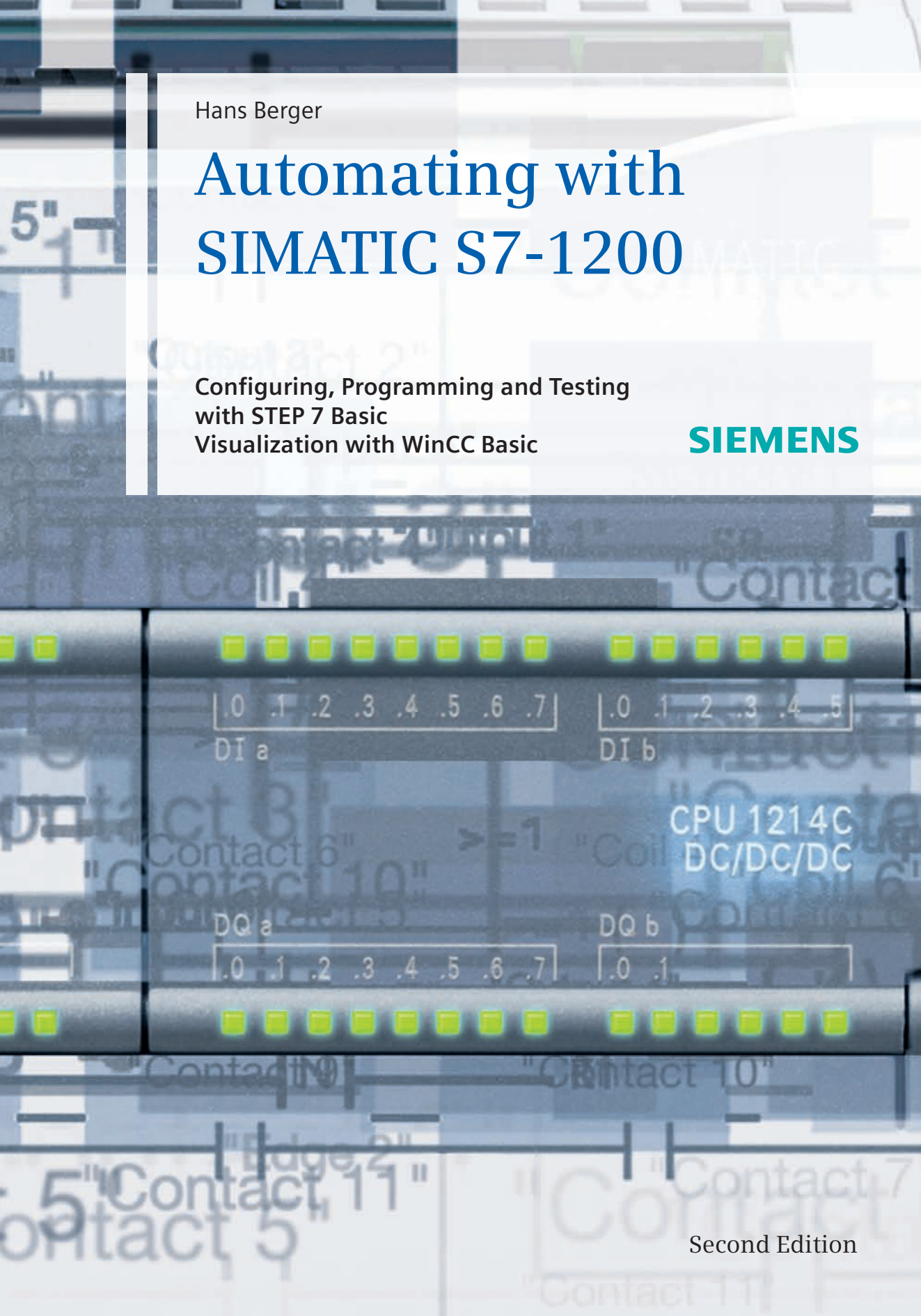
Hans Berger

Automating with SIMATIC S7-1200

Configuring, Programming and Testing with STEP 7 Basic
Visualization with WinCC Basic

SIEMENS



Second Edition

Berger Automating with SIMATIC S7-1200

Automating with SIMATIC S7-1200

Configuring, Programming and
Testing with STEP 7 Basic
Visualization with HMI Basic

by Hans Berger

2nd enlarged and revised 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 on the Internet at http://dnb.d-nb.de.

www.publicis-books.de

Print ISBN: 978-3-89578-385-2
ePDF ISBN: 978-3-89578-901-4

2nd edition, 2013

Editor: Siemens Aktiengesellschaft, Berlin and Munich
Publisher: Publicis Publishing, Erlangen

Printed in Germany

Preface

The SIMATIC automation system unites all the subsystems of an automation solution under a uniform system architecture to form a homogenous whole from the field level right up to process control.

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 automation system which provides uniformity for configuration, programming, data management and communication.

This book describes the newly developed SIMATIC S7-1200 automation system. The S7-1200 programmable controllers are of compact design and allow modular expansion. Many small applications can be solved using the CPU module with on-board I/O. The technological functions integrated in the CPU module mean that extremely versatile use of the device is possible. Two established programming languages are available for solving automation tasks: ladder logic (LAD) and function block diagram (FBD).

New SIMATIC HMI Basic Panels have been designed for operator control and monitoring appropriate to the S7-1200 programmable controllers, and provide a performance and functionality optimized for small applications. A touch screen with various monitor sizes and coordinated communication over Industrial Ethernet are ideal prerequisites for interaction with S7-1200.

The STEP 7 Basic engineering software makes it possible to use all S7-1200 controller options. STEP 7 Basic is the common tool for hardware configuration, generation of the control program, and for debugging and diagnostics. The SIMATIC WinCC Basic configuration software included in STEP 7 Basic is used to configure the Basic Panels. Modern and intuitive user guidance allows efficient and task-oriented engineering of control and visualization devices.

This book describes the S7-1200 automation system with S7-1200 programmable controllers and HMI Basic Panels. The description focuses on the generation of the control program using STEP 7 Basic engineering software Version 11 SP2.

Nuremberg, February 2013 Hans Berger

The contents of the book at a glance

Start

Introduction

SIMATIC S7-1200: Overview of the SIMATIC S7-1200 automation system.
STEP 7 Basic: Introduction to the engineering software for SIMATIC S7-1200.
SIMATIC project: Basic functions for the automation solution.

Devices & networks

The hardware components of S7-1200

Modules: Overview of the SIMATIC S7-1200 modules.

Device configuration

Hardware configuration: Configuration of the hardware design.
Network configuration: Configuration of a communication network.

PLC programming

The control program

Operating modes: How the CPU module responds with STARTUP, RUN and STOP.
Processing modes: Restart characteristics, main program, interrupt processing, and error handling define the processing of the control program.
Blocks: Organization blocks, function blocks, functions, and data blocks structure the control program.

The program editor

Programming: How the control program is produced.
Program information: Tools for supporting programming.

Ladder logic and function block diagram as programming languages

Program elements: The characteristics of LAD and FBD programming; the use of contacts, coils, standard boxes, Q boxes and EN/ENO boxes.

Tags and data types

Tags: Operand areas, project-wide and block-local tags, addressing.
Data types: Description of elementary and compound data types.

Description of the control functions

Basic functions: Binary operations, memory functions, edge evaluation, timer and counter functions.
Digital functions: Move, comparison, arithmetic, math, conversion, shift, and logic functions.
Program flow control: Jump functions, block end function, block calls.

Online & diagnostics

Connection of programming device to PLC station

Online operation: Establish connection to PLC station.
Status LEDs: The modules signal an error.
Diagnostics information: Find the error using the diagnostics information.
Online tools: Control the CPU module using the online tools.

Online & offline project data

Download: Download control program into CPU memory.
Blocks: Edit and compare the blocks offline/online.
Test: Test the control function using program status and monitoring tables.

Data communication

Open user communication

Data transmission: Data exchange from PLC to PLC over Ethernet.

Point-to-point connection

PtP: Data transmission with CM modules via RS232 and RS485.

Visualization

Configuration of Basic Panels

Introduction: Overview of Basic Panels.
Start: Create an HMI project, the HMI device wizard.
Connection to the PLC: Create HMI tags and area pointers.
Create screens: Configuration of process screens – templates, layers and screen changeover.
Working with image elements: Arrange and edit operator control and display elements, configure a message system, create recipes, transfer data records, configure user management.

Complete the HMI program

Simulation: Simulate the HMI program with PLC station or with tag table.
Connection: Transfer the HMI program to the HMI station.

Appendix

Integral and technological functions

Functions: High-speed counter, pulse generator, motion control, PID controller.

Global libraries

Overview: USS drive control, MODBUS blocks.

Table of contents

1 Introduction

1.1 Overview of the S7-1200 automation system

The SIMATIC S7-1200 automation system consists of the four controllers S7-1211C, S7-1212C, S7-1214C, and S7-1215C, which can exchange data with each other, with SIMATIC HMI Basic Panels, or with other programmable controllers over SIMATIC NET. STEP 7 (TIA Portal) is used to configure and program the devices (Fig. 1.1).

The **SIMATIC S7-1200 controllers** are programmable logic controllers (PLC) and constitute the basis of the automation system. Four different controllers with graded performances cover the low-end range of industrial controls.

SIMATIC HMI refers to the Human Machine Interface for operator control and monitoring. The Basic Panels are designed such that they interact optimally with SIMATIC S7-1200. The devices are available with display dimensions of 3.8, 5.7, 10.4 and 15 inches, and are operated using the touch screen. Except for the 15-inch device, they have additional function keys.

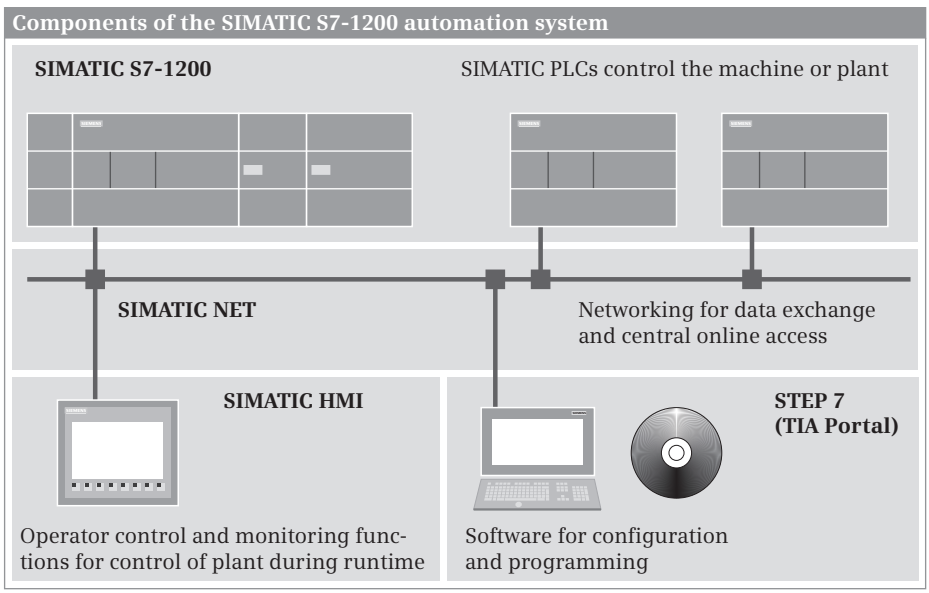


Fig. 1.1 Components of the SIMATIC S7-1200 automation system

SIMATIC NET links all SIMATIC stations, and allows trouble-free data exchange. SIMATIC S7-1200 with PROFINET interface uses the Industrial Ethernet network to exchange data with other PLC stations, HMI stations, and programming devices. Communication modules expand the communication capabilities to other networks such as PROFIBUS DP, AS-Interface, or point-to-point coupling based on RS232 or RS485.

The **STEP 7** programming software provides the nesting function for Totally Integrated Automation (TIA), the automation system with uniform configuration and programming, data management, and data transfer. STEP 7 is used to configure and parameterize the SIMATIC components, and STEP 7 is also used to generate and debug the user program. The *TIA Portal* is the central user interface for management of the tools and automation data. STEP 7 in the TIA Portal is available in the versions STEP 7 Professional and STEP 7 Basic. Both versions can be used to configure and program an S7-1200 station. This book describes the use of STEP 7 Basic.

1.1.1 SIMATIC S7-1200

SIMATIC S7-1200 is the modular microsystem for the lower and medium performance range. The central processing unit (**CPU**) contains the operating system and the user program. The user program is located in the load memory and is power failure-proof. The parts of the user program relevant to execution are processed in a work memory with fast access. Tags whose values are to be retained in the event of a power failure or when switching off/on are stored in the retentive memory (Fig. 1.2).

The user program can be transferred to the CPU using a plug-in memory card **(MC)** – as an alternative to transfer via an online connection to the programming device. The memory card can also be used as an external load memory or for updating the firmware.

The connections to the plant or process are made by **onboard inputs and outputs**, their number being determined by the CPU version. The onboard inputs and outputs are designed especially for operation of the integral high-speed counters (HSC). The operating system additionally includes pulse generators with a pulse-width modulated output and also the technology objects *Axis* for controlling stepper motors and servo motors with pulse interface and *PID Compact*, a PID controller with optimized self-tuning.

A signal board (SB) can be used to expand the onboard inputs and outputs. The communication board (CB) creates a point-to-point connection for the CPU and the battery board (BB) increases the power reserve of the integrated hardware clock to about one year.

If further inputs and outputs are required, signal modules (**SM**) can be plugged onto the CPU depending on its version. These are available for digital and analog signals.

The **PROFINET interface** connects the CPU to the Industrial Ethernet subnet. The programming device is connected to this interface if, for example, the user pro-

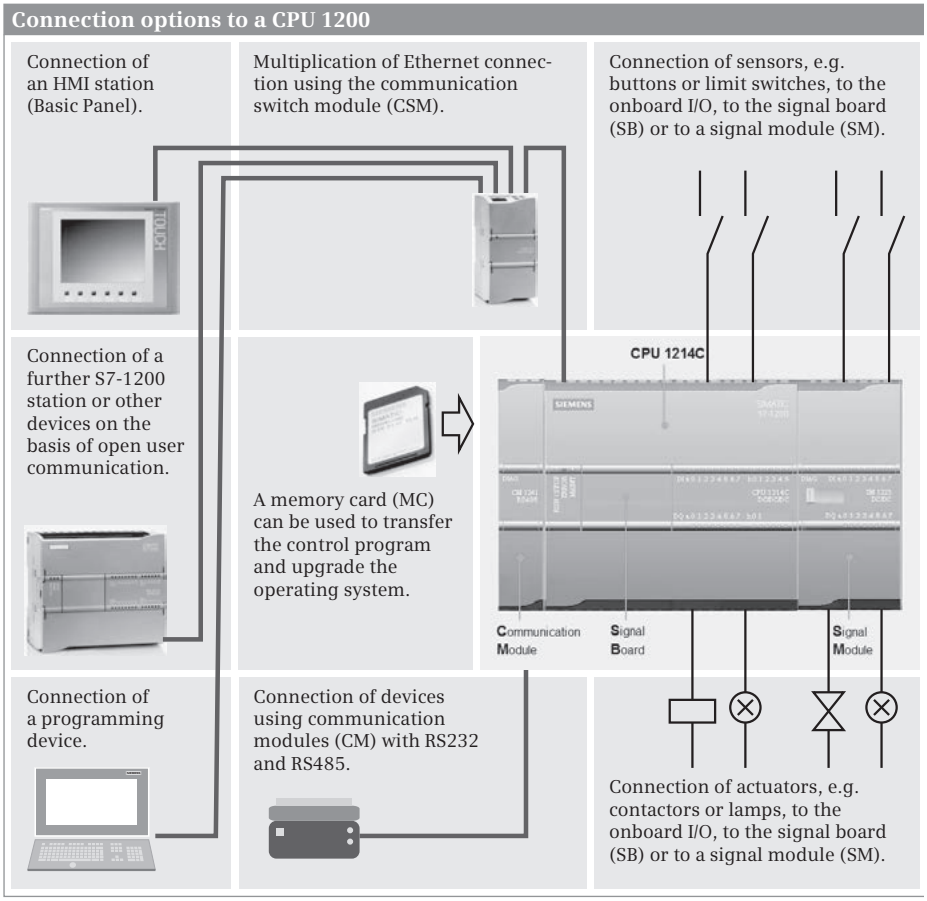


Fig. 1.2 Connection options to a PLC station with CPU 1200

gram is to be transferred online to the CPU and tested on the machine. Data is exchanged with HMI stations and other automation devices via this interface.

If the CPU is only connected to one device over Ethernet, a standard or crossover cable can be used. If more than two devices that only have a PROFINET interface are networked, the connecting cables must be routed via a multiplier, e.g. the **communication switch module (CSM)**. A CPU 1215 has two ports connected with a switch so that they can be networked with the next programmable controller without an interposed connection multiplier.

Communication modules (**CM**) permit the operation on further bus systems such as PROFIBUS DP. Here, an S7-1200 station in a DP master system can be both DP master and DP slave. An S7-1200 station can be the AS-Interface master on AS-Interface and can control up to 62 AS-Interface field devices. The communication module for the point-to-point connection is available with RS232 or RS485 interface, to which, for example, a barcode or RFID reader can be connected.

1.1.2 Overview of STEP 7 Basic

STEP 7 is the central automation tool for SIMATIC. STEP 7 requires authorization (licensing), and is executed on the current Microsoft Windows operating systems. STEP 7 Basic can be used to configure the S7-1200 controllers and – with WinCC Basic – the Basic Panels. Configuration is carried out in two views: the Portal view and the Project view.

The **Portal view** is task-oriented.

In the Start portal you can open an existing project, create a new project, or migrate an (HMI) project. A "project" is a data structure containing all the programs and data required for your automation task. The most important STEP 7 tools and functions can be accessed from here via further portals (Fig. 1.3):

- In the *Devices & networks* portal you configure the programmable controllers, i.e. you position the modules in a rack and assign them parameters.
- In the *PLC programming* portal you create the user program in the form of individual sections referred to as "blocks".
- The *Visualization* portal provides the most important tools for configuration and simulation of Basic Panels.
- The *Online & Diagnostics* portal allows you to connect the programming device online to a CPU. You can control the CPU's operating modes, and transfer and test the user program.

The **Project view** is an object-oriented view with several windows whose contents change depending on the current activity. In the *Device configuration*, the focal point is the working area with the device to be configured. The Device view includes the rack and the modules which have already been positioned (Fig. 1.4). A further

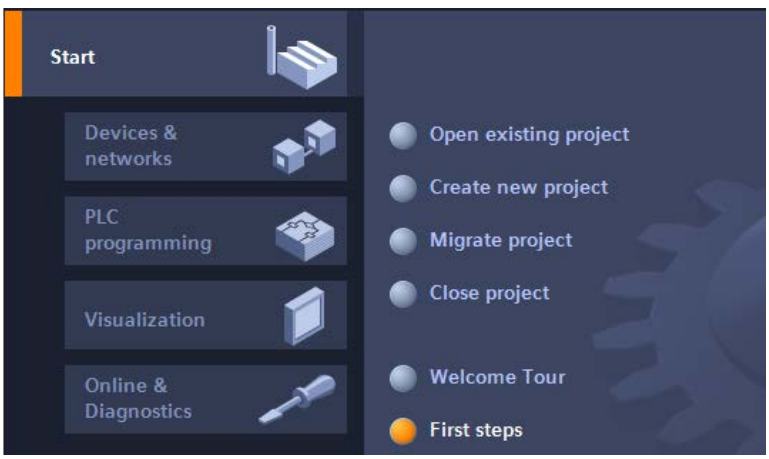


Fig. 1.3 Tools in the Start portal of STEP 7 Basic

window – the inspector window – displays the properties of the selected module, and the task window provides support by means of the hardware catalog with the available modules. The Network view shows the networking between the devices and permits the configuration of communication connections.

When carrying out *PLC programming* you edit the selected block in the working area. You are again shown the properties of the selected object in the inspector window where you can adjust them. In this case, the task window contains the catalog of statements with the available program elements and functions. The same applies to the processing of PLC tags, to the online program test using watch tables, or to configuration of an HMI device.

And you always have a view of the *project tree*. This contains all objects of the STEP 7 project. You can therefore select an object at any time, for example a program block or watch table, and edit this object using the corresponding editors which start automatically when the object is opened.

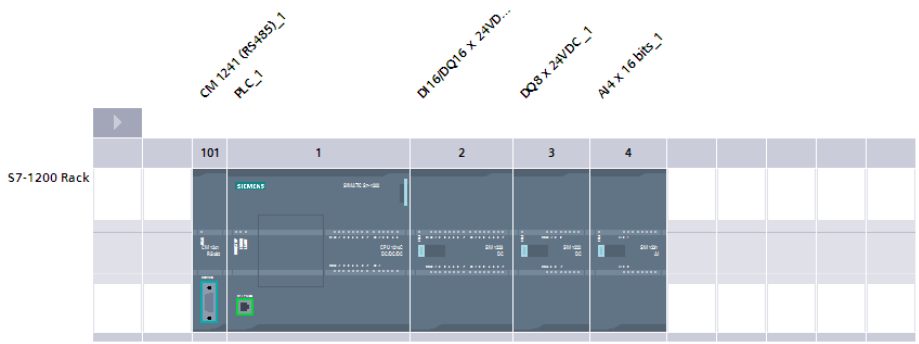


Fig. 1.4 Example of working area of device configuration

1.1.3 Three programming languages

You can select between three programming languages for the user program: ladder logic (LAD), function block diagram (FBD), and structured control language (SCL). The user program can be structured into individual parts known as “blocks”. The programming language is a property of a block, which means you can use the programming language that is best suited to resolve the block function for every block in the user program.

Using the **ladder logic**, you program the control task based on the circuit diagram. Operations on binary signal states are represented by serial or parallel arrangement of contacts (Fig. 1.5). A current path is terminated by a coil. Complex functions are represented by boxes which you handle like contacts or coils. Examples of boxes are mathematical functions or functions for processing strings.

Using the **function block diagram**, you program the control task based on electronic circuitry systems. Binary operations are implemented by linking AND and OR

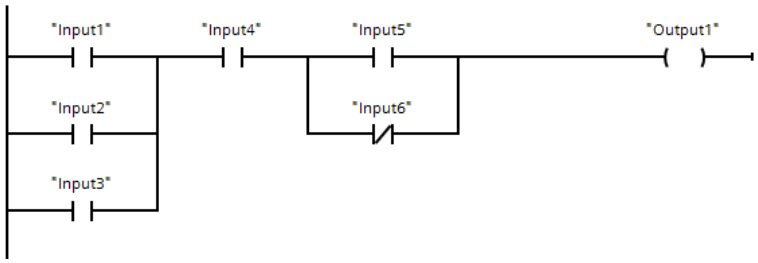


Fig. 1.5 Example of binary operations in ladder logic representation

functions and terminated by simple boxes (Fig. 1.6). Complex boxes are used to handle the operations on digital tags, for example with mathematical functions or functions for strings.

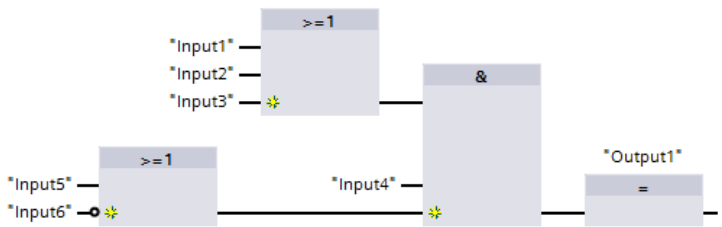


Fig. 1.6 Example of binary operations in function block diagram representation

Structured control language is particularly suitable for programming complex algorithms or for tasks in the area of data management. The program is made up of SCL statements which, for example, can be value assignments, comparisons, or control statements (Fig. 1.7).

```
18  (*****************************************************************************)
19  Write_register:
20  IF #Level = #Register_length - 1
21      THEN #Full := TRUE;
22      ELSE #Register[#Write_pointer] := #Input_value;
23          #Level := #Level + 1;
24          IF #Write_pointer = #Register_length
25              THEN #Write_pointer := 0;
26              ELSE #Write_pointer := #Write_pointer + 1;
27          END_IF;
28          #Empty := FALSE;
29  END_IF; RETURN;
```

Fig. 1.7 Example of SCL statements

1.1.4 Execution of the user program

After the power supply has been switched on, the control processor checks the consistency of the hardware and parameterizes the modules. A startup program is then executed once, if present. The startup program belongs to the user program that you program. Settings and initialization operations for the user program can be present here.

The user program is usually divided into individual sections called “blocks”. The organization blocks (OB) represent the interface between operating system and user program. The operating system calls an organization block for specific events, and the user program is then processed in it (Fig. 1.8).

Function blocks (FB) and functions (FC) are available for structuring the program. Function blocks have a memory in which local tags are saved permanently, functions do not have this memory.

Program statements are available for calling function blocks and functions (start of execution). Each block call can be assigned inputs and outputs, referred to as “block parameters”. During calling, tags can be transferred with which the program in the block is to work. In this manner, a block can be repeatedly called with a certain function (e.g. addition of three tags) but with different parameters sets (e.g. for different calculations) (Fig. 1.9).

The data of the user program is saved in data blocks (DB). Instance data blocks have a fixed assignment to a call of a function block; they are the tag memory of the function block. Global data blocks contain data which is not assigned to any block.

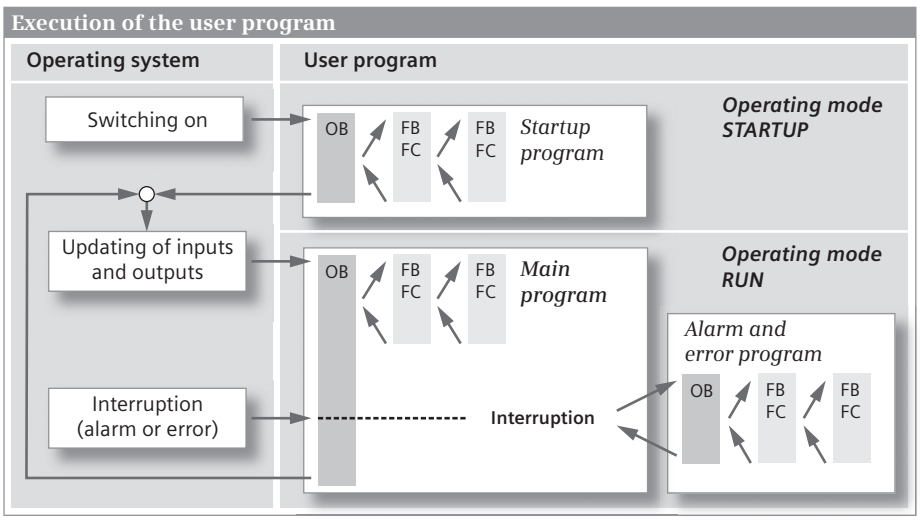


Fig. 1.8 Execution of the user program

Following a restart, the control processor updates the input and output signals in the process images and calls the organization block OB 1. The main program is present here. Structuring is also possible (and recommended) in the main program. Once the main program has been processed, the control processor returns to the operating system, retains (for example) communication with the programming device, updates the input and output signals, and then recommences with execution of the main program.

Cyclic program execution is a feature of programmable controllers. 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”.

In addition to the cyclically executed main program it is possible to carry out interrupt-controlled program execution. You must enable the corresponding interrupt event for this. This can be a hardware interrupt, such as a request from the controlled machine for a fast response, or a cyclic interrupt, in other words an event which takes place at defined intervals.

The control processor interrupts execution of the main program when an event occurs, and calls the assigned interrupt program. You can assign organization blocks to certain events, and these blocks are then processed in such a case. Once the interrupt program has been executed, the control processor continues execution of the main program from the point of interruption.

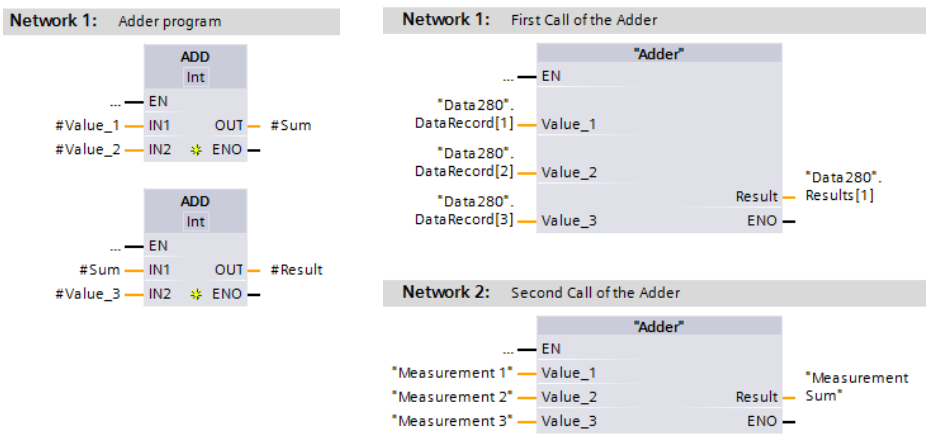


Fig. 1.9 Example of two block calls with different tags in each case

1.1.5 Data management in the SIMATIC automation system

The automation data is present in various memory locations in the automation system. Initially there is the programming device, referred to generally as the generation or engineering system. All automation data of a STEP 7 project is saved on its hard disk. Configuration and programming of the project data with STEP 7 is carried out in the main memory of the programming device (Fig. 1.10).

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 down-loaded to a programmable controller. The data downloaded into the user memory of the CPU module are known as the *online project data*.

The user memory on the CPU is divided into three components: The *load memory* contains the complete user program including the configuration data, the *work memory* contains the executable user program with the current control data, and the *retentive memory* contains the tags whose current values are saved power-failure-proof.

The memory card as a *transfer card* can transfer the user program to the CPU memory, or as a *program card* expand the CPU's internal load memory. When used as a program card, the memory card remains inserted in the CPU during runtime.

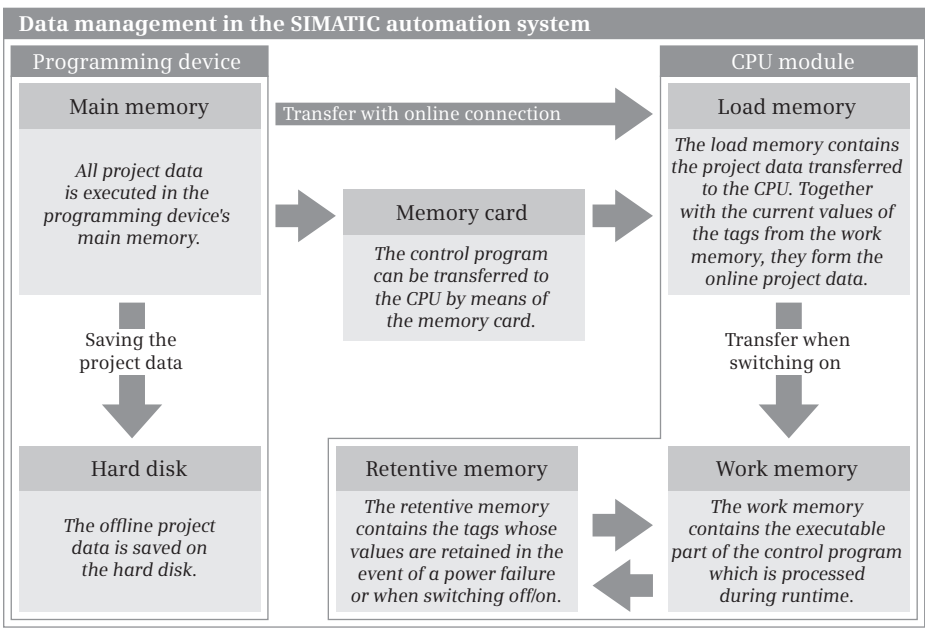


Fig. 1.10 Data management in the SIMATIC automation system