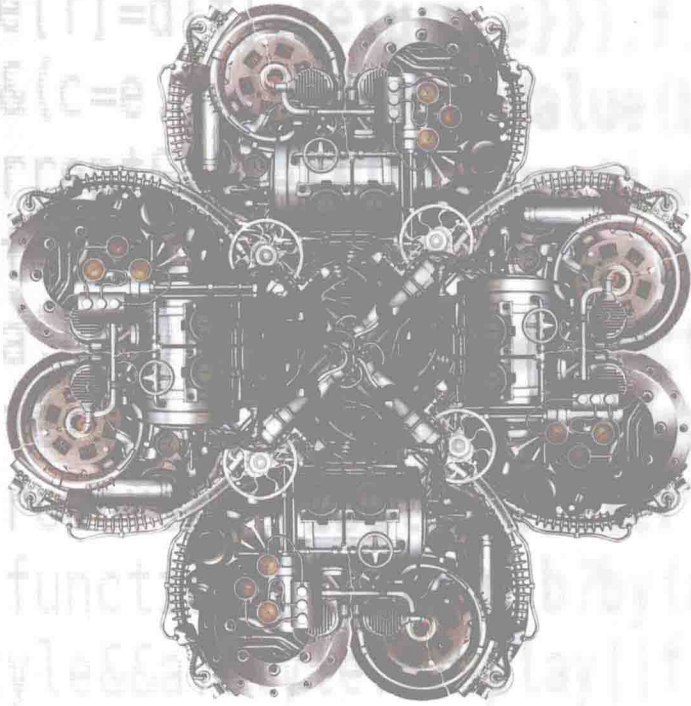


DAG H. HANSEN



PROGRAMMABLE
LOGIC
CONTROLLERS

A PRACTICAL APPROACH TO
IEC 61131-3 USING CODESYS®

WILEY

PROGRAMMABLE LOGIC CONTROLLERS

A PRACTICAL APPROACH TO IEC 61131-3 USING CODESYS

Dag H. Hanssen

Institute of Engineering and Safety, University of Tromsø, Norway

Translated by

Dan Lufkin

WILEY

This edition first published 2015
© 2015 John Wiley & Sons, Ltd

Registered Office

John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. It is sold on the understanding that the publisher is not engaged in rendering professional services and neither the publisher nor the author shall be liable for damages arising herefrom. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Authorised Translation from the Norwegian language edition published by Akademika forlag, Programmerbare Logiske Styrtninger – basert på IEC 61131-3, 4. Utgave. This translation has been published with the financial support of NORLA.

Library of Congress Cataloging-in-Publication Data

Hanssen, Dag Håkon, author.

Programmable Logic Controllers: A Practical Approach to IEC 61131-3 using CODESYS / Dag Hakon Hanssen.
pages cm

Includes bibliographical references and index.

ISBN 978-1-118-94924-5 (pbk.)

1. Sequence controllers, Programmable. 2. Programmable logic devices. I. Title. TJ223.P76H36 2015 621.39'5-dc23

2015018742

A catalogue record for this book is available from the British Library.

Set in 10/12pt Times by SPi Global, Pondicherry, India
Printed and bound in Singapore by Markono Print Media Pte Ltd

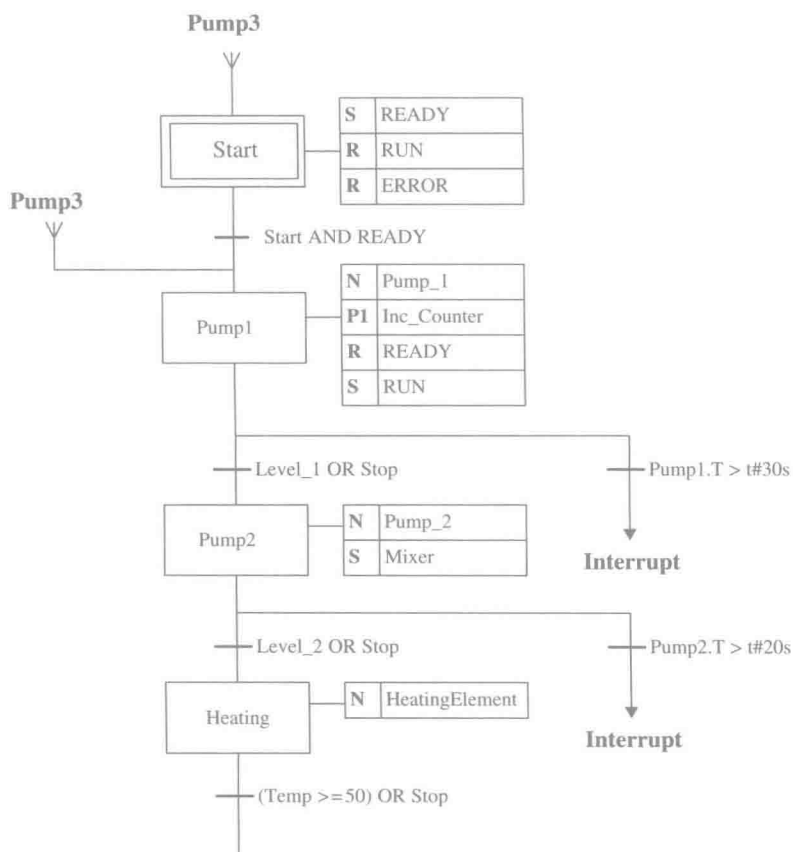
PROGRAMMABLE LOGIC CONTROLLERS

Programmable Logic Controllers

A Practical Approach to IEC 61131-3

Using CODESYS

First edition



Dag H. Hanssen

Preface

As long as there have been competing producers of PLCs on the market, there have been different programming languages from one PLC brand to another. Even though the same languages, beginning with Instruction Lists (IL) and Ladder diagram (LD), have been used by most of the producers, all of them added their own “dialects” to the languages. When physical programming terminals replaced software-based programming tools, the differences between languages of the various producers escalated. Several programming languages also saw the light of day. This development was the natural result of the attempt by the producers to make themselves stand out among increasing competition by developing the most user-friendly languages and tools.

When the IEC¹ 61131-3 standard came out in 1993, the situation started to improve. This standard was the result of the work that had been ongoing for several years in which the best from the various languages and dialects from different producers was assembled into a single document. This is not a rigid standard in the sense that the producers *must* follow all requirements and specifications, but more a set of guidelines that the producers could choose to follow to a certain extent. Today, most of the equipment producers have come to realize the advantages of organizing themselves in accordance with the standard. All of the major producers of PLCs, such as Telemecanique, Wago, Mitsubishi, Klockner Moeller, Allen-Bradley, Omron, Siemens, and so on, have therefore, to a greater or lesser extent, adapted their programming tools to IEC 61131-3.

This book covers close to 100% of the specifications and guidelines that are given in Standard (International Electrotechnical Commission, 2013).² The book will therefore be interested to everyone who works with, or wants to learn about programming PLCs, no matter which PLC brand they use.

¹ IEC—International Electrotechnical Commission. This edition of the book was updated in conformity with the 3rd edition of IEC 61131-3, issued February 2013.

² The Standard IEC 61131-3 is introduced in Chapter 5.

The book does not assume any previous knowledge of programming. Comments and suggestions for contents will be gratefully received. The book is divided into five main parts:

- Part 1: Hardware Chapters 1–3
- Part 2: Methodic Chapter 4
- Part 3: IEC 61131-3 Chapters 5–8
- Part 4: Programming Chapters 9–13
- Part 5: Implementation Chapters 14–15

Chapter 1 contains a brief history and a short description of the design and operation of PLCs in general. Chapters 2 and 3 give a basic introduction to digital and analog signals and equipment for detection, measurement, and manipulation of discrete and continuous quantities.

Chapter 4 focuses on methods for planning and design of structurally efficient programs. It also provides an introduction into Boolean algebra. Chapters 5 and 6 introduce the IEC standard elements such as literals, keywords, data types, variables, and addressing. Chapters 7 and 8 cover standardized functions and functional blocks.

Chapters 9 to 13 deal with programming: Chapter 9 covers programming with LD. Chapter 10 covers functional block diagrams (FBD). Chapter 11 covers the structured text (ST) language. The last language covered in the book is actually not a programming language as such, but rather a tool for structuring program code. This is called a Sequential Function Chart (SFC) and is described in Chapter 12.

Chapter 13 contains some larger practical programming examples.

The last two chapters in the book cover programming tools. Here, I have chosen to focus on CODESYS. There are several reasons for this; first, CODESYS follows the standard almost 100%. Furthermore, CODESYS is a hardware-independent programming tool that is currently used by well over 250 hardware suppliers. Finally yet importantly, the program can be downloaded free and it contains a simulator. Most of the program code in the book was written and tested with this tool.

I would like to thank the following persons and companies:

- Associate Professor Tormod Drenstvig, University of Stavanger, for much good feedback, suggestions for improvements, and the contribution of several examples
- Assistant Professor Inge Vivås, Bergen University College, for giving his permission to reuse some problems (Section 4.6.4 and Problems 4.10 and 10.5)
- Assistant Professor Veslemøy Tyssø, Oslo and Akershus University College of Applied Science, for having read an earlier edition of the book and having provided expert contributions
- Colleagues and management at the University of Tromsø, Department of Engineering and Safety, for the support and patience
- Schneider Electric for granting me permission to use material from their “Automation Solution Guide” when writing about sensors in Chapter 2

Dag H. Hanssen

Contents

Preface	xiv
PART ONE HARDWARE	1
1 About PLCs	3
1.1 History	4
1.1.1 More Recent Developments	6
1.2 Structure	7
1.2.1 Inputs and Outputs	10
1.3 PLC Operation	13
1.3.1 Process Knowledge	14
1.3.2 Standard Operations	16
1.3.3 Cyclic, Freewheeling, or Event-Controlled Execution	18
1.4 Test Problems	19
2 Digital Signals and Digital Inputs and Outputs	20
2.1 Introduction	20
2.2 Terminology	21
2.2.1 Discrete, Digital, Logical, and Binary	21
2.2.2 Sensors, Transducers, and Transmitters	22
2.3 Switches	24
2.3.1 Limit Switches	24
2.3.2 Safety Devices	24
2.3.3 Magnetic Switches	25
2.4 Logical Sensors	26
2.4.1 Inductive Sensors	27
2.4.2 Capacitive Sensors	29
2.4.3 Photocells	30

2.4.4	Ultrasonic Sensors	33
2.4.5	Rotating Sensors (Encoders)	34
2.4.6	Other Detection Principles and Sensors	37
2.5	Connection of Logical Sensors	39
2.5.1	Sink/Source	41
2.5.2	Selecting a Sensor with the Proper Type of Output	43
2.6	Properties of Discrete Inputs	44
2.7	Discrete Actuators	45
2.7.1	Relays and Contactors	46
2.7.2	Solenoids and Magnetic Valves	47
2.7.3	Transistor Outputs versus Relay Outputs	49
2.8	Test Problems	50
3	Analog Signals and Analog I/O	52
3.1	Introduction	52
3.2	Digitalization of Analog Signals	53
3.2.1	Filtering	53
3.2.2	A/D Conversion	55
3.3	Analog Instrumentation	58
3.3.1	About Sensors	58
3.3.2	Standard Signal Formats	59
3.3.3	On the 4–20 mA Standard	59
3.3.4	Some Other Properties of Sensors	61
3.4	Temperature Sensors	61
3.4.1	Thermocouple	61
3.4.2	PT100/NI1000	62
3.4.3	Thermistors	64
3.5	Connection	64
3.5.1	About Noise, Loss, and Cabling	64
3.5.2	Connecting Sensors	67
3.5.3	Connection of a PT100 (RTD)	68
3.5.4	Connecting Thermocouples	72
3.6	Properties of Analog Input Modules	72
3.6.1	Measurement Ranges and Digitizing: Resolution	72
3.6.2	Important Properties and Parameters	74
3.7	Analog Output Modules and Standard Signal Formats	75
3.8	Test Problems	76
PART TWO	METHODIC	79
4	Structured Design	81
4.1	Introduction	81
4.2	Number Systems	82
4.2.1	The Decimal Number Systems	82
4.2.2	The Binary Number System	82

4.2.3	The Hexadecimal Number System	83
4.2.4	Binary-Coded Decimal Numbers	85
4.2.5	Conversion between Number Systems	86
4.3	Digital Logic	87
4.4	Boolean Design	91
4.4.1	Logical Functional Expressions	91
4.4.2	Boolean Algebra	93
4.5	Sequential Design	97
4.5.1	Flowchart	97
4.5.2	Example: Flowchart for Mixing Process	99
4.5.3	Example: Flowchart for an Automated Packaging Line	101
4.5.4	Sequence Diagrams	107
4.5.5	Example: Sequence Diagram for the Mixing Process	110
4.5.6	Example: Batch Process	112
4.6	State-Based Design	113
4.6.1	Why Use State Diagrams?	114
4.6.2	State Diagrams	114
4.6.3	Example: Batch Process	117
4.6.4	Example: Level Process	118
4.6.5	Example: Packing Facility for Apples	121
4.7	Summary	124
4.8	Test Problems	125
PART THREE IEC 61131-3		131
5	Introduction to Programming and IEC 61131-3	133
5.1	Introduction	133
5.1.1	Weaknesses in Traditional PLCs	134
5.1.2	Improvements with IEC 61131-3	136
5.1.3	On Implementation of the Standard	137
5.2	Brief Presentation of the Languages	138
5.2.1	ST	138
5.2.2	FBD	138
5.2.3	LD	139
5.2.4	IL	139
5.2.5	SFC	141
5.3	Program Structure in IEC 61131-3	141
5.3.1	Example of a Configuration	145
5.4	Program Processing	146
5.4.1	Development of Programming Languages	146
5.4.2	From Source Code to Machine Code	147
5.5	Test Problems	151
6	IEC 61131-3: Common Language Elements	152
6.1	Introduction	152

6.2	Identifiers, Keywords, and Comments	153
6.2.1	Identifiers	153
6.2.2	Keywords	154
6.2.3	Comments	154
6.3	About Variables and Data Types	156
6.4	Pragmas and Literals	156
6.4.1	Literal	157
6.5	Data Types	158
6.5.1	Numerical and Binary Data Types	158
6.5.2	Data Types for Time and Duration	161
6.5.3	Text Strings	163
6.5.4	Generic Data Types	164
6.5.5	User-Defined Data Types	166
6.6	Variables	169
6.6.1	Conventional Addressing	170
6.6.2	Declaration of Variables with IEC 61131-3	171
6.6.3	Local Versus Global Variables	174
6.6.4	Input and Output Variables	175
6.6.5	Other Variable Types	176
6.7	Direct Addressing	176
6.7.1	Addressing Structure	176
6.7.2	I/O-Addressing	178
6.8	Variable versus I/O-Addresses	179
6.8.1	Unspecified I/O-Addresses	179
6.9	Declaration of Multielement Variables	180
6.9.1	Arrays	181
6.9.2	Data Structures	182
6.10	Test Problems	184
7	Functions	187
7.1	Introduction	187
7.2	On Functions	188
7.3	Standard Functions	189
7.3.1	Assignment	190
7.4	Boolean Operations	191
7.5	Arithmetic Functions	192
7.5.1	Overflow	193
7.6	Comparison	194
7.7	Numerical Operations	195
7.7.1	Priority of Execution	196
7.8	Selection	197
7.9	Type Conversion	197
7.10	Bit-String Functions	199
7.11	Text-String Functions	200
7.12	Defining New Functions	202
7.13	EN/ENO	203
7.14	Test Problems	204

8	Function Blocks	206
8.1	Introduction	206
8.1.1	The Standard's FBs	207
8.2	Declaring and Calling FBs	207
8.3	FBs for Flank Detection	208
8.4	Bistable Elements	209
8.5	Timers	210
8.6	Counters	211
8.6.1	Up-Counter	212
8.6.2	Down-Counter	212
8.6.3	Up/Down-Counter	212
8.7	Defining New FBs	213
8.7.1	Encapsulation of Code	214
8.7.2	Other Nonstandardized FBs	216
8.8	Programs	217
8.8.1	Program Calls	218
8.8.2	Execution Control	219
8.9	Test Problems	220
PART FOUR PROGRAMMING		221
9	Ladder Diagram (LD)	223
9.1	Introduction	223
9.2	Program Structure	224
9.2.1	Contacts and Conditions	225
9.2.2	Coils and Actions	226
9.2.3	Graphical Elements: An Overview	227
9.3	Boolean Operations	227
9.3.1	AND/OR-Conditions	227
9.3.2	Set/Reset Coils	230
9.3.3	Edge Detecting Contacts	233
9.3.4	Example: Control of a Mixing Process	234
9.4	Rules for Execution	237
9.4.1	One Output: Several Conditions	237
9.4.2	The Importance of the Order of Execution	238
9.4.3	Labels and Jumps	239
9.5	Use of Standard Functions in LD	240
9.6	Development and Use of FBs in LD	242
9.7	Structured Programming in LD	244
9.7.1	Flowchart versus RS-Based LD Code	248
9.7.2	State Diagrams versus RS-Based LD Code	253
9.8	Summary	259
9.9	Test Problems	260
10	Function Block Diagram (FBD)	262
10.1	Introduction	262

10.2	Program Structure	263
10.2.1	Concepts	264
10.3	Execution Order and Loops	264
10.3.1	Labels and Jumps	265
10.4	User-Defined Functions and FBs	266
10.5	Integer Division	268
10.6	Sequential Programming with FBD	271
10.7	Test Problems	273
11	Structured Text (ST)	278
11.1	Introduction	278
11.2	ST in General	279
11.2.1	Program Structure	280
11.3	Standard Functions and Operators	281
11.3.1	Assignment	282
11.4	Calling FBs	283
11.4.1	Flank Detection and Memories	284
11.4.2	Timers	287
11.4.3	Counters	288
11.5	IF Statements	288
11.6	CASE Statements	290
11.7	ST Code Based upon State Diagrams	292
11.7.1	Example: Code for the Level Process	295
11.8	Loops	298
11.8.1	WHILE ... DO... END_WHILE	298
11.8.2	FOR ... END_FOR	299
11.8.3	REPEAT ... END_REPEAT	300
11.8.4	The EXIT Instruction	300
11.9	Example: Defining and Calling Functions	301
11.10	Test Problems	302
12	Sequential Function Chart (SFC)	306
12.1	Introduction	306
12.1.1	SFC in General	307
12.2	Structure and Graphics	307
12.2.1	Overview: Graphic Symbols	309
12.2.2	Alternative Branches	309
12.2.3	Parallel Branches	311
12.3	Steps	312
12.3.1	Step Addresses	313
12.3.2	SFC in Text Form (for Those Specially Interested...)	314
12.4	Transitions	314
12.4.1	Alternative Definition of Transitions	315
12.5	Actions	317
12.5.1	Action Types	318
12.5.2	Action Control	319
12.5.3	Alternative Declaration and Use of Actions	321

12.6	Control of Diagram Execution	322
12.7	Good Design Technique	323
12.8	Test Problems	326
13	Examples	331
13.1	Example 1: PID Controller Function Block: Structured Text	331
13.2	Example 2: Sampling: SFC	333
13.2.1	List of Variables	334
13.2.2	Possible Solution	334
13.3	Example 3: Product Control: SFC	337
13.3.1	Functional Description	338
13.3.2	List of Variables	338
13.3.3	Possible Solution	339
13.4	Example 4: Automatic Feeder: ST/SFC/FBD	342
13.4.1	Planning and Structuring	344
13.4.2	Alternative 1: SFC	345
13.4.3	Alternative 2: ST/FBD	347
PART FIVE IMPLEMENTATION		351
14	CODESYS 2.3	353
14.1	Introduction	353
14.2	Starting the Program	354
14.2.1	The Contents of a Project	356
14.3	Configuring the (WAGO) PLC	357
14.4	Communications with the PLC	360
14.4.1	The Gateway Server	361
14.4.2	Local Connection via Service Cable	362
14.4.3	Via Ethernet	363
14.4.4	Communication with a PLC Connected to a Remote PC	364
14.4.5	Testing Communications	365
14.5	Libraries	365
14.6	Defining a POU	367
14.7	Programming in FBD/LD	368
14.7.1	Declaring Variables	369
14.7.2	Programming with FBD	371
14.7.3	Programming with LD	372
14.8	Configuring Tasks	375
14.9	Downloading and Testing Programs	376
14.9.1	Debugging	377
14.10	Global Variables and Special Data Types	379
15	CODESYS Version 3.5	381
15.1	Starting a New Project	381
15.1.1	Device	382
15.1.2	Application	384

15.2	Programming and Programming Units (POUs)	386
15.2.1	Declaration of Variables	388
15.3	Compiling and Running the Project	389
15.3.1	Start Gateway Server and PLS and Set Up Communications	390
15.4	Test Problems	393
	Bibliography	395
	Index	396

Part One

Hardware

