

Australian Standard™

## **Programmable controllers**

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

This Australian Standard was prepared by Committee IT-006, Information Technology for Industrial Automation. It was approved on behalf of the Council of Standards Australia on 15 January 2004 and published on 22 March 2004.

---

The following are represented on Committee IT-006:

Association of Consulting Engineers Australia  
Australian Electrical and Electronic Manufacturers Association  
CSIRO Centre for Planning and Design  
CSIRO Manufacturing and Infrastructure Technology  
Department of Defence (Australia)  
Institute of Instrumentation, Control and Automation, Australia  
Institution of Engineers Australia  
Monash University  
RMIT University  
The University of Melbourne

---

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

Standards are living documents which 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 Standards may also be withdrawn.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 which may have been published since the Standard was purchased.

Detailed information about Standards can be found by visiting the Standards Web Shop at [www.standards.com.au](http://www.standards.com.au) and looking up the relevant Standard in the on-line catalogue.

Alternatively, the printed Catalogue provides information current at 1 January each year, and the monthly magazine, *The Global Standard*, has a full listing of revisions and amendments published each month.

Australian Standards™ and other products and services developed by Standards Australia are published and distributed under contract by SAI Global, which operates the Standards Web Shop.

We also welcome suggestions for improvement in our Standards, and especially encourage 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 the Chief Executive, Standards Australia International Ltd, GPO Box 5420, Sydney, NSW 2001.

---

Australian Standard™

## **Programmable controllers**

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

First published as AS IEC 61131.8—2004.

#### **COPYRIGHT**

© Standards Australia International

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.

Published by Standards Australia International Ltd  
GPO Box 5420, Sydney, NSW 2001, Australia

ISBN 0 7337 5776 6

## PREFACE

This Standard was prepared by the Standards Australia Committee IT-006, Information Technology for Industrial Automation.

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

This Standard is identical with, and has been reproduced from, IEC 61131-8:2000, *Programmable controllers—Part 8: Guidelines for the application and implementation of programming languages*.

This Standard is Part 8 of AS IEC 61131 *Programmable controllers*, which consists of the following:

Part 1: General information

Part 2: Equipment requirements and tests

Part 3: Programming languages

Part 4: User guidelines

Part 5: Communications

Part 7: Fuzzy control programming

Part 8: Guidelines for the application and implementation of programming languages (this Standard)

AS IEC 61131 does not have a Part 6. A project to develop IEC 61131-6 *Programmable controller communications via field bus* was deleted in September 2000 by the IEC.

In this Standard, the following print types are used:

- requirements proper: in arial type;
- *test specifications: in italic type;*
- explanatory matter: in smaller arial type.

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

- (a) Its number does not appear on each page of text and its identity is shown only on the cover and title page.
- (b) In the source text ‘this part of IEC 61131’ should read ‘this part of AS IEC 61131’.
- (c) A full point should be substituted for a comma when referring to a decimal marker.

## CONTENTS

	<i>Page</i>
1 General.....	1
1.1 Scope .....	1
1.2 Reference documents .....	1
1.3 Overview.....	3
2 Introduction to IEC 61131-3 .....	3
2.1 General considerations .....	3
2.2 Historical limitations.....	5
2.3 New features in IEC 61131-3 .....	6
2.4 Software engineering considerations.....	7
2.4.1 Software quality measures .....	7
2.4.2 Application of software engineering principles.....	8
2.4.3 Portability.....	10
3 Application guidelines .....	11
3.1 Use of data types.....	11
3.1.1 Type vs. variable initialization .....	11
3.1.2 Use of enumerated and subrange types .....	11
3.1.3 Use of BCD data .....	12
3.1.4 Use of REAL data types .....	14
3.1.5 Use of character string data types.....	14
3.1.6 Use of time data types .....	14
3.1.7 Use of multi-element variables .....	15
3.2 Data import and export .....	15
3.2.1 Global and external variables.....	16
3.2.2 Input/output (VAR_IN_OUT) variables.....	16
3.3 Use of function blocks.....	18
3.3.1 Function block types and instances .....	18
3.3.2 Scope of data within function blocks.....	19
3.3.3 Function block access and invocation .....	20
3.4 Differences between function block instances and functions .....	21
3.5 Use of indirectly referenced function block instances .....	21
3.5.1 Establishing an indirect function block instance reference .....	21
3.5.2 Access to indirectly referenced function block instances .....	23
3.5.3 Invocation of indirectly referenced function block instances.....	23
3.5.4 Recursion of indirectly referenced function block instances.....	26
3.5.5 Execution control of indirectly referenced function block instances.....	26
3.5.6 Use of indirectly referenced function block instances in functions.....	26
3.6 Recursion within programmable controller programming languages .....	26
3.7 Single and multiple invocation.....	27
3.8 Language specific features .....	28
3.8.1 Edge triggered functionality.....	28
3.8.2 Use of EN/ENO in functions and function blocks .....	29
3.8.3 Use of non-IEC 61131-3 languages.....	30

	<i>Page</i>
3.9 Use of SFC elements .....	30
3.9.1 Action control .....	31
3.9.2 Boolean actions .....	32
3.9.3 Non-SFC actions .....	36
3.9.4 SFC actions .....	37
3.9.5 SFC function blocks .....	39
3.9.6 "Indicator" variables .....	39
3.10 Scheduling, concurrency and synchronization mechanisms .....	40
3.10.1 Operating system issues .....	40
3.10.2 Task scheduling .....	42
3.10.3 Semaphores .....	43
3.10.4 Messaging .....	44
3.10.5 Time stamping .....	45
3.11 Communication facilities in ISO/IEC 9506-5 and IEC 61131-5 .....	45
3.11.1 Communication channels .....	45
3.11.2 Reading and writing variables .....	46
3.11.3 Communication function blocks .....	46
3.12 Recommended programming practices .....	47
3.12.1 Global variables .....	47
3.12.2 Jumps in function block diagram (FBD) language .....	47
3.12.3 Multiple invocations of function block instances in FBD .....	47
3.12.4 Coupling of sequential function chart (SFC) networks .....	47
3.12.5 Dynamic modification of task priorities .....	48
3.12.6 Execution control of function block instances by tasks .....	48
3.12.7 Use of RTC (real time clock) function blocks .....	48
4 Implementation guidelines .....	49
4.1 Resource allocation .....	49
4.2 Implementation of data types .....	49
4.2.1 REAL and LREAL data types .....	49
4.2.2 Character strings .....	50
4.2.3 Time data types .....	50
4.2.4 Multi-element variables .....	50
4.3 Execution of functions and function blocks .....	51
4.3.1 Functions .....	51
4.3.2 Function blocks .....	51
4.4 Implementation of sequential function charts (SFCs) .....	52
4.5 Task scheduling .....	53
4.5.1 Classification of tasks .....	53
4.5.2 Task priorities .....	53
4.6 Error handling .....	54
4.6.1 Error handling mechanisms .....	54
4.6.2 Run-time error handling procedures .....	56
4.7 System interface .....	58
4.8 Compliance .....	58
4.8.1 Compliance statement .....	58
4.8.2 Controller instruction sets .....	58
4.8.3 Compliance testing .....	59
4.9 Compatibility with IEC 60617-12, IEC 60617-13 and IEC 60848 .....	59

	<i>Page</i>
5 Programming support environment (PSE) requirements .....	59
5.1 User interface .....	59
5.2 Programming of programs, functions and function blocks .....	60
5.3 Application design and configuration .....	60
5.4 Separate compilation .....	61
5.5 Interface/body separation.....	62
5.5.1 Invocation of a function from a programming unit .....	62
5.5.2 Declaration and invocation of a function block instance.....	62
5.6 Linking of configuration elements with programs .....	63
5.7 Library management .....	66
5.8 Analysis tools.....	67
5.8.1 Simulation and debugging .....	67
5.8.2 Performance estimation .....	67
5.8.3 Feedback loop analysis.....	67
5.8.4 SFC analysis.....	67
5.9 Documentation requirements .....	70
5.10 Security of data and programs .....	70
5.11 On-line facilities .....	70
Bibliography .....	71
Index.....	72
Figure 1 – Distributed application .....	4
Figure 2 – Stand-alone applications .....	4
Figure 3 – Scanning a cyclic program.....	5
Figure 4 – Function block BCD_DIFF a) External interface, b) Body .....	13
Figure 5 – Function block SBCD_DIFF a) Definition of structured data type SBCD_BYTE b) External interface c) Body .....	13
Figure 6 – Structured text (ST) example of time data type usage.....	15
Figure 7 – Example of array usage.....	15
Figure 8 – Examples of VAR_IN_OUT usage.....	17
Figure 9 – Hiding of function block instances a) Declaration of contained function block type b) Declaration of containing program type c) Visibility of function block instances.....	20
Figure 10 – Establishing an indirectly referenced function block instance a) Establishment as input variable b) Establishment as input/output variable c) Establishment as external variable .....	22
Figure 11 – Access to an indirectly referenced function block instance.....	23
Figure 12 – Invocation of an indirectly referenced function block instance a) Textual declaration and invocation; b) Graphical invocation; c) Textual passing of instance name; d) Graphical passing of instance name .....	25
Figure 13 – Timing of edge triggered functionality a) Example network b) Worst-case timing.....	28
Figure 14 – Execution control example (see clause F.4 of IEC 61131-3) a) ST language without EN/ENO b) FBD language with EN/ENO.....	30
Figure 15 – Example of an action block .....	31
Figure 16 – Timing of Boolean actions a) N (non-stored) action b) P (pulse) action c) S/R (set/reset) action .....	33

	<i>Page</i>
Figure 17 – Example of a programmed non-Boolean action .....	36
Figure 18 – Use of pulse (P) qualifier a) SFC fragment b) Incorrect body for action A15 (ST language) c) Correct body for action A15 (ST language) d) Correct body for action A15 (FBD language) .....	37
Figure 19 – Example of an SFC action enabled by an SFC .....	38
Figure 20 – An SFC-function block a) External interface b) Body .....	40
Figure 21 – Essential phases of program organization unit (POU) creation .....	60
Figure 22 – Essential phases of application creation .....	60
Figure 23 – Separate compilation of functions and function blocks without external reference .....	61
Figure 24 – Compilation of a program which access to external or directly represented variables .....	61
Figure 25 – Compilation of a function whose body contains an invocation of another function a) Declaration of example function b) Compilation of example function .....	62
Figure 26 – Compilation of a program containing local instances of function blocks a) Declaration of example program b) Compilation of example program .....	63
Figure 27 – Separate compilation example a) Sketch of programming units to be compiled b) Separate compilation .....	65
Figure 28 – Configuration process a) Declaration of the configuration b) Production of the linked application .....	66
Figure 29 – Reduction steps .....	68
Figure 30 – Reduction of SFCs a) No reduction required as per termination rule 1 b) No reduction required as per termination rule 2 c) An irreducible (unsafe) SFC d) An irreducible (unreachable) SFC .....	69
Table 1 – IEC 61131-3 elements supporting encapsulation and hiding .....	8
Table 2 – Differences between multi-user and real-time systems .....	42
Table 3 – Recommended error handling mechanisms .....	55

## STANDARDS AUSTRALIA

## Australian Standard

## Programmable controllers

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

**1 General****1.1 Scope**

This 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. IEC 61508-1 and IEC 61508-7 should be consulted for such considerations.

**1.2 Reference documents**

References to international standards that are struck through in this clause are replaced by references to Australian or Australian/New Zealand Standards that are listed immediately thereafter and identified by shading. Any Australian or Australian/New Zealand Standard that is identical to the International Standard it replaces is identified as such.

~~IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*~~

~~IEC 60617-12:1991, *Graphical symbols for diagrams — Part 12: Binary logic elements*~~

AS/NZS 1102.112:1995, *Graphical symbols for electrotechnology—Binary logic elements* (identical to IEC 60617-12:1991)

~~IEC 60617-13:1993, *Graphical symbols for diagrams — Part 13: Analogue elements*~~

AS/NZS 1102.113:1995, *Graphical symbols for electrotechnology—Analogue elements* (identical to IEC 60617-13:1993)

~~IEC 60848:1988, *Preparation of functional charts for control systems*~~

AS/NZS 4382:1996, *Preparation of function charts for control systems* (identical to IEC 60848:1988)

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

AS IEC 61131.1, *Programmable controllers – Part 1: General information* (identical to IEC 61131.1:2003)