected to the master station via PROFIBUS DP and/or PROFINET IO. Many SIMATIC CPUs feature an integrated PROFIBUS or PROFINET interface, which greatly facilitates the connection of distributed I/O. Since operation on the

PROFIBUS and PROFINET is standardized independent of the vendor, it is also possible to connect third-party devices to a SIMATIC controller.

2.1 Components of a SIMATIC Station

A complete programmable controller including all I/O modules is referred to as a “station”. The core is the CPU, which is expanded with I/O modules if needed.

The following list shows the components a SIMATIC station can consist of:

▷ Racks
These accommodate the modules and form the basis for central and expansion units. The S7-1200 and S7-300 use a simple mounting rail; its length is determined by the number and width of the modules. The S7-400 uses an aluminum rack that has a defined number of slots with backplane bus and bus connectors.

▷ Power supply (PS)
It provides the internal supply voltage; the input voltage is either 120/230 V AC voltage or 24 V DC voltage.

▷ Central processor unit (CPU)
This stores and processes the user program; communicates with the programming device and any other stations over the MPI bus; controls the central and distributed I/O modules; and can also be a DP slave on PROFIBUS DP or IO Device on PROFINET IO.

▷ Interface modules (IM)
These connect the racks with each other.

▷ Signal modules (SM)
These adapt the signals from the controlled plant to the internal signal level or control contactors, actuators, lights, etc. Signal modules are available as input and output modules for digital and analog signals and can also be used to connect sensors and actuators located in hazardous areas of zones 1 and 2.

▷ Function modules (FM)
These handle complex or time-critical processes independently of the CPU, e.g. counting, position control, and closed-loop control.

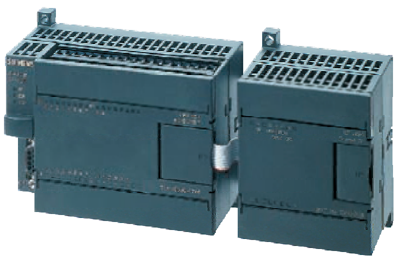
▷ Communications processors (CP)
These connect the SIMATIC station with the subnets such as Industrial Ethernet, PROFIBUS FMS, AS-Interface, or a serial point-to-point connection.

The distributed I/O modules connected to a station are also part of this station. If the distributed I/O system is connected over PROFIBUS DP, a DP master controls “its” DP slaves and thus the field units; if the connection is over PROFINET IO, an IO-Controller controls the IO-Devices. The DP slaves or IO-Devices are integrated in the address area of the centralized I/O system and are principally treated just like the I/O modules installed locally in the central and expansion racks.

2.2 The Micro PLC SIMATIC S7-200

The SIMATIC S7-200 is a compact micro PLC that replaces relay and contactor control arrangements and increasingly also specific electronic circuits in mechanical and system engineering applications. It can be used as a stand-alone system or in a network with other control systems. Various expansion modules for the connection to the machine or plant are available. STEP 7 Micro/WIN is used for programming a SIMATIC S7-200.

Setup of an S7-200 station



The figure shows a CPU 224, version AC/DC/RLY, i.e. 230 V supply voltage, 24 V load voltage for inputs and implementation of the outputs with a relay. Next to it on the right is a CP 243-1 communications processor, which permits the connection of the SIMATIC S7-200 station to Industrial Ethernet. The expansion modules are connected to the CPU by means of a flexible connecting cable.

Expansion options

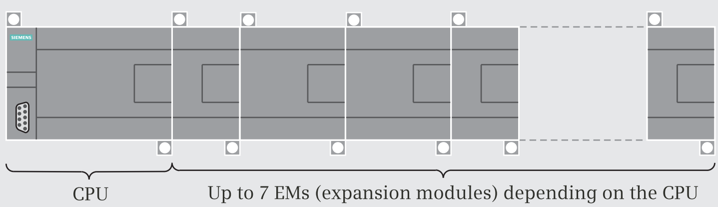


Fig. 2.1 Setup and expansion options for an S7-200 station

Compact-type basic modules and expansion modules

There are various S7-200 basic modules graded by configuration and performance capability. The basic module contains the CPU and – in different types and numbers for each basic module – integrated inputs and outputs at 24 V DC, 100 V AC/120 V to 230 V AC, as well as relay outputs. Fast alarm and counter inputs enhance the real-time performance. A digital value can be set without a programming device using a screwdriver via a potentiometer on the basic module.

You can increase the number of inputs/outputs by connecting an expansion module. Such modules are available for both digital and analog inputs/outputs. The expansion modules are snapped onto the rail (e.g., standard DIN rail) next to the basic module and electrically connected by means of a bus connector.

Operating modes of a CPU 200

A CPU 200 has two operating modes, STOP and RUN, which are indicated by LEDs on the front of the CPU. The user program is not executed in STOP mode. The CPU must be in STOP mode to load the user program or the hardware configuration. In RUN mode, the CPU executes the control functions in the user program. To switch between modes, use the mode switch on the CPU or the programming device in online operation.

User memory in a CPU 200

The user memory of a CPU 200 consists of the program memory and the data memory. When loading the user program to the CPU with a programming device, the program code and the data are stored in an EEPROM memory, where they are protected against power outages. The current data is then copied to the RAM area where it is processed during runtime.

The user program can also be transferred via a memory module that can be plugged into the CPU. Additional data such as recipes, data log configurations, and documentation files in any format can be stored on this memory module. This additional data can also be loaded from a programming device to the CPU if the memory module is plugged in.

On a CPU 200, you can use a program to store a variable value in the failsafe EEPROM memory area. Note that – due to the physical limitations of the medium – only a limited number of write operations is allowed. Excessive writing, for example with every program cycle, reduces the lifespan of the EEPROM memory. The data on any plugged-in memory module is not changed.

Table 2.1 Quantity structure of the S7-200 CPUs

CPU	**Integrated I/O** Input/output channels			**Expandable with** Expansion modules (EM) of the S7-22x series	**Memory configuration** Data memory / program memory / with active RunTime Edit	**Counter** Number / frequency
	DI	DO	AI *)			
CPU 221	6	4	1	cannot be expanded	2 KB/ 4 KB	4/30 kHz
CPU 222	8	6	1	Max. 2 EMs	2 KB/ 4 KB	4/30 kHz
CPU 224	14	10	2	Max. 7 EMs	8 KB/ 12 KB/ 8 KB	6/30 kHz
CPU 224 XP	14	10	2	Max. 7 EMs	10 KB/ 16 KB/ 12 KB	2/200 kHz 4/30 kHz
CPU 226	24	16	2	Max. 7 EMs	10 KB/ 24 KB/ 16 KB	6/30 kHz

*) Analog potentiometer with 8-bit resolution

Retentive behavior

Part of the data memory is reserved for retentive data defined by the user. Retentive data is even retained if the supply voltage is interrupted. A high-performance capacitor or an optional battery module bridges the loss of voltage.

If, when the power supply is switched on, the high-performance capacitor and – if present – the optional battery module do not exhibit errors, the values of the retentively configured variables (bit memories, timers, counters) are not changed and the non-retentive memory areas are cleared.

If the contents of the RAM cannot be buffered, the CPU 200 deletes all user data, retrieves the user program from the EEPROM memory area, and restores the first 14 bytes of the bit memories from the non-volatile memory, as long as these bytes have previously been configured as retentive.

Communication capability just like a “large” station

Depending on its design, a CPU 200 hasone or two RS-485 interfaces, which as point-to-point interfaces (PPI) permit the connection of programming and HMI devices and allow linking to another CPU 200. As multi-point interfaces (MPI), they permit the operation of a CPU 200 as MPI slave on an MPI master, e.g. a CPU 300/400. This interface can also be used as a freely programmable interface with interrupt facility for serial data exchange with third-party devices such as barcode readers using the ASCII protocol.

Various supplementary modules expand the communication possibilities, such as the EM 241 modem for remote maintenance and diagnostics, the EM 277 DP module for operation of a CPU 200 as PROFIBUS DP slave, and the CP 243-2 (AS-Interface master), CP 243-1 (connection to Industrial Ethernet), and CP 243-1 IT (Industrial Ethernet with IT communication) communications processors.

Operator control and monitoring with S7-200

For the S7-200 controllers, the Micro Panels are the ideal operator control and monitoring devices. They are connected to the interface of the CPU by means of a supplied cable. For a description of these HMI devices, refer to chapter 7.5 "Micro Panels" on page 259.

Configuring and programming with STEP 7-Micro/Win

STEP 7-Micro/Win is the engineering software for the controllers of the S7-200 series and runs under Windows 2000 and Windows XP. A CPU 200 can be programmed with the statement list (STL), ladder logic (LAD), and function block diagram (FBD) programming languages. These programming languages for S7-200 differ in function and handling from the programming languages for the other SIMATIC S7 automation systems.

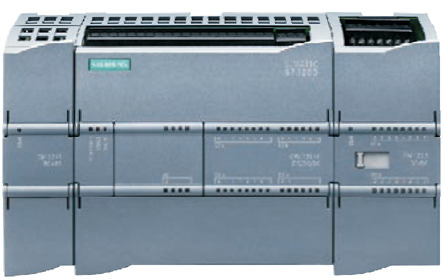
2.3 The SIMATIC S7-1200 Modular Micro Controller

The youngest member of the controller family is the SIMATIC S7-1200. An S7-1200 automation system consists of a central processing unit which – depending on the CPU version – can be expanded with digital and analog input and output modules. Using the PROFINET interface, the central processing unit can be connected to Industrial Ethernet. S7-1200 is configured and programmed inside TIA Portal using STEP 7 Basic/Professional.

Compact design for S7-1200

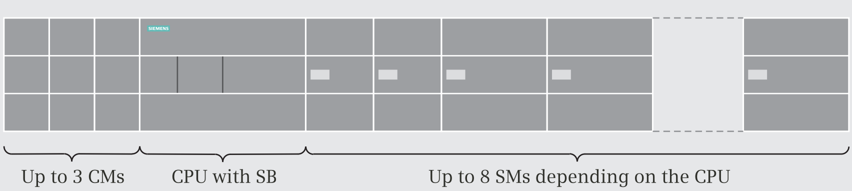
Three central processing units with different performance capability in each of the variants DC/DC/DC, DC/DC/relay, and AC/DC/relay are offered. The first specification refers to the supply voltage (24 V DC, 85 to 264 V AC), the second to the signal voltage of the digital inputs (24 V DC), and the third to the type of digital outputs (24 V DC electronic or relay outputs 5 to 30 V DC, 5 to 250 V AC). Table 2.2 shows the expandability and the memory configuration. Rapid counters with counting frequencies of up to 100/200 kHz are integrated with the central processing unit, which in connection with a pulse generator and the "Axis" technology object can control a stepper motor or a servomotor with pulse interface.

Setup of an S7-1200 station



The figure shows a CPU 1214C in the center, version DC/DC/DC, i.e. 24 V supply voltage and 24 V load voltage for the inputs and outputs. The communication modules (a CM 1341 for RS485 connection in this figure) are attached to the left of the CPU; the signal modules (an SM 1223 with 8 x 24 V digital inputs and 8 x 24 V/0.5 A digital outputs in this figure) are plugged in to the right of the CPU. A signal board can be operated on the CPU (an SB 1232 with one analog output in this figure).

Expansion options



CM Communication module
SM Signal module
SB Signal board

Fig. 2.2 Setup and expansion options for an S7-1200 station

Table 2.2 Quantity structure of the S7-1200 CPUs

CPU	Onboard input/output channels		Expandable with SB = signal board SM = signal module CM = comm. module	Memory configuration Load memory/ work memory / retentive memory
	digital	analog		
CPU 1211C	6 DI/4 DO	2 AI/-	1 SB, 3 CM	1 MB/25 KB/2 KB
CPU 1212C	8 DI/6 DO	2 AI/-	1 SB, 2 SM, 3 CM	1 MB/25 KB/2 KB
CPU 1214C	14 DI/10 DO	2 AI/-	1 SB, 8 SM, 3 CM	2 MB/50 KB/2 KB

A two-tier design is possible using a 2-meter long extension cable. But the number of modules that can be used is not changed as a result.

Operating modes of the CPU

The CPU 1200 has the operating modes STOP, STARTUP, and RUN. In STOP, the user program is not processed, but the CPU is capable of communication and can, for example, be loaded with the user program. If the supply voltage is switched on, the CPU is first in STOP mode, then switches to STARTUP mode, in which it parameterizes the modules and passes through a user start-up routine, and after an error-free start reaches RUN mode. Now the main user program is processed. In the case of a “serious” error, the CPU switches from STARTUP or RUN mode back to STOP mode.

The modes are switched with the programming device in online operation. A mode switch is not provided.

The user memory consists of a load memory and a work memory

The user program is located on the CPU in two areas: in the load memory and work memory. The load memory contains the entire user program including configuration data; it is integrated into the CPU or available on a plug-in memory card. The work memory in the CPU is integrated fast RAM that contains the execution-relevant program code and user data.

The programming device transfers the entire user program, including configuration data, to the load memory. The operating system interprets the configuration data, and parameterizes the CPU and – during startup – the I/O modules. The execution-relevant program code and user data are copied into the work memory.

Retentivity without backup battery

Retentivity means that the contents of a memory area remain after the supply voltage is switched off and on again. With a CPU 1200, this behavior makes possible a retentive memory for bit memories and data variables and non-volatile load memory for the user program, so it does not require a backup battery. During runtime, data areas such as recipes can be read from the load memory with a user program, and other data areas such as archives can be written to the load memory.

A memory card expands the load memory

The memory card for S7-1200 is an SD Card that has been pre-formatted by Siemens. The memory card can be set as a program card or a transfer card. As a program card, the memory card replaces the integrated load memory and must be inserted during operation of the CPU. As a transfer card, the memory card allows the user program to be transferred without a programming device. It is also possible to update the CPU firmware with a transfer card.

Fig. 2.3 SIMATIC Memory Card

The signal board extends the onboard I/O

The signal board (SB) expands the onboard I/O without changing the dimensions of the CPU. The associated slot is located on the front of the CPU.

Signal boards are available with 24 V and 5 V digital inputs and outputs, which can be operated at a frequency of up to 200 kHz. The frequency of the high-speed counters (HSC) and pulse generators integrated into the CPU can thus be increased.

Voltage transmitters (± 10 V), current transmitters (0 to 20 mA), thermocouples (type J or K), or resistance thermometers (PT 100 or PT 1000) can be connected to a signal board with analog input module. The signal board with analog output module is available for ± 10 V output voltage or 0 to 20 mA output current.

Fig. 2.4 Signal Board 1223

High-speed counter

A high-speed counter (HSC) is a high-speed hardware counter in the CPU. The CPU 1211 contains three counters, the CPU 1212 has four counters, and the CPU 1214 has six counters. A high-speed counter as up/down counter has a counting range of $\pm 2^{31}$. There are special counter inputs on the CPU to capture the pulse train output; these allow a maximum frequency of 100 kHz. If a signal board with fast inputs is used, the maximum counting frequency increases to 200 kHz.

Pulse generators

A pulse generator generates pulses at a special output channel. If the output channel to the onboard I/O belongs to the CPU, the maximum pulse frequency is 100 kHz. If the output channel is on the signal board, the maximum achievable frequency increases to 200 kHz. Each CPU 1200 has two pulse generators. The pulse generators have two modes of operation: PTO (pulse train output) and PWM (pulse width modulation).

"Axis" technology object

The axis technology object controls a stepper motor or a servomotor with pulse interface. It forms the interface between the motion control instructions in the user program and the drive. Motion profiles of the drive can be created using the job table technology object.

A maximum of two axis technology objects can be set up in each CPU 1200. Each axis technology object requires a pulse generator in PTO mode, an HSC, and a rapid pulse output channel.

"PID controller" technology object

The PID controller technology object is available in two versions: as a universal controller (PID_Compact) for technical processes with continuous I/O signals and as a controller for motor-operated devices such as valves (PID_3STEP) that use digital signals to open and close.

A PID controller requires an analog input channel for the actual value and an analog output channel for the (analog) manipulated variable. Digital output channels are required if the manipulated variable is issued as a pulse width modulated signal (PID_Compact) or as a close/open signal (PID_3STEP). The PID controller technology object calculates the PID shares independently during the self-adjustment at initial start. Further optimization is possible by means of fine adjustment during operation.

Peripheral expansion with digital and analog modules

The onboard peripherals of a CPU 1212 or CPU 1214 can be expanded with two or eight signal modules (SM). Digital modules are available with 8 or 16 binary channels for 24 V input and output voltage or with relay outputs. Voltage transmitters, current transmitters, thermocouples, or resistance thermometers can be connected to an analog input module with 8 or 16 analog channels. The analog output module is available with 2 or 4 analog channels for ± 10 V output voltage or 0 to 20 mA output current. Which properties are important in the selection of I/O modules can be seen in chapters 2.10 "Process Connection with Digital Modules" on page 47 and 2.11 "Process connection with analog modules" on page 48.

Communication for S7-1200

The PROFINET interface connects a CPU 1200 with other devices via Industrial Ethernet. This can be a programming device, a Basic Panel, or another PLC. Open User Communication performs the data exchange between the programmable controllers. If only one device is connected, this can be done directly with a standard or crossover cable. The connection of multiple devices requires an interface multiplier, for example, the CSM 1277 compact switch module, to which up to three additional stations can be connected.

A CPU 1200 may be the IO Controller in a PROFINET IO system. Additional information on PROFINET IO is available in chapter 6.12 "Distributed I/O with PROFINET IO" on page 225.

The CM 1241 communication module permits a point-to-point connection based on RS232 or RS485. With a CB 1241 communication board – plugged into the front of the CPU – a point-to-point connection based on RS485 can be set up without changing the dimensions of the CPU. The following standard protocols are available: ASCII protocol, MODBUS protocol with RTU format, and USS drive protocol.

The CM 1242-5 (DP slave) and CM 1243-5 (DP master) communication modules permit the connection of a CPU 1200 to a PROFIBUS DP master system. Further information on PROFIBUS DP can be found in chapter 6.15 "Distributed I/O with PROFIBUS DP" on page 242.

Operator control and monitoring with S7-1200

The Basic Panels are the ideal operator control and monitoring devices for the S7-1200 controllers. They are connected via the PROFINET interface and configured with WinCC Basic, which is supplied with STEP 7 Basic/Professional V1x inside TIA Portal. The HMI devices for S7-1200 are described in chapter 7.2 "Basic Panels" on page 255.

Configuring and programming with STEP 7 inside TIA Portal

A CPU 1200 is configured and programmed with the STEP 7 Basic/Professional engineering software inside TIA Portal. Programming in ladder logic (LAD), function block diagram (FBD), and structured control language (SCL) is possible with version V11 of STEP 7 Basic and V2.x of the CPU firmware.

STEP 7 inside TIA Portal contains all functions for hardware configuration, networking with PROFIBUS and PROFINET, and programming and testing the user program. The engineering software is described in chapter 3.4 "Editing projects with STEP 7 inside TIA Portal" on page 70.

2.4 The SIMATIC S7-300 modular mini controller

SIMATIC S7-300 is the modular mini controller system for the lower and medium performance ranges. Possible applications include the control of packaging, textile and special-purpose machinery. An S7-300 station consists of a central controller and – as required – up to four expansion devices.

Central unit

The central controller contains the CPU and up to 8 I/O modules. The CPU requires a supply voltage of 24 V DC, which, for example, can be drawn from one of the power supplies on the mounting rail to the left of the CPU. A serial backplane bus that transfers both the I/O signals and the parameterization data connects the mod-