IEC 61131-3: Programming Industrial Automation Systems

Karl-Heinz John · Michael Tiegelkamp

IEC 61131-3: Programming Industrial Automation Systems

Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids

Second Edition

Springer

Karl-Heinz John
Irrlrinnig 13
91301 Forchheim
Germany
karlheinz.john@gmx

Michael Tiegelkamp
Kurpfalzstr. 34
90602 Pyrbaum
Germany
Michael.Tiegelkamp@gmx.de

This book contains one Trial DVD. **"SIMATIC STEP 7 Professional, Edition 2006 SR5, Trial License"** encompasses: SIMATIC STEP 7 V5.4 SP4, S7-GRAPH V5.3 SP6, S7-SCL V5.3 SP5, S7-PLCSIM V5.4 SP2 and can be used for trial purposes for 14 days.

This Software can only be used with the Microsoft Windows XP Professional Edition SP3 or Microsoft Windows Vista 32 Bit Business SP1/SP2 or Microsoft Windows Vista 32 Bit Ultimate SP1/SP2 operating systems.

Additional information can be found in the Internet at:
http://www.siemens.com/sce/contact
http://www.siemens.com/sce/modules
http://www.siemens.com/sce/tp

This book also contains one Trial CD-ROM: "Open PCS", a system (full version) for programming with IEC 61131.3, running on any standard Windows PC, using the languages: IL, LD, FBD, SFC, ST and CFC; running under Windows Server 2003, Windows XP SP2 or Windows Vista 32bit. PLC simulation SmartPLC is available for simulating the programs on a PC. The dedicated OPC server SmartPLC/OPC is only required, if additional third-party hardware and/or external OPC clients are connected.

Additional information can be found in the Internet at:
http://www.infoteam.de

ISBN 978-3-642-12014-5 e-ISBN 978-3-642-12015-2
DOI 10.1007/978-3-642-12015-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010925149

Cover design: WMXDesign GmbH, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface of the 2nd revised edition

IEC 61131 ("IEC 1131" until 1998) has become widely established in recent years as the programming standard in automation industry. Today, a wide range of small to large PLC manufacturers offer programming systems that are based on this standard. Additional standards and recommendations (e.g. for Motion Control) complement IEC 61131 with functionality in response to growing market requirements.

One of the most important advancements is IEC 61499 (formerly IEC 1499). The basic concepts and ideas of this standard are described in a separate chapter (Chapter 9). Its significance in connection with distributed PLC systems is discussed in Section 7.8.

IEC 61131 is now available in a second edition. The numerous changes and supplements to this standard have been incorporated in the 2nd edition of this book.

A comprehensive index at the end of the book facilitates the search for specific topics.

The enclosed DVD and CD contain the complete demo versions of two programming systems (in the latest versions), enabling the reader to immediately implement and consolidate the knowledge gained from this book by practical application.

We would like to thank SIEMENS AG and infoteam Software AG for providing the enclosed software.

Our special thanks go again to Hans-Peter Otto, member of the IEC and DKE standardisation committees for his active support and mutual inspiration.

With our sincere thanks also to all the people who helped to translate and finish this English version: Andrea Thieme, Kay Thomas-Sukrow, Robie O'Brien, Ormond O'Neill and Michael Sperber.

Above all, we want to thank our families, Susanne, Andreas, Tobias and Andrea, Vera, Olaf, Vanessa and Sebastian, for being so understanding and giving us the freedom to write this book.

We are grateful about the great interest in this book and would like to thank our attentive readers for their numerous suggestions, comments and feedback on typographical errors.

Karl-Heinz John Michael Tiegelkamp

Winter 2009/2010

Contents

1 Introduction

The rapid advances in performance and miniaturisation in microtechnology are constantly opening up new markets for the programmable logic controller (PLC). Specially designed controller hardware or PC-based controllers, extended by hardware and software with real-time capability, now control highly complex automation processes. This has been extended by the new subject of “safety-related controllers”, aimed at preventing injury by machines during the production process.

The different types of PLC cover a wide task spectrum - ranging from small network node computers and distributed *compact units* right up to modular, fault-tolerant, high-performance PLCs. They differ in performance characteristics such as processing speed, networking ability or the selection of I/O modules they support.

Throughout this book, the term PLC is used to refer to the technology as a whole, both hardware and software, and not merely to the hardware architecture. The IEC 61131 programming languages can be used for programming classical PLCs, embedded controllers, industrial PCs and even standard PCs, if suitable hardware (e.g. fieldbus board) for connecting sensors and actors is available.

The broad spectrum of capability of the hardware requires corresponding support from suitable programming tools, to allow low-cost, quality-conscious creation of both simple and complex software solutions. Desirable features of programming tools include:

- Simultaneous use of several PLC programming languages
- "Online" modification of programs in the PLC
- Reverse documentation of the programs from the PLC
- Reusability of PLC program blocks
- "Offline" testing and simulation of user programs
- Integrated configuring and commissioning tools
- Quality assurance, project documentation
- Use of systems with open interfaces.

Modern PCs have enabled increasingly efficient *PLC programming tools* to be developed in the last 10 years.

K.-H. John, M. Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*, 2nd ed., DOI 10.1007/978-3-642-12015-2_1,

The classical PLC programming methods, such as the Instruction List, Ladder Logic or Control System Function Chart, which have been employed until now, have reached their limits. Users want uniform, manufacturer-independent language concepts, high-level programming languages and development tools similar to those that have already been in existence in the PC world for many years.

With the introduction of the international standard IEC 1131 (meanwhile renamed to IEC 61131), a basis has been created for uniform PLC programming taking advantage of the modern concepts of software technology. The standard is now available in a revised second edition, which has been fully incorporated into this book.

1.1 Subject of the Book

The aim of this book is to give the reader an understandable introduction to the concepts and languages of standard IEC 61131. Simple examples are given to explain the ideas and application of the new PLC programming languages. An extensive example program summarises the results of each section.

The book serves as a helpful guide and introduction for people in training and at work who want to become acquainted with the possibilities of the new standard. It describes the methods of the standard from a manufacturer-independent perspective. Characteristics and specific versions of individual programming systems should be described in the relevant manuals.

Some experience with personal computers and basic knowledge in the field of PLC technology are required. Experienced PLC programmers will also find information here which will ease working with these programming systems. The book makes a point of describing the standard itself and less the relevant versions of programming systems available on the market.

This book is a useful reference work for students and facilitates the systematic learning of the new programming standard.

Readers can also use the enclosed "Buyer's Guide" to evaluate individual PLC programming systems for themselves. See the enclosed CD-ROM.

The formal contents and structure of the IEC standard are presented in a practice-oriented way. Difficult topics are clearly explained within their context, and the interpretation scope as well as extension possibilities of the standard are demonstrated.

This book is intended to give the reader concrete answers to the following questions:

- How do you program in accordance with IEC 61131? What are the essential ideas of the standard and how can they be applied in practice?
- What are the advantages of the new international standard IEC 61131 compared with other microcontroller programming or PC programming?
- What features must contemporary programming systems have in order to be consistent with IEC 61131 and to fulfil this standard?
- What do users need to look for when selecting a PLC programming system; what criteria are decisive for the performance of programming systems?

Chapter 2 presents the three basic building blocks of the standard: **program, function and function block**. An introductory example which includes the most important language elements of the standard and provides an overview of its programming methods gives an initial introduction to the concepts of IEC 61131.

Chapter 3 describes the **common language elements** of the five programming languages as well as the possibilities of data description with the aid of declarations.

The **five programming languages** of IEC 61131 are explained at length and illustrated by an extensive example in **Chapter 4**.

The strength of IEC 61131 is partly due to the uniform description of frequently used elements, the **standard functions** and **standard function blocks**. Their definition and application are described in **Chapter 5**.

After programming, the programs and the data have to be assigned to the features and hardware of the relevant PLC by means of **configuration**. This is to be found in **Chapter 6**.

The PLC market is developing into a technology with very specific requirements. These **special features of programming** for a PLC as well as their implementation using the facilities of IEC 61131 are the subject of **Chapter 7**.

Chapter 8 summarises the most important qualities of the standard from Chapters 2 to 7. The essential advantages of the standard and of consistent programming systems are outlined here for reference.

Chapter 9 introduces the standard **IEC 61499** for distributed automation processes. It is based on IEC 61131-3, but adopts a wider approach to cater for the demands for parallelism and decentralisation imposed by modern automation tasks.

Chapter 10 explains the use of the enclosed CD-ROM. It includes all programming examples in this book, a buyer's guide in tabular form, and executable versions of two IEC programming systems.

The **Appendices** supply further detailed information.

The **Glossary** in **Appendix I** gives a brief explanation of the most important terms used in this book in alphabetical order.

Appendix J contains the bibliography, which gives **references** not only to books but also to specialised papers on the subject of IEC 61131-3.

Appendix K is a general index which can be very helpful for locating keywords.

1.2 The IEC 61131 standard

In several parts, standard IEC 61131 summarises the requirements of PLC systems. These requirements concern the PLC hardware and the programming system.

The standard includes both the common concepts already in use in PLC programming and additional new programming methods.

IEC 61131-3 sees itself as a **guideline for PLC programming**, not as a rigid set of rules. The enormous number of details defined means that programming systems can only be expected to implement part but not all of the standard. PLC manufacturers have to document this amount: if they want to conform to the standard, they have to prove in which parts they do or do not fulfil the standard.

For this purpose, the standard includes 62 **feature tables** with requirements, which the manufacturer has to fill in with comments (e.g. "fulfilled; not implemented; the following parts are fulfilled:...").

The standard provides a **benchmark** which allows both manufacturers and user to assess how closely each programming system keeps to the standard, i.e. complies with IEC 61131-3.

For further proof of compliance, PLCopen (see Section 1.3) defines further tests for compliance levels which can be carried out by independent institutions.

The standard was established by working group SC65B WG7 (originally: SC65A WG6) of the international standardisation organisation **IEC (International Electrotechnical Commission)** which consists of representatives of different PLC manufacturers, software houses and users. This has the advantage that it is accepted as a guideline by most PLC manufacturers. Thus, IEC 61131-3 has made its way to become the only worldwide standard for PLC programming in recent years.

1.2.1 Goals and benefits of the standard

Because of the constantly increasing complexity of PLC systems there is a steady rise in costs for:

- Training of applications programmers
- The creation of increasingly larger programs
- The implementation of more and more complex programming systems.

PLC programming systems are gradually following the mass software market trend of the PC world. Here too, the pressure of costs can above all be reduced by standardisation, synergy, and reusability.

Manufacturers (PLC hardware and software).
Several manufacturers can invest together in the multi-million dollar software required to fulfil the functionality necessary in today's market.

The basic form of a programming system is determined to a large extent by the standard. Basic software such as editors, with the exception of particular parts like code generators or "online" modules, can be shared. Market differentiation results from supplementary elements to the basic package, which are required in specific market segments, as well as from the PLC hardware. Development costs can be substantially reduced by buying ready-made products. The error proneness of newly developed software can be greatly reduced by the use of previously tested software.

The development costs of contemporary programming tools have increased significantly as a result of the required functionality. By buying ready-made software components (offered only by some of the programming system manufacturers) or complete systems the "time to market" can be significantly shortened, which is essential in order to keep pace with the rapid hardware evolution.

Users
Users often work simultaneously with PLC systems from different manufacturers. Up to now this has meant that employees have needed to take several different training courses in programming, whereas with IEC 61131-3-compliant systems training is limited to the finer points of using the individual programming systems and additional special features of the PLCs. This cuts down on the need for system specialists and training personnel, and PLC programmers are more flexible.

The requirements of the standard ease the selection of suitable programming systems because systems that conform to the standard are easily comparable.

Though it is not expected that complete application programs will be able to be exchanged between different *PLC systems* in the foreseeable future, language elements and program structure are nevertheless similar among the different IEC systems. This facilitates porting onto other systems.

1.2.2 History and components

The standard *IEC 61131* represents a combination and continuation of different standards. It refers to 10 other international standards (IEC 50, IEC 559, IEC 617-12, IEC 617-13, IEC 848, ISO/AFNOR, ISO/IEC 646, ISO 8601, ISO 7185, ISO 7498). These include rules about the employed character code, the definition of the nomenclature used or the structure of graphical representations.

Several efforts have been made in the past to establish a standard for PLC programming technology. Standard IEC 61131 is the first standard to receive the necessary international (and industrial) acceptance. The most important precursor documents to IEC 61131 are listed in Table 1.1.

Year	German	international
1977	DIN 40 719-6 (function block diagrams)	IEC 848
1979		Start of the working group for the first IEC 61131 draft
1982	VDI guideline 2880, sheet 4 PLC programming languages	Completion of the first IEC 61131 draft; Splitting into 5 sub-workgroups
1983	DIN 19239 PLC programming	Christensen Report (Allen Bradley) PLC programming languages
1985		First results of the IEC 65 A WG6 TF3

Table 1.1. Important precursors of IEC 61131-3

The international English version is named IEC 61131followed by a number. Ed. is short for Edition and indicates the relevant issue status.

The *standard* consists of seven parts (status: December 2009). The overview below shows the relevant titles as well as the year of their first publication and their most recent edition in parentheses:

- IEC 61131-1 Ed. 2: General information (2003) [IEC 61131-1]
- IEC 61131-2 Ed. 3.0: Equipment requirements and tests (1994; 2007) [IEC 61131-2]
- IEC 61131-3 Ed. 2.0: Programming languages (1993; 2003); next revision envisaged for 2010 [IEC 61131-3]
- IEC 61131-4 Ed 2.0: User guidelines (1995; 2004) [IEC 61131-4]
- IEC 61131-5 Ed.1.0: Communications (2000) [IEC 61131-5]
- IEC 61131-7 Ed.1.0: Fuzzy control programming (2000) [IEC 61131-7]
- IEC 61131-8 Ed. 2.0: Guidelines for the application and implementation of programming languages for programmable controllers (1997; 2003) [IEC 61131-8].

In addition, **Corrigenda** are published for some of the standards. They include error descriptions in the currently valid edition of the standard and suggest corrections.

Part 1: General information:
Part 1 contains general definitions and typical functional features which distinguish a PLC from other systems. These include standard PLC properties, for example,

the cyclic processing of the application program with a stored image of the input and output values or the division of labour between programming device, PLC and human-machine interface.

Part 2: **Equipment requirements and tests:**
This part defines the electrical, mechanical and functional demands on the devices as well as corresponding qualification tests. The environmental conditions (temperature, air humidity etc.) and stress classes of the controllers and of the programming devices are listed.

Part 3: Programming languages:
Here the PLC programming languages widely used throughout the world have been co-ordinated into a harmonised and future-oriented version.

The basic software model and programming languages are defined by means of formal definitions, lexical, syntactical and (partially) semantic descriptions, as well as examples.

Part 4: User guidelines
The fourth part is intended as a guide to help the PLC user in all project phases of automation. Practice-oriented information is given on topics ranging from systems analysis and the choice of equipment to maintenance.

Part 5: Communications:
This part is concerned with communication between PLCs from different manufacturers with each other and with other devices.

In co-operation with ISO 9506 (Manufacturing Message Specification; MMS) conformity classes are defined to allow PLCs to communicate, for example, via networks. These cover the functions of device selection, data exchange, alarm processing, access control and network administration.

Part 6: Safety-related PLC:
The standardisation committee is currently working on the first issue of IEC 61131-6 “Safety-related PLC” with the goal of adapting the requirements of safety standard IEC 61508 (“Functional safety of electrical/electronic/programmable electronic safety-related systems”) and machine requirement IEC 62061 (“Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems”) to PLCs.

Part 7: Fuzzy Control Language:
The goal of this part of the standard is to provide manufacturers and users with a common understanding of the integration of fuzzy control applications based on IEC 61131-3 and to facilitate the portability of fuzzy programs between different manufacturers.

Part 8: Guidelines for the application and implementation of programming languages for Programmable Logic Controllers:
This document offers interpretations for questions not answered by the standard. It includes implementation guidelines, instructions for use by the final user as well as assistance in programming.

The standard describes a modern technology and is therefore subject to strong innovation pressure. This explains why further development of the findings of the standard is being carried out at both national and international level.

This book is concerned with Part 3 "Programming Languages", in short IEC 61131-3. It includes the latest modifications and extensions incorporated with Edition 2 in 2003.

1.3 The Organisation PLCopen

PLCopen [PLCopen Europe] is a manufacturer-independent and product-independent international organisation. Many PLC manufacturers, software houses and independent institutions in Europe and overseas are members of the organisation. Coming from different industry sectors, the members are focused on the harmonisation of controller programming and the development of applications and software interfaces in the IEC 61131-3 environment.

In order to reduce the costs in industrial engineering, uniform specifications and implementation guidelines have been devised. These efforts resulted, for example, in standardised libraries for different application fields, the specification of a conformity level for programming languages, and interfaces for an enhanced exchange of software. The PLCopen expert members are organised in technical committees and define these open standards in co-operation with final users.

1.3.1 Aims of PLCopen

PLCopen was founded in 1992, immediately after publication of the standard IEC 61131-3. At that time, the controller market was highly heterogeneous, with a multitude of programming methods for numerous different types of PLCs. Today, IEC 61131-3 has gained worldwide acceptance as a programming standard and serves as the basis for products offered by many software and hardware manufacturers. Owing to PLCopen, it has thus been incorporated in many different machines and other fields of applications.

Today’s control system market poses completely new challenges. PLCopen complies with the market’s requirements and – as its core activity – has been

defining general standards to make automation more efficient. The latest topics of this standardisation work include:

- *Motion Control and safety functions*,
- *XML data exchange format* for standardising basic data of IEC projects in software systems and
- *Benchmarking* projects for devising a detailed benchmark standard .

New requirements in industry and new products will result in more, new automation tasks in the future. It will remain the mission of PLCopen to reach global harmonisation and a standardised understanding.
PLCopen is not another standardisation committee, but rather a group with a common interest wanting to help existing standards to gain international acceptance. Detailed information can be found on the Internet (http://www.plcopen.org).

1.3.2 Committees and fields of activity

PLCopen is divided into several committees, each of which handles a specific field of interest, as shown in Figure 1.1:

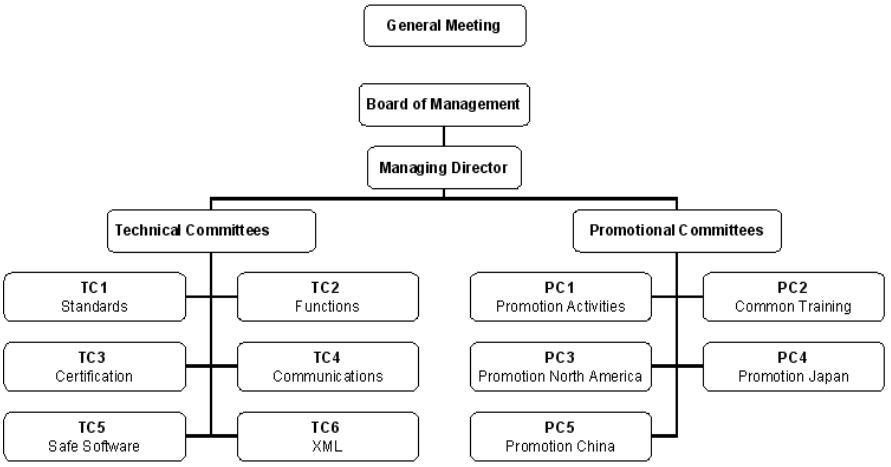

Figure 1.1. Committees of PLCopen

The technical committees work out guidelines for common policy; the promotional committees are responsible for marketing measures.

The work in the committees is carried out exclusively by representatives of individual companies and institutions. This ensures that the resulting papers will be accepted in industry.

1.3.3 Results

As a result of the preparatory work of the promotional committees, PLCopen is represented at several fairs in Europe, USA and the Far East. Workshops and advanced training seminars have brought the desired international recognition for PLCopen.

As a discussion forum for users, manufacturers and software houses some impressive technical results have been achieved:

- *Certification* for manufacturers of PLC programming systems,
- *Exchange format* for user programs.

The committees in detail:

TC 1 – Standards.
This committee is the interface to the international standardisation committees of IEC and OPC [OPC Foundation]. Its members collect suggestions on improvement or error correction for the reponsible IEC 61131 standardisation committee *IEC 65B WG7 working group* [IEC 65B WG7] ; they develop common positions and pass these on to the standardisation committees. In addition, they publish the latest results of the committees‘ work. In particular, improvements to the 2nd edition of the standard have thus been implemented.

TC 2 – Function and Function Blocks
The members of this committe define libraries of function blocks. The PLCopen Motion Control Specification, for example, has meanwhile become the market standard. This document is subdivided into the following parts:

- Part 1: Basics,
- Part 2: Extensions, additional function blocks,
- Part 3: User guidelines (guidelines and examples for users),
- Part 4: Coordinated motion, focused to the coordinated multi-axes motion in 3D space,
- Part 5: Homing (this function references a specific mechanical position).

TC 3 – Certification.
Several certification levels have been defined and test software has been developed in order to test programming systems for compliance with IEC 61131-3. The test is carried out by institutions accredited by PLCopen to guarantee the desired quality of compliance demonstration. The certification levels include:

- **Base Level (BL):** Original basic definition specifying the basic structure of a program in compliance with the IEC 61131-3 method and the programming manufacturer's declaration of conformity with the standard.
- **Reusability Level (RL):** Functions and function blocks are compatible to an extent that they are portable to different, RL-compliant programming systems.
- **Conformity Level (CL):** Certifies the highest-level of conformance. In many cases, the programming systems utilise only part of the multitude of data types in IEC 61131-3 (26). All data types listed as used in the system by the manufacturer are subjected to a strict conformity test.

The Benchmark working group is also organised under TC 3. It defines specifications and tests for scripts that allow reproducible and portable performance testing of programming systems.

TC 4 – Communication.
TC 4 is involved in definitions at the interface between IEC 61131-3 programming systems and communication systems such as Profibus or CAN.

TC 5 – Safe Software.
This committee makes recommendations for using IEC 61131-3 systems in safety-critical environments. In particular, this includes the new standards IEC 61508 and IEC 61511. Moreover, TC 5 prepares user guidelines covering safety aspects in safety-critical applications.

TC 6 – XML.
TC 6 defines XML schemes for the description of IEC 61131-3 application programs and projects in XML. This includes the textual and graphical languages, variable declaration, and configuration. The specification supports

- the exchange of blocks between systems
- the interface to other software packets such as documentation, simulation, or verification tools.

PC 1 – General Promotion.
This committee is involved in promotional activities in Europe and areas for which there is no separate promotional committee (PCs have also been set up for North America, China, and Japan) [PLCopen Europe].

PC 2 – Common Training.

PC 2 prepares documents for IEC 61131-3 training courses to be held by accredited training institutions [PLCopen Europe].

PC 3 – Promotion North America.
This committee is involved in promotional activities in North America [PLCopen North America].

PC 4 – Promotion Japan.
PC 4 is involved in promotional activities in Japan, in close co-operation with the local PLCopen office [PLCopen Japan].

PC 4 – Promotion China.
PC 5 is involved in promotional activities in China, in close co-operation with the local PLCopen office [PLCopen China].

2 Building Blocks of IEC 61131-3

This chapter explains the meaning and usage of the main language elements of the IEC 61131-3 standard. These are illustrated by several examples from real life, with each example building upon the previous one.

The reader is introduced to the terms and ways of thinking of IEC 61131-3. The basic ideas and concepts are explained clearly and comprehensively without discussing the formal language definitions of the standard itself [IEC 61131-3].

The first section of this chapter gives a compact introduction to the conceptual range of the standard by means of an example containing the most important language elements and providing an overview of the methodology of PLC programming with IEC 61131-3.

The term *"POU" (Program Organisation Unit)* is explained in detail because it is fundamental for a complete understanding of the new language concepts.

As the programming language Instruction List (IL) is already well known to most PLC programmers, it has been chosen as the basis for the examples in this chapter. IL is widespread on the European PLC market and its simple syntax makes it easy to comprehend.

The programming language IL itself is explained in Section 4.1.

2.1 Introduction to the New Standard

IEC 61131-3 not only describes the PLC programming languages themselves, but also offers comprehensive concepts and guidelines for creating PLC projects.

The purpose of this section is to give a short summary of the important terms of the standard without going into details. These terms are illustrated by a simple example. More detailed information will be found in the subsequent sections and chapters.

K.-H. John, M. Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*, 2nd ed., DOI 10.1007/978-3-642-12015-2_2,

2.1.1 Structure of the building blocks

POUs correspond to the *Blocks* in previous (conventional) programming systems. POUs can call each other with or without parameters. As the name implies, POUs are the smallest independent software units of a user program.

There are three types of POUs: *Function (FUN)*, *Function block (FB)* and *Program (PROG)*, in ascending order of functionality. The main difference between functions and function blocks is that functions always produce the same result (function value) when called with the same input parameters, i.e. they have no "memory". Function blocks have their own data record and can therefore "remember" status information (*instantiation*). Programs (PROG) represent the "top" of a PLC user program and have the ability to access the I/Os of the PLC and to make them accessible to other POUs.

IEC 61131-3 predefines the calling interface and the behaviour of frequently needed s*tandard functions (std. FUN)* such as arithmetic or comparison functions, as well as *standard function blocks (std. FB),* such as timers or counters.

Declaration of variables

The IEC 61131-3 standard uses *variables* to store and process information. Variables correspond to (global) flags or bit memories in conventional PLC systems. However, their storage locations no longer need to be defined manually by the user (as absolute or global addresses), but they are managed automatically by the programming system and each possess a fixed *data type*.

IEC 61131-3 specifies several data types (Bool, Byte, Integer, ...). These differ, for example, in the number of bits or the use of signs. It is also possible for the user to define new data types: user-defined data types such as structures and arrays.

Variables can also be assigned to a certain I/O address and can be battery-backed against power failure.

Variables have different forms. They can be defined (declared) outside a POU and used program-wide, they can be declared as interface parameters of a POU, or they can have a local meaning for a POU. For declaration purposes they are therefore divided into different variable types. All variables used by a POU have to be declared in the declaration part of the POU.

The declaration part of a POU can be written in textual form independently of the programming language used. Parts of the declaration (input and output parameters of the POU) can also be represented graphically.

```
VAR_INPUT                          (* Input variable *)
  ValidFlag      : BOOL;           (* Binary value *)
END_VAR
VAR_OUTPUT                         (* Output variable *)
  RevPM          : REAL;           (* Floating-point value *)
END_VAR
VAR RETAIN                         (* Local variable, battery-backed *)
  MotorNr        : INT;            (* Signed integer *)
  MotorName      : STRING [10];    (* String of length 10 *)
  EmStop AT %IX2.0 : BOOL;         (* Input bit 2.0 of I/O *)
END_VAR
```

Example 2.1. Example of typical variable declarations of a POU

Example 2.1 shows the variable declaration part of a POU. A signed integer variable (16 bits incl. sign) with name MotorNr and a text of length 10 with name MotorName are declared. The binary variable EmStop (emergency stop) is assigned to the I/O signal input 2.0 (using the keyword "AT"). These three variables are known only within the corresponding POU, i.e. they are "local". They can only be read and altered by this POU. During a power failure they retain their value, as is indicated by the qualifier "RETAIN". The value for input variable ValidFlag will be set by the calling POU and have the Boolean values TRUE or FALSE. The output parameter returned by the POU in this example is the floating-point value RevPM.

The Boolean values TRUE and FALSE can also be indicated by "1" and "0".

Code part of a POU

The code part, or instruction part, follows the declaration part and contains the instructions to be processed by the PLC.

A POU is programmed using either the textual programming languages Instruction List (IL) and Structured Text (ST) or the graphical languages Ladder Diagram (LD) and Function Block Diagram (FBD). IL is a programming language closer to machine code, whereas ST is a high-level language. LD is suitable for Boolean (binary) logic operations. FBD can be used for programming both Boolean (binary) and arithmetic operations in graphical representation.

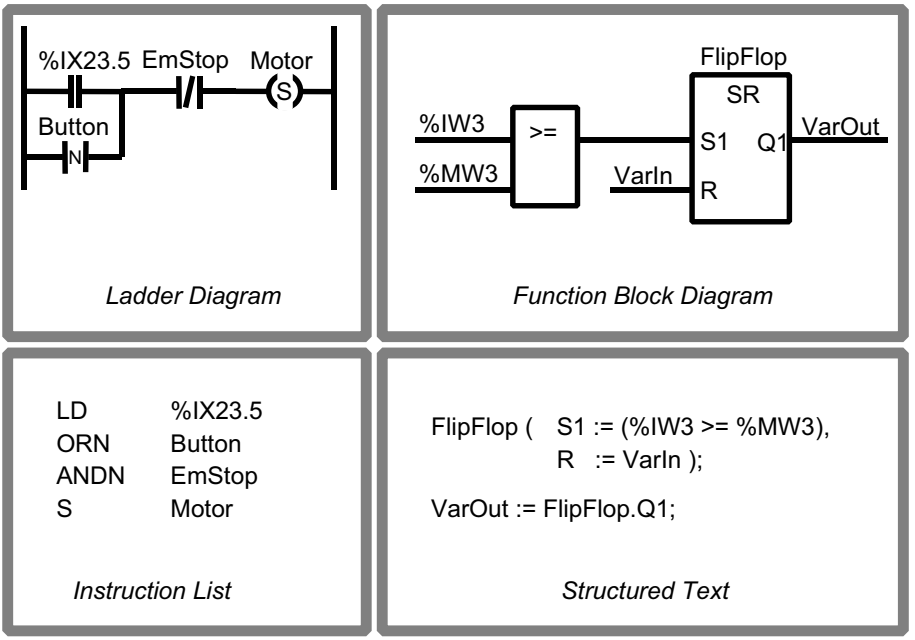

Figure 2.1. Simple examples of the programming languages LD, FBD, IL and ST. The examples in LD and IL are equivalent to one another, as are those in FBD and ST.

Additionally, the description language Sequential Function Chart (SFC) can be used to describe the structure of a PLC program by displaying its sequential and parallel execution. The various subdivisions of the SFC program (steps and transitions) can be programmed independently using any of the IEC 61131-3 programming languages.

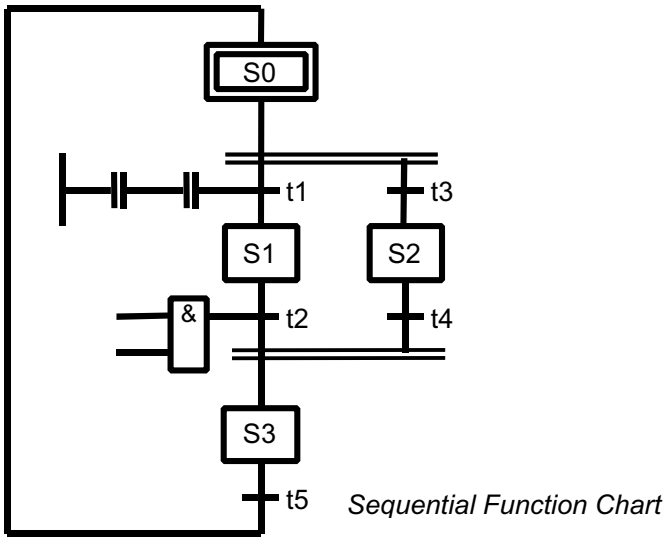

Figure 2.2. Schematic example of structuring using SFC. The execution parts of the steps (S0 to S3) and the transitions (t1 to t5) can be programmed using any other programming language.

Figure 2.2 shows an SFC example: Steps S0, S1 and S3 are processed sequentially. S2 can be executed alternatively to S1. Transitions t1 to t5 are the conditions which must be fulfilled before proceeding from one step to the next.

2.1.2 Introductory example written in IL

An example of an IEC 61131-3 program is presented in this section. Figure 2.3 shows its POU calling hierarchy in tree form.

This example is not formulated as an executable program, but simply serves to demonstrate POU structuring.