



## **Programmable controllers**

### **Part 8: Guidelines for the application and implementation of programming languages**



This Australian Standard® was prepared by Committee IT-006, Industrial Process Measurement, Control and Automation. It was approved on behalf of the Council of Standards Australia on 17 November 2014.  
This Standard was published on 19 December 2014.

---

The following are represented on Committee IT-006:

- Australian Computer Society
  - Australian Industry Group
  - Australian Petroleum Production and Exploration Association
  - Australian Safety Critical Systems Association
  - Consult Australia
  - Engineers Australia
  - Institute of Chemical Engineers Australia
  - Institute of Instrumentation, Control and Automation Australia
  - ISACA
  - Process Control Society
  - University of Queensland
  - Workplace Health and Safety Queensland
- 

This Standard was issued in draft form for comment as DR AS IEC 61131.8:2014.

Standards Australia wishes to acknowledge the participation of the expert individuals that contributed to the development of this Standard through their representation on the Committee and through the public comment period.

---

#### **Keeping Standards up-to-date**

Australian Standards® are living documents that reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued.

Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments that may have been published since the Standard was published.

Detailed information about Australian Standards, drafts, amendments and new projects can be found by visiting [www.standards.org.au](http://www.standards.org.au)

Standards Australia welcomes suggestions for improvements, and encourages readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at [mail@standards.org.au](mailto:mail@standards.org.au), or write to Standards Australia, GPO Box 476, Sydney, NSW 2001.

---

Australian Standard<sup>®</sup>

## **Programmable controllers**

# **Part 8: Guidelines for the application and implementation of programming languages**

Originated as AS IEC 61131.8—2004.  
Second edition 2014.

### **COPYRIGHT**

© Standards Australia Limited

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968.

Published by SAI Global Limited under licence from Standards Australia Limited, GPO Box 476, Sydney, NSW 2001, Australia

ISBN 978 1 74342 933 4

## PREFACE

This Standard was prepared by the Standards Australia Committee IT-006, Industrial Process Measurement, Control and Automation, to supersede AS IEC 61131.8—2004.

The objective of this Standard is to provide guidelines for the implementation of the programming languages defined in AS IEC 61131.3, in programmable controller systems and their programming support environments (PSEs).

This Standard should be read in conjunction with the other parts of the AS IEC 61131 series.

This Standard is identical with, and has been reproduced from, IEC/TR 61131-8, Ed. 2.0 (2003), *Programmable controllers—Part 8: Guidelines for the application and implementation of programming languages*.

The principal differences between this and the previous edition have been made to align this edition with the current edition of AS IEC 61131.3.

As this Standard is reproduced from an International Standard, the following applies:

- (a) In the source text ‘this part of IEC 61131’ should read ‘this Australian Standard’.
- (b) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian Standards, as follows:

<i>Reference to International Standard</i>	<i>Australian Standard</i>
IEC	AS IEC
61131 Programmable controllers	61131 Programmable controllers
61131-1 Part 1: General information	61131.1 Part 1: General information
61131-2 Part 2: Equipment requirements and tests	61131.2 Part 2: Equipment requirements and tests
61131-3 Part 3: Programming languages	61131.3 Part 3: Programming languages
61131-5 Part 5: Communications	61131.5 Part 5: Communications

The term ‘informative’ has been used in this Standard to define the application of the annex to which it applies. An ‘informative’ annex is only for information and guidance.

## CONTENTS

1	General.....	9
1.1	Scope .....	9
1.2	Normative references .....	9
1.3	Abbreviated terms .....	9
1.4	Overview .....	10
2	Introduction to IEC 61131-3.....	10
2.1	General considerations.....	10
2.2	Overcoming historical limitations.....	12
2.3	Basic features in IEC 61131-3 .....	13
2.4	New features in the second edition of IEC 61131-3 .....	14
2.5	Software engineering considerations .....	14
2.5.1	Application of software engineering principles .....	14
2.5.2	Portability .....	17
3	Application guidelines.....	17
3.1	Use of data types .....	17
3.1.1	Type versus variable initialization.....	18
3.1.2	Use of enumerated and subrange types .....	18
3.1.3	Use of BCD data.....	19
3.1.4	Use of REAL data types .....	21
3.1.5	Use of character string data types.....	21
3.1.6	Use of time data types .....	22
3.1.7	Declaration and use of multi-element variables.....	22
3.1.8	Use of bit-string functions .....	23
3.1.9	Strongly typed assignment.....	24
3.2	Data passing .....	25
3.2.1	Global and external variables.....	25
3.2.2	In-out (VAR_IN_OUT) variables .....	26
3.2.3	Formal and non-formal invocations and argument lists .....	28
3.3	Use of function blocks .....	30
3.3.1	Function block types and instances.....	30
3.3.2	Scope of data within function blocks.....	31
3.3.3	Function block access and invocation .....	32
3.4	Differences between function block instances and functions.....	33
3.5	Use of indirectly referenced function block instances .....	33
3.5.1	Establishing an indirect function block instance reference .....	34
3.5.2	Access to indirectly referenced function block instances.....	35
3.5.3	Invocation of indirectly referenced function block instances.....	35
3.5.4	Recursion of indirectly referenced function block instances .....	38
3.5.5	Execution control of indirectly referenced function block instances .....	38
3.5.6	Use of indirectly referenced function block instances in functions .....	38
3.6	Recursion within programmable controller programming languages.....	39
3.7	Single and multiple invocation .....	39

3.8	Language specific features .....	40
3.8.1	Edge-triggered functionality .....	40
3.8.2	Use of EN/ENO in functions and function blocks .....	41
3.8.3	Use of non-IEC 61131-3 languages .....	42
3.9	Use of SFC elements .....	42
3.9.1	Action control .....	42
3.9.2	Boolean actions .....	44
3.9.3	Non-SFC actions .....	47
3.9.4	SFC actions .....	48
3.9.5	SFC function blocks .....	48
3.9.6	“Indicator” variables .....	49
3.10	Scheduling, concurrency, and synchronization mechanisms .....	50
3.10.1	Operating system issues .....	50
3.10.2	Task scheduling .....	51
3.10.3	Semaphores .....	52
3.10.4	Messaging .....	53
3.10.5	Time stamping .....	53
3.11	Communication facilities in ISO/IEC 9506/5 and IEC 61131-5 .....	54
3.11.1	Communication channels .....	54
3.11.2	Reading and writing variables .....	54
3.11.3	Communication function blocks .....	55
3.12	Deprecated programming practices .....	56
3.12.1	Global variables .....	56
3.12.2	Jumps in FBD language .....	56
3.12.3	Multiple invocations of function block instances in FBD .....	56
3.12.4	Coupling of SFC networks .....	56
3.12.5	Dynamic modification of task priorities .....	57
3.12.6	Execution control of function block instances by tasks .....	57
3.12.7	Incorrect use of WHILE and REPEAT constructs .....	57
3.13	Use of TRUNC and REAL_TO_INT functions .....	58
4	Implementation guidelines .....	58
4.1	Resource allocation .....	59
4.2	Implementation of data types .....	59
4.2.1	REAL and LREAL data types .....	59
4.2.2	Bit strings .....	59
4.2.3	Character strings .....	59
4.2.4	Time data types .....	60
4.2.5	Multi-element variables .....	60
4.3	Execution of functions and function blocks .....	60
4.3.1	Functions .....	60
4.3.2	Function blocks .....	61
4.4	Implementation of SFCs .....	62
4.4.1	General considerations .....	62
4.4.2	SFC evolution .....	62
4.5	Task scheduling .....	63
4.5.1	Classification of tasks .....	63
4.5.2	Task priorities .....	63

4.6	Error handling .....	64
4.6.1	Error-handling mechanisms.....	64
4.6.2	Run-time error-handling procedures .....	65
4.7	System interface .....	67
4.8	Compliance .....	67
4.8.1	Compliance statement .....	67
4.8.2	Controller instruction sets .....	68
4.8.3	Compliance testing .....	68
5	PSE requirements .....	68
5.1	User interface.....	68
5.2	Programming of programs, functions and function blocks .....	69
5.3	Application design and configuration.....	70
5.4	Separate compilation .....	70
5.5	Separation of interface and body .....	71
5.5.1	Invocation of a function from a programming unit.....	71
5.5.2	Declaration and invocation of a function block instance .....	72
5.6	Linking of configuration elements with programs .....	73
5.7	Library management.....	75
5.8	Analysis tools .....	75
5.8.1	Simulation and debugging .....	75
5.8.2	Performance estimation .....	76
5.8.3	Feedback loop analysis.....	76
5.8.4	SFC analysis .....	76
5.9	Documentation requirements .....	79
5.10	Security of data and programs .....	79
5.11	On-line facilities .....	79
	Annex A (informative) Changes to IEC 61131-3, Second edition.....	80
	Annex B (informative) Software quality measures.....	90
	Annex C (informative) Relationships to other standards.....	92
	INDEX.....	93
	Bibliography .....	105
	Figure 1 – A distributed application .....	11
	Figure 2 – Stand-alone applications .....	11
	Figure 3 – Cyclic or periodic scanning of a program .....	12
	Figure 4 – Function block BCD_DIFF .....	20
	Figure 5 – Function block SBCD_DIFFF .....	20
	Figure 6 – ST example of time data type usage.....	22
	Figure 7 – Example of declaration and use of “anonymous array types” .....	23
	Figure 8 – Examples of VAR_IN_OUT usage.....	28
	Figure 9 – Hiding of function block instances .....	32
	Figure 10 – Graphical use of a function block name .....	35
	Figure 11 – Access to an indirectly referenced function block instance .....	35

Figure 12 – Invocation of an indirectly referenced function block instance .....	37
Figure 13 – Timing of edge triggered functionality .....	40
Figure 14 – Execution control example.....	42
Figure 15 – Timing of Boolean actions .....	47
Figure 16 – Example of a programmed non-Boolean action .....	47
Figure 17 – Use of the pulse (P) qualifier .....	48
Figure 18 – An SFC function block.....	49
Figure 19 – Example of incorrect and allowed programming constructs .....	58
Figure 20 – Essential phases of POU creation .....	69
Figure 21 – Essential phases of application creation .....	70
Figure 22 – Separate compilation of functions and function blocks .....	70
Figure 23 – Compiling a program accessing external or directly represented variables .....	71
Figure 24 – Compiling a function that invokes another function.....	71
Figure 25 – Compiling a program containing local instances of function blocks .....	72
Figure 26 – Separate compilation example.....	73
Figure 27 – The configuration process .....	74
Figure 28 – Reduction steps .....	77
Figure 29 – Reduction of SFCs .....	78
Table 1 – IEC 61131-3 elements supporting encapsulation and hiding.....	15
Table 2 – Examples of textual invocations of functions and function blocks .....	29
Table 3 – Differences between multi-user and real-time systems.....	51
Table 4 – Recommended run-time error-handling mechanisms.....	64
Table A.1 – Changes in usage to achieve program compliance .....	89

## INTRODUCTION

This part of IEC 61131 is being issued as a technical report in order to provide guidelines for the implementation and application of the programming languages defined in IEC 61131-3: 2003, second edition.

Its contents answer a number of frequently asked questions about the intended application and implementation of the normative provisions of IEC 61131-3, second edition and about its differences from IEC 61131-3:1993, first edition.

## AUSTRALIAN STANDARD

**Programmable controllers****Part 8:  
Guidelines for the application and implementation of programming  
languages****1 General****1.1 Scope**

This part of IEC 61131, which is a technical report, applies to the programming of programmable controller systems using the programming languages defined in IEC 61131-3. It also provides guidelines for the implementation of these languages in programmable controller systems and their programming support environments (PSEs).

IEC 61131-4 should be consulted for other aspects of the application of programmable controller systems.

NOTE Neither IEC 61131-3 nor this technical report explicitly addresses safety issues of programmable controller systems or their associated software. The various parts of IEC 61508 should be consulted for such considerations.

**1.2 Normative references**

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-1:1992, *Programmable controllers – Part 1: General information*

IEC 61131-2:2003, *Programmable controllers – Part 2: Equipment requirements and tests*

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC 61131-5:2000, *Programmable controllers – Part 5: Communications*

**1.3 Abbreviated terms**

FB	Function Block
FBD	Function Block Diagram
LD	Ladder Diagram
IL	Instruction List
POU	Program Organization Unit
PSE	Programming Support Environment
SFC	Sequential Function Chart
ST	Structured Text