SIEMENS

Hans Berger

Automating with SIMATIC

Hardware and Software, Configuration and Programming, Data Communication, Operator Control and Monitoring

Sixth Edition

Berger Automating with SIMATIC

Automating with SIMATIC

Hardware and Software,
Configuration and Programming,
Data Communication, Operator Control
and Process Monitoring

by Hans Berger

6th revised and enlarged edition, 2016

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.

The author, translator and publisher have taken great care with all texts and illustrations in this book. Nevertheless, errors can never be completely avoided. The publisher, author and translator accept no liability, regardless of legal basis. Designations used in this book may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

www.publicis-books.de

Print ISBN: 978-3-89578-459-0
ePDF ISBN: 978-3-89578-943-4

6^{th} edition, 2016

Editor: Siemens Aktiengesellschaft, Berlin and Munich
Translation: Siemens AG, Business Process Shared Services, Erlangen
Publisher: Publicis Publishing, Erlangen, Germany

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 requirements are fulfilled by the SIMATIC industrial automation system, 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 V13 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 6th 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, April 2016 Hans Berger

Contents

1 **Introduction** 9

1.1 Components of the SIMATIC Automation System 9
1.2 From the Automation Task to the Finished Program 11
1.3 How Does a Programmable Logic Controller Work? 13
1.4 The path of a binary signal from the sensor to the program 15
1.5 Data management in the SIMATIC automation system 17

2 **SIMATIC Controllers – the Hardware Platform** 19

2.1 Components of a SIMATIC Station 20
2.2 The SIMATIC S7-1200 Basic Controller 21
2.3 The SIMATIC S7-1500 Advanced Controller 26
2.4 The technology functions of a CPU 1500C 33
2.5 The SIMATIC S7-300 modular mini controller 34
2.6 Technological functions of a CPU 300C 39
2.7 SIMATIC S7-400 for demanding tasks 40
2.8 High Availability with SIMATIC 46
2.9 Safety Integrated with SIMATIC S7 49
2.10 Use Under Difficult Conditions: SIPLUS 52
2.11 Process Connection with Digital Modules 53
2.12 Process connection with analog modules 56
2.13 FM/TM modules relieve the CPU 58
2.14 Bus connection with communication modules 60
2.15 Distributed controllers 62
2.16 SIMATIC PC-based Automation 63
2.17 ET 200 distributed I/O system 70
2.18 The SIMATIC programming device 76

3 **STEP 7: Engineering Tool for SIMATIC** 77

3.1 Overview of STEP 7 variants 77
3.2 Automating with STEP 7 79
3.3 Editing projects with STEP 7 inside TIA Portal 80
3.4 Editing projects with STEP 7 V5.5 85
3.5 Configuring a SIMATIC station 89
3.6 Tools for programming 93
3.7 Giving the addresses a name 94
3.8 Programming a logic block 96

3.9 Programming a data block 100
3.10 Programming a user-defined data type 103
3.11 Working with program source files 105
3.12 Help on Program Creation 110
3.13 Downloading the user program to the CPU 112
3.14 Processing the user program online 115
3.15 Controlling the user program with online tools 119
3.16 Finding hardware faults using diagnostic functions 122
3.17 Testing with watch tables 124
3.18 Testing the program with the program status 127
3.19 Testing user programs offline using S7-PLCSIM 131
3.20 Documenting project data 135

4 The programming languages 139

4.1 Ladder Logic LAD 141
4.2 Function Block Diagram FBD 145
4.3 Statement List STL 149
4.4 Structured Control Language SCL 153
4.5 S7-GRAPH sequence control 156
4.6 The function library of LAD, FBD, and STL 159
4.7 The function library of SCL 161
4.8 Global address areas 163
4.9 Absolute and symbolic addressing 169
4.10 Indirect addressing 172
4.11 Elementary data types 174
4.12 Structured data types 178
4.13 Data types for block parameters 180
4.14 Further data types 182

5 The user program 184

5.1 Program execution with SIMATIC 184
5.2 The start-up routine 187
5.3 The main program 189
5.4 The process images 191
5.5 Cycle Time, Reaction Time 193
5.6 Program functions 196
5.7 Time-of-day interrupts 201
5.8 Time-Delay Interrupts 202
5.9 Cyclic Interrupts 204
5.10 Hardware Interrupts 205
5.11 Multiprocessor Interrupt 207
5.12 Synchronous errors 208
5.13 Asynchronous errors 210

5.14 Local error handling 213
5.15 Diagnostic functions in the user program 214
5.16 Overview of user blocks 219
5.17 Block properties 220
5.18 Know-how protection, copy protection 223
5.19 Block interface 225
5.20 Calling blocks 228

6 Communication 232
6.1 Configuring the network 232
6.2 The MPI subnet 237
6.3 External station S7 basic communication 238
6.4 Global data communications 240
6.5 The Industrial Ethernet subnet 241
6.6 Open User Communication, IE communication 242
6.7 S7 communication 245
6.8 The PROFIBUS subnet 247
6.9 Internal station S7 basic communication 249
6.10 The AS-Interface subnet 250
6.11 The point-to-point connection 251
6.12 Distributed I/O with PROFINET IO 252
6.13 Real-time communication with PROFINET IO 261
6.14 Special functions for PROFINET IO 266
6.15 Distributed I/O with PROFIBUS DP 267
6.16 Special functions for PROFIBUS DP 273
6.17 Isochronous mode program 275
6.18 DPV1 interrupts 279

7 Operator control and monitoring 282
7.1 Key Panels 283
7.2 Basic Panels 284
7.3 Comfort Panels 286
7.4 Mobile Panels 288
7.5 SIMATIC Panel PC 291
7.6 Configuring SIMATIC HMI 293
7.7 Remote maintenance with TeleService 301

Index 303
Abbreviations 306

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 (Fig. 1.1).

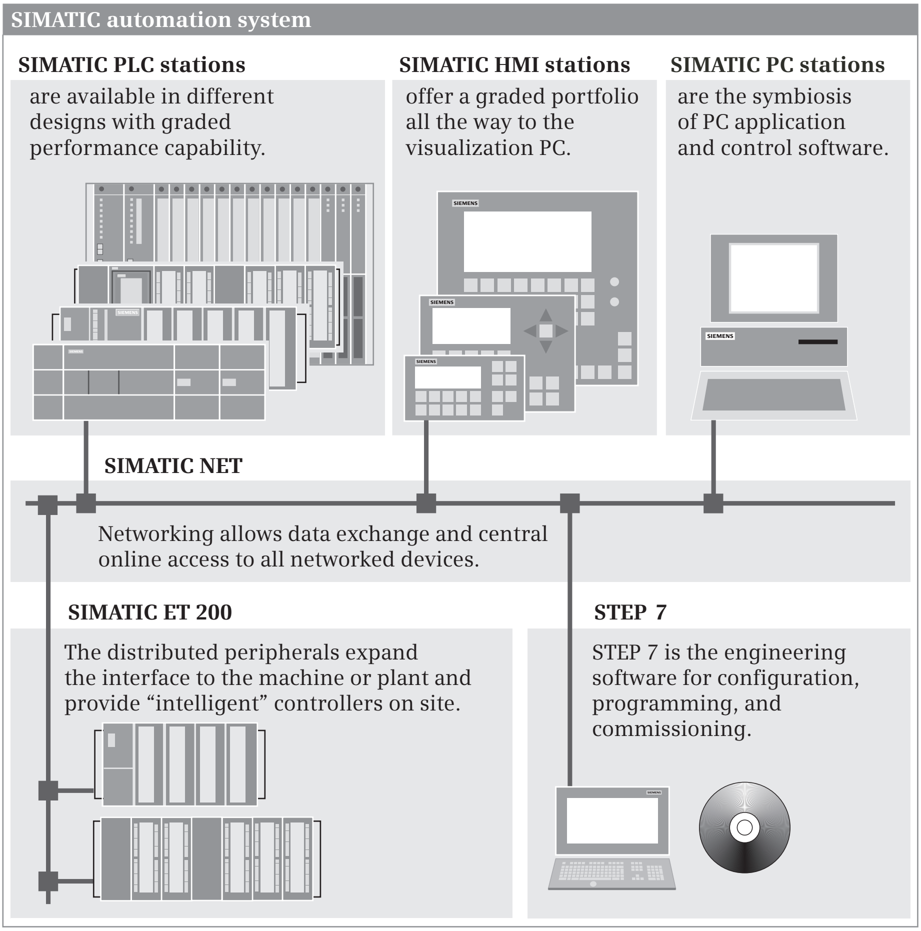

SIMATIC automation system
SIMATIC PLC stations
are available in different designs with graded performance capability.
SIMATIC HMI stations
offer a graded portfolio all the way to the visualization PC.
SIMATIC PC stations
are the symbiosis of PC application and control software.
SIEMENS
SIMATIC NET
Networking allows data exchange and central online access to all networked devices.
SIMATIC ET 200
The distributed peripherals expand the interface to the machine or plant and provide “intelligent” controllers on site.
STEP 7
STEP 7 is the engineering software for configuration, programming, and commissioning.

Fig. 1.1 Components of the SIMATIC automation system

As programmable logic controllers (PLCs), the **SIMATIC S7 Controllers** form the basis of the automation system. SIMATIC S7-1200 Basic Controllers are the ideal choice for simply and autonomously resolving tasks in the low to medium performance range. The SIMATIC S7-1500 Advanced Controllers ensure maximum performance capability and maximum user friendliness for medium and high-end applications in machine and plant automation. The characteristics of the SIMATIC S7-300 controller are especially aimed at innovative system solutions in the manufacturing industry – e.g. in the automotive or packaging industry. Within SIMATIC, the S7-400 Controllers are designed for system solutions in factory and process automation and are especially suitable for data-intensive tasks, for example in the process industry. All controllers can be modularly expanded with signal, technology and communication modules.

SIMATIC ET 200 expands the interface between the central controller and the machine or system by I/O modules directly on-site. If autonomously functioning, “intelligent” controllers are needed at the field level, the **Distributed Controllers ET 200** CPU are used. SIMATIC ET 200 provides a multi-functional, modular and finely scalable system for the distributed automation: For solutions in the control cabinet or without a control cabinet directly at the machine and for use in hazardous areas. All of the products can be integrated into the automation via the PROFIBUS or PROFINET bus systems.

SIMATIC PC-based automation is the ideal basis to efficiently and economically implement the combination of Windows applications and SIMATIC control software. The industrial PCs of the **SIMATIC IPC** product family provide maximum performance with the latest Intel processor technology, pre-installed and enabled Windows operating system, and integrated communication interfaces.

The **software controllers** with the scope of services of the S7-1500 family of controllers are especially well-suited for the flexible control of special-purpose machines with high performance and functional requirements.

SIMATIC HMI stands for Human Machine Interface. The SIMATIC Panels are available in numerous different performance classes and permit the efficient, machine-level control of the machine or plant. As ideal basis for HMI solutions on PCs, Panel PC systems offer a comprehensive and finely graded portfolio. Powerful HMI 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.

STEP 7 is the engineering software which is used to configure, parameterize, and program all SIMATIC components. The “classic” version of STEP 7 with the SIMATIC Manager or the innovated version of STEP 7 inside TIA Portal 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

If the machine to be controlled is a small one, will an S7-1200 be big enough or do you need an S7-1500? 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 S7-1200, S7-1500, 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 (one device), either a programmable logic control (PLC) station, a human-machine interface (HMI) station, or a personal computer (PC) 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 tag 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 a successful commissioning, you can document the project as a printable circuit manual and archive it as a zipped file.

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 (Fig. 1.2).

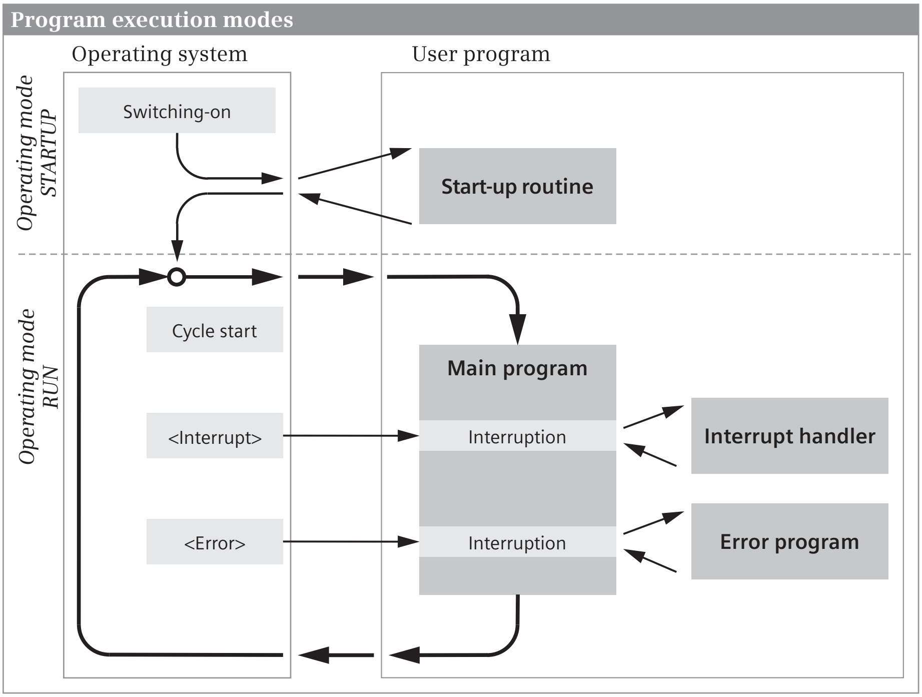


Fig. 1.2 Execution of the user program in a SIMATIC controller

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”.

After the power is switched on and the operating function test runs, the operating system starts an (optional) start-up routine once. 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 are available to store user data.

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

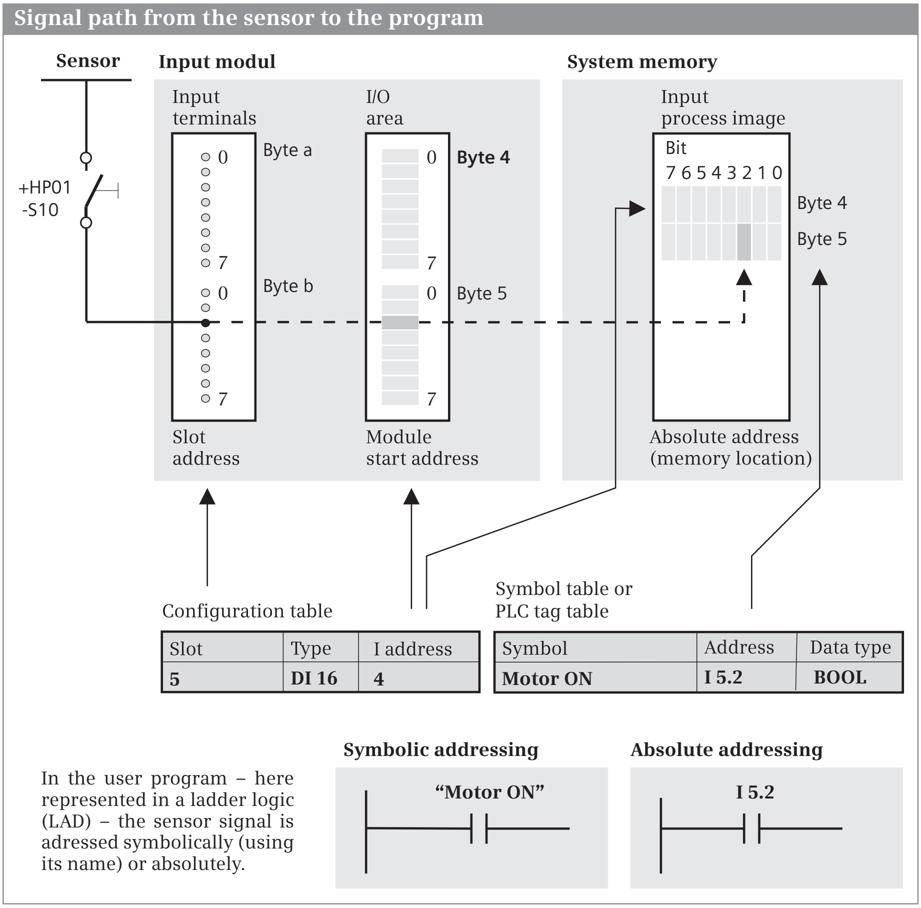
In order to do its job, the control processor in the controller needs to be connected to the machine or plant it is controlling. I/O modules that are wired to the sensors and actuators create this connection.

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 I/O address. In the I/O 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 I/O address consecutively, which you can change in the configuration table. In the connection diagram, the module start address is identified with “Byte a”. If a module has more bytes, the byte addresses are automatical-



Configuration table

Slot	Type	I address
5	**DI 16**	**4**

Symbol table or PLC tag table

Symbol	Address	Data type
Motor ON	**I 5.2**	**BOOL**

In the user program – here represented in a ladder logic (LAD) – the sensor signal is adressed symbolically (using its name) or absolutely.

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

ly incremented and are assigned the designations "Byte b", "Byte c", and so on. The "Motor ON" signal is connected to terminal two of the second byte (Byte b). 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 tag 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).

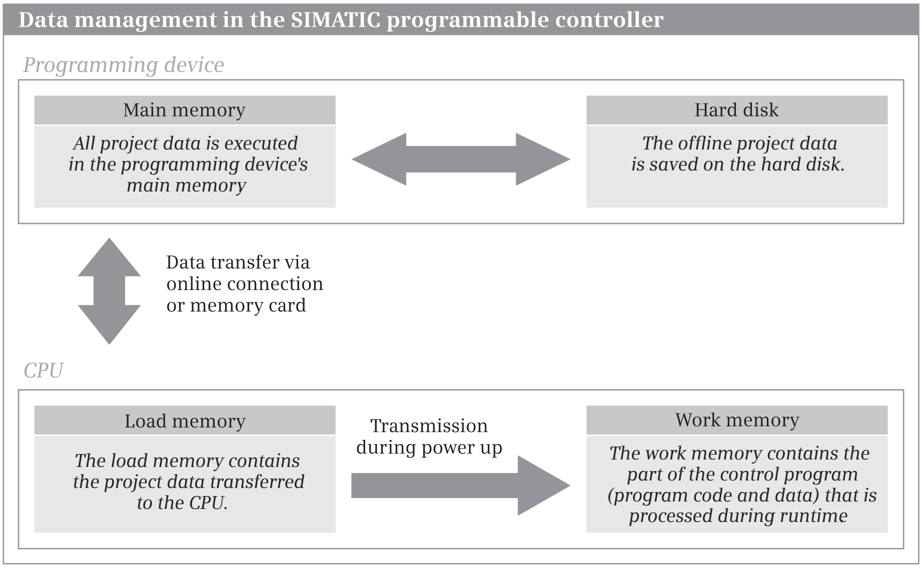

Data management in the SIMATIC programmable controller
Programming device
Main memory
All project data is executed in the programming device's main memory
Hard disk
The offline project data is saved on the hard disk.
Data transfer via online connection or memory card
CPU
Load memory
The load memory contains the project data transferred to the CPU.
Transmission during power up
Work memory
The work memory contains the part of the control program (program code and data) that is processed during runtime

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 or CPU 1500, the load memory is on the memory card, which therefore must always be inserted in the CPU for use. The memory card for these CPUs 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).

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

- SIMATIC S7-1200 for the lower and medium performance ranges,
- SIMATIC S7-1500 for the medium and upper performance ranges,
- SIMATIC S7-300 for the medium performance range, and
- SIMATIC S7-400 for the upper performance range.

An S7-1200/1500 station is comprised of a single-tier rack with the CPU and the I/O modules, which establish the connection to the controlled machine or system. For an S7-300/400 station, the rack with the CPU and the I/O modules (the central controller) can be supplemented by racks with additional I/O modules (expansion devices).

SIMATIC ET 200 are modules installed on site at the machine or in the plant and are connected to the master station via PROFINET IO and/or PROFIBUS DP. Many SIMATIC CPUs feature an integrated PROFINET or PROFIBUS interface, which greatly facilitates the connection of distributed I/O. Since operation on the PROFINET or PROFIBUS is standardized independent of the vendor, it is also possible to connect third-party devices to a SIMATIC controller. For ET 200SP, ET 200S and ET 200pro, there are also interface modules with CPU functionality (Distributed Controller), which permit a distributed stand-alone solution. Furthermore, a software controller is available for ET 200SP (Open Controller).

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.
- The S7-1500S software controllers are PC applications with a control program and engineering that are fully compatible with an S7-1500 standard 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 as 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. For S7-300, S7-1200 and S7-1500, this is a mounting rail; its length is determined by the number and width of the modules used. The electrical connection of the modules to one another is realized by bus connectors on the rear 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; controls the central and distributed I/O modules; can also be a DP slave on PROFIBUS DP or an IO Device on PROFINET IO.

- ▷ Interface modules (IM)
  These connect the racks with each other for S7-300 and S7-400.

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

- ▷ Communication modules (CM), 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. They are integrated in the address space of the centralized I/O system and are principally treated just like the I/O modules installed locally in the central and expansion racks.

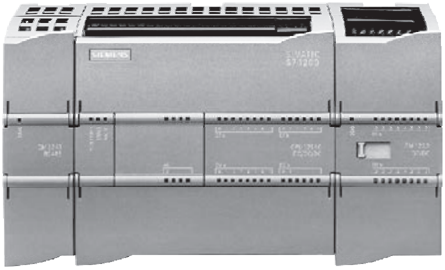
2.2 The SIMATIC S7-1200 Basic Controller

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 (Fig. 2.1). 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

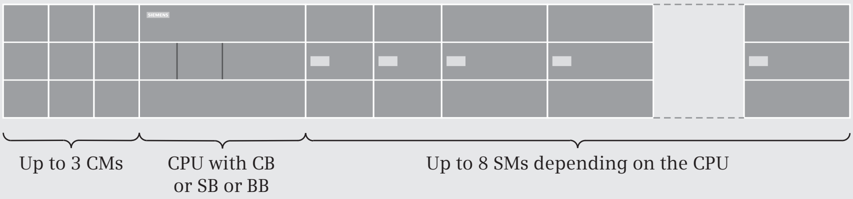
Five CPUs with different performance capability in the versions DC/DC/DC, DC/DC/relay, or AC/DC/relay are offered. The first specification refers to the supply voltage (24 V DC, 85 to 264 V AC), the second to the signal voltage of the digital inputs (24 V DC), and the third to the type of digital outputs (24 V DC electronic or relay outputs 5 to 30 V DC, 5 to 250 V AC). Table 2.1 shows the expandability and the memory configuration. Rapid counters with counting frequencies of up to 100/200 kHz (for CPU 1217 up to 1 MHz) are integrated with the central processing unit, which in connection with 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 (CM 1341 for RS485 connection in this figure) are attached to the left of the CPU; the signal modules (SM 1223 with 8 × 24 V digital inputs and 8 × 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	CB	Communication Board
SM	Signal Module	SB	Signal Board
		BB	Battery Board

Fig. 2.1 Setup and expansion options for an S7-1200 station

Table 2.1 Selected data of a CPU 1500 with Firmware V4.1

	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
User memory					
internal load memory *)	1 MB	1 MB	4 MB	4 MB	4 MB
Work memory	30 KB	75 KB	100 KB	100 KB	150 KB
Retentivity memory	10 KB	10 KB	10 KB	10 KB	10 KB
Onboard I/O					
Digital inputs	6 DI, 24V DC	8 DI, 24V DC	14 DI, 24V DC	14 DI, 24V DC	14 DI, 24V DC
Digital outputs	4 DO, 24V DC or relay	6 DO, 24V DC or relay	10 DO, 24V DC or relay	10 DO, 24V DC or relay	10 DO, 24V DC or relay
Analog inputs	2AI (10 bit)	2AI (10 bit)	2 AI (10 bit)	2 AI (10 bit)	2 AI (10 bit)
Analog outputs	–	–	–	2 AO (10 bit)	2 AO (10 bit)
Expansion with					
a board (SB, CB, BB)	1	1	1	1	1
Signal modules (SM)	–	2	8	8	8
Communication modules (CM)	3	3	3	3	3
Operands					
Inputs (Byte)	1024	1024	1024	1024	1024
Outputs (Byte)	1024	1024	1024	1024	1024
Bit memories (Byte)	4096	4096	8192	8192	8192
max. Block number	1024	1024	1024	1024	1024
max. Block size	30 KB	50 KB	64 KB	64 KB	64 KB
PROFINET Connection	1	1	1	2 with Switch	2 with Switch
Execution times					
for binary operations	0.085 µs	0.085 µs	0.085 µs	0.085 µs	0.085 µs
for word operations	1.7 µs	1.7 µs	1.7 µs	1.7 µs	1.7 µs
for fixed-point arithmetic	1.7 µs	1.7 µs	1.7 µs	1.7 µs	1.7 µs
for floating-point arithmetic	2.3 µs	2.3 µs	2.3 µs	2.3 µs	2.3 µs

*) Expandable up to SD memory size (maximum 2 GB)

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.

FailSafe CPU S7-1200F

The CPU 1214F and CPU 1215F central processing units allow you to set up a fail-safe automation system for plants with increased safety requirements. In a fail-safe 1200 station, both the standard and fail-safe I/O modules can be operated.

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 selector 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 during startup and parameterizes the modules. The execution-relevant program code and user data are copied into the work 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.

SIMATIC Memory Cards are available in the sizes 4 MB, 12 MB, 24 MB, and 2 GB.

Fig. 2.2 SIMATIC Memory Card

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.

Plug-in boards extend the onboard I/O

An installation slot on the front side of the CPU permits the expansion of the onboard I/O without changing the dimensions of the CPU. Signal boards, a communication board, and a battery board are available.

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.

The CB 1241 communication board takes over the serial data exchange via a point-to-point connection according to RS 485.

The BB 1297 battery board, the buffered runtime of the real-time clock can be extended from a typical 10 days to up to one year.

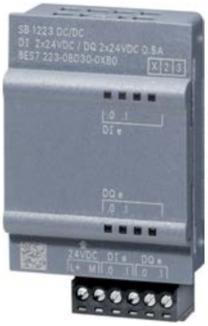


Fig. 2.3
Signal Board 1223

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 a CPU 1214, 1215 or 1217 have 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. With the differential inputs of the CPU 1217, the counting frequency can be increased up to 1 MHz.

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, for a CPU 1217 a maximum of 1 MHz. If the output channel is on the signal board, a maximum frequency of 200 kHz can be reached. One CPU 1200 has four pulse generators. The pulse generators have two modes of operation: PTO (pulse train output) and PWM (pulse width modulation).

Technology objects for motion control

The technology object *TO_PositioningAxis* controls a stepper motor or a servomotor with pulse interface. Motion profiles of the drive can be created using the technology object *TO_CommandTable*.

In a CPU 1211, 1212 and 1214, a maximum of two technology objects for motion control can be set up, in a CPU 1215 and 1217 a maximum of four. Each technology object *TO_PositioningAxis* requires a pulse generator in PTO mode.

Technology objects for PID control

For a PID controller there are three technology objects: *PID_Compact* as universal controller for technical processes with continuous input/output signals, *PID_3Step* as step controller with 3-point mode for motor-operated devices such as valves, which use digital signals to open and close, and *PID_Temp* as universal temperature controller.

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 or as a close/open signal. The technology objects for PID_Control calculate the PID shares independently during the self-adjustment at initial start.

Peripheral expansion with digital and analog modules

The onboard peripherals of a CPU 1212 can be expanded with two, the CPU 1214, 1215 or 1217 with 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 2.11 “Process Connection with Digital Modules” on page 53 and 2.12 “Process connection with analog modules” on page 56.

Communication for S7-1200

The PROFINET interface connects a CPU 1200 with other devices via Industrial Ethernet. This can be a programming device, an HMI device, or another PLC. Open User Communication performs the data exchange between the programmable controllers. The interface automatically detects the transfer rate of 10 or 100 Mbit/s (autosensing) and accepts both a standard Ethernet cable and a “cross-over” cable.

If the PROFINET interface only has a single port, the connection of several devices requires an interface multiplier, e.g. the CSM 1277 compact switch module, to which up to three additional stations can be connected. For a CPU 1215 or CPU 1217, the interface has two connections which are connected to a switch and thus permits the setup of a linear structure without additional devices.

A CPU 1200 may be the IO Controller or an IO device in a PROFINET IO system. Additional information on PROFINET IO is available in chapter 6.12 “Distributed I/O with PROFINET IO” on page 252.

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 267.

The CM 1243-2 communication module can control up to 62 AS-Interface slaves as AS-Interface master (see chapter 6.10 “The AS-Interface subnet” on page 250).

The communication modules CP 1242-7 (GSM network with GPRS) and CP 1243-1 (Internet) permit the connection of an S7-1200 station to the industrial telecommunication via the public infrastructure.

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. As of STEP 7 V13, the CPU firmware V4.0 is supported. Programming in ladder logic (LAD), function block diagram (FBD), and structured control language (SCL) is possible.

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 3.3 “Editing projects with STEP 7 inside TIA Portal” on page 80.

The PLCSIM optional software is available for testing the user program without programmable controller (see chapter 3.19 “Testing user programs offline using S7-PLCSIM” on page 131).

2.3 The SIMATIC S7-1500 Advanced Controller

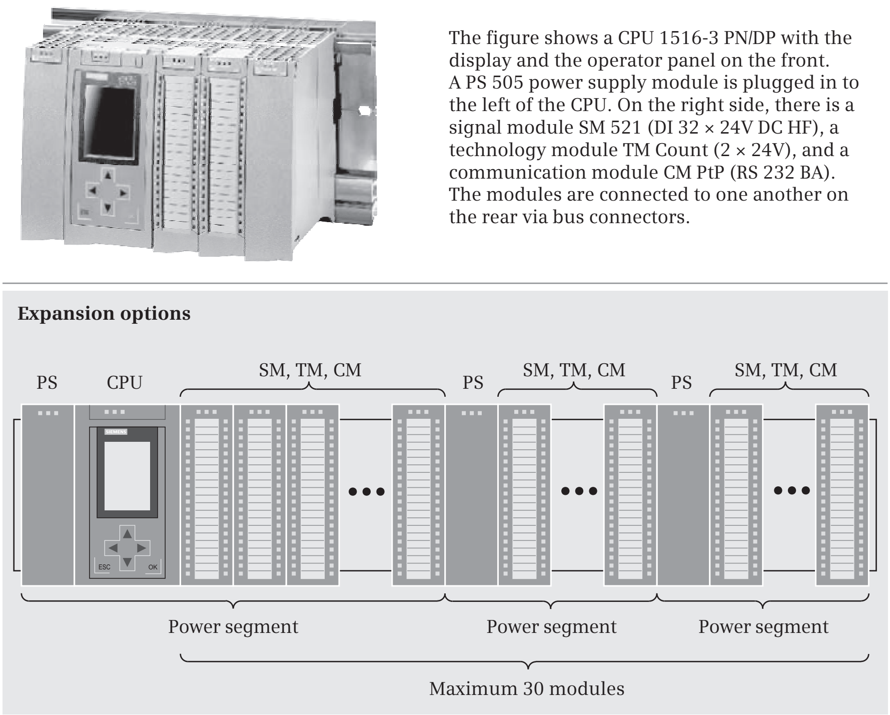
SIMATIC S7-1500 is a modular, scalable automation system that can be universally used in the manufacturing industry. An S7-1500 station comprises one rack with a CPU and, as required, several signal, technology and communication modules (Fig. 2.4). Distributed I/O stations can additionally be connected via the PROFINET IO and PROFIBUS DP bus systems. CPU 1515, CPU 1516 and CPU 1517 have one, CPU 1518 has two additional connections to Industrial Ethernet with separate IP address for network separation. S7-1500 is configured and programmed using STEP 7 Professional inside TIA Portal.

S7-1500 station

An S7-1500 station comprises a maximum of 32 slots. Each module occupies one slot independent of its width. Slot 1 is occupied by the CPU. It can be connected to 24 V direct voltage and then supply the system voltage to the modules that are plugged in to the right of it. If the CPU cannot supply the required power, a system power supply is plugged into slot 0, which then supplies voltage to the CPU and the modules plugged in to the right of it.

The combination of a PS module or CPU and the modules to be supplied are called a “power segment”. Up to three power segments can be set up in an S7-1500 station. The number of modules a power segment encompasses depends on the elec-

Setup of an S7-1500 station



The figure shows a CPU 1516-3 PN/DP with the display and the operator panel on the front. A PS 505 power supply module is plugged in to the left of the CPU. On the right side, there is a signal module SM 521 (DI 32 × 24V DC HF), a technology module TM Count (2 × 24V), and a communication module CM PtP (RS 232 BA). The modules are connected to one another on the rear via bus connectors.

Expansion options

The rack has 32 slots. An optional power supply occupies slot 0 and the CPU occupies slot 1. To the right of the CPU, there is room for up to 30 modules (including power supply modules).

In a power segment, a PS module provides the internal system voltage for the modules plugged in on the right.

Up to three power segments are permitted.

PS	Power Supply
CPU	Central Processor Unit
SM	Signal Module
TM	Technology Module
CM	Communication Module

Fig. 2.4 Setup and expansion options for an S7-1500 station

trical power that is provided and consumed. An additional load power supply is needed for supplying the sensors and actuators.

The rack is a mounting rail, which only needs to be long enough to accommodate the existing modules. The power supply for the module electronics and the data exchange between the modules is accomplished via the backplane bus. The backplane bus is made up of “U-connectors” between the modules, which must be inserted without gaps. One U-type-connector is needed for each module.

CPU

The front of the CPU module is comprised of a collapsible and removable control panel with a color display and control keys. The color display shows – structured in several menus – the status and properties of the CPU, diagnostics alarms, the date/time, and information about the connected modules.

The control keys are designed as membrane keyboard. These can be used to select the menus in the display and to set the date, time, access protection, language, and IP address. The memory of the CPU can also be reset to the factory settings.

The mode selector and the bus connections are located under the control panel; their type and number depends on the CPU version. Each CPU 1500 has at least one PROFINET interface with two ports, which are connected by a switch diagnostics alarms, the date/time, and information about the connected modules.

Table 2.2 Selected data of a CPU 1500 with Firmware V1.8

CPU	1511-1 PN	1513-1 PN	1515-2 PN	1516-3 PN/DP	1517-3 PN/DP	1518-4 PN/DP
User memory						
Work memory						
for program	150 KB	300 KB	500 KB	1 MB	2 MB	4 MB
for data	1 MB	1,5 MB	3 MB	5 MB	8 MB	20 MB
Retentive memory	88 KB	88 KB	472 KB	472 KB	700 KB	700 KB
Load memory *)						
Operands						
Inputs	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB
Outputs	32 KB	32 KB	32 KB	32 KB	32 KB	32 KB
Bit memories	16 KB	16 KB	16 KB	16 KB	16 KB	16 KB
SIMATIC timers	2048	2048	2048	2048	2048	2048
SIMATIC counters	2048	2048	2048	2048	2048	2048
Max. block size						
Logic block	150 KB	300 KB	500 KB	512 KB	512 KB	512 KB
Data block ***)	1 MB	1,5 MB	3 MB	5 MB	8 MB	16 MB
Max. number **)	2000	2000	6000	6000	10 000	10 000
Interfaces						
PROFINET (2 ports)	1	1	1	1	1	1
Industrial Ethernet	–	–	1	1	1	2
PROFIBUS DP master	–	–	–	1	1	1
Number of CMs (PROFINET + PROFIBUS)	4	6	8	8	8	8
Connections						
maximum	96	128	192	256	320	384
reserved for PG, HMI, and Web server	10	10	10	10	10	10
via integrated interfaces	64	88	108	128	160	192
Execution times						
Binary operation	60 ns	40 ns	30 ns	10 ns	2 ns	1 ns
Word operation	72 ns	48 ns	36 ns	12 ns	3 ns	2 ns
Fixed point arithmetic	96 ns	64 ns	48 ns	16 ns	3 ns	2 ns
Floating point arithmetic	384 ns	256 ns	192 ns	64 ns	12 ns	6 ns

*) on the memory card up to SD card size (max. 32 GB)
**) global objects (logic and data blocks, user data types, global constants, etc.)
***) with standard access: 64 KB

Versatile application

Six **standard controllers CPU 15xx** with graded performance capability are available for a wide variety of different applications. Table 2.2 shows selected data of the different versions.

All of the standard CPUs are also available in a fail-safe version. The **failsafe CPUs 15xxF** allow you to set up a fail-safe automation system for plants with increased safety requirements. In a fail-safe 1500 station, both the standard and fail-safe I/O modules can be operated.

The **compact CPUs 15xxC** contain onboard I/O to be able to use the technology functions (count, control, position detection for motion control) without additional modules and thus allow a compact layout of mini-controllers (see Chapter 2.5 "The technology functions of a CPU 1500C" on page 40).

With the **distributed controllers CPU 15xxSP** – and **CPU 15xxSP F** in the fail-safe version – a distributed ET200SP station can be an I-Device on the PROFINET IO or an I-Slave on the PROFIBUS DP.

The **software controllers** are PC applications, which run under the Windows operating system. A **CPU 15xxS** runs on an industrial PC, a **CPU 15xxSP PC** has the design of an ET 200SP station and can control this as an open controller with PC power.

Operating modes of a CPU 1500

A CPU 1500 has the operating modes STOP, STARTUP, and RUN. After switching on the power supply, the CPU is in STOP mode. The user program is not processed, but the CPU is still capable of communication, i.e. the user program can be loaded or the diagnostic buffer can be read, for example.

The operating state can be changed on the CPU via the display, with the mode selector, or with a programming device in online mode. If the RUN mode is activated, the STARTUP mode is executed, in which the modules are parameterized and a user start-up routine is executed. The main user program is executed in RUN mode.

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 in the work memory. The load memory contains the entire user program including configuration data; it is designed as plug-in SIMATIC Memory Card. The work memory is fast RAM that is integrated in the CPU and 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 during startup and parameterizes the modules. The execution-relevant program code and user data are copied into the work memory.