

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Programmable controllers –
Part 3: Programming languages**

**Automates programmables –
Partie 3: Langages de programmation**

IEC 61131-3:2003

<https://standards.iteh.ai/catalog/standards/iec/34e7370d-d9cf-4237-a2b4-ff7fcd2b8c94/iec-61131-3-2003>



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2003 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 60 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 60 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Programmable controllers –
Part 3: Programming languages**

**Automates programmables –
Partie 3: Langages de programmation**

<https://standards.iteh.ai/catalog/standards/iec/34e7370d-d9cf-4237-a2b4-ff7fcd2b8c94/iec-61131-3-2003>

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

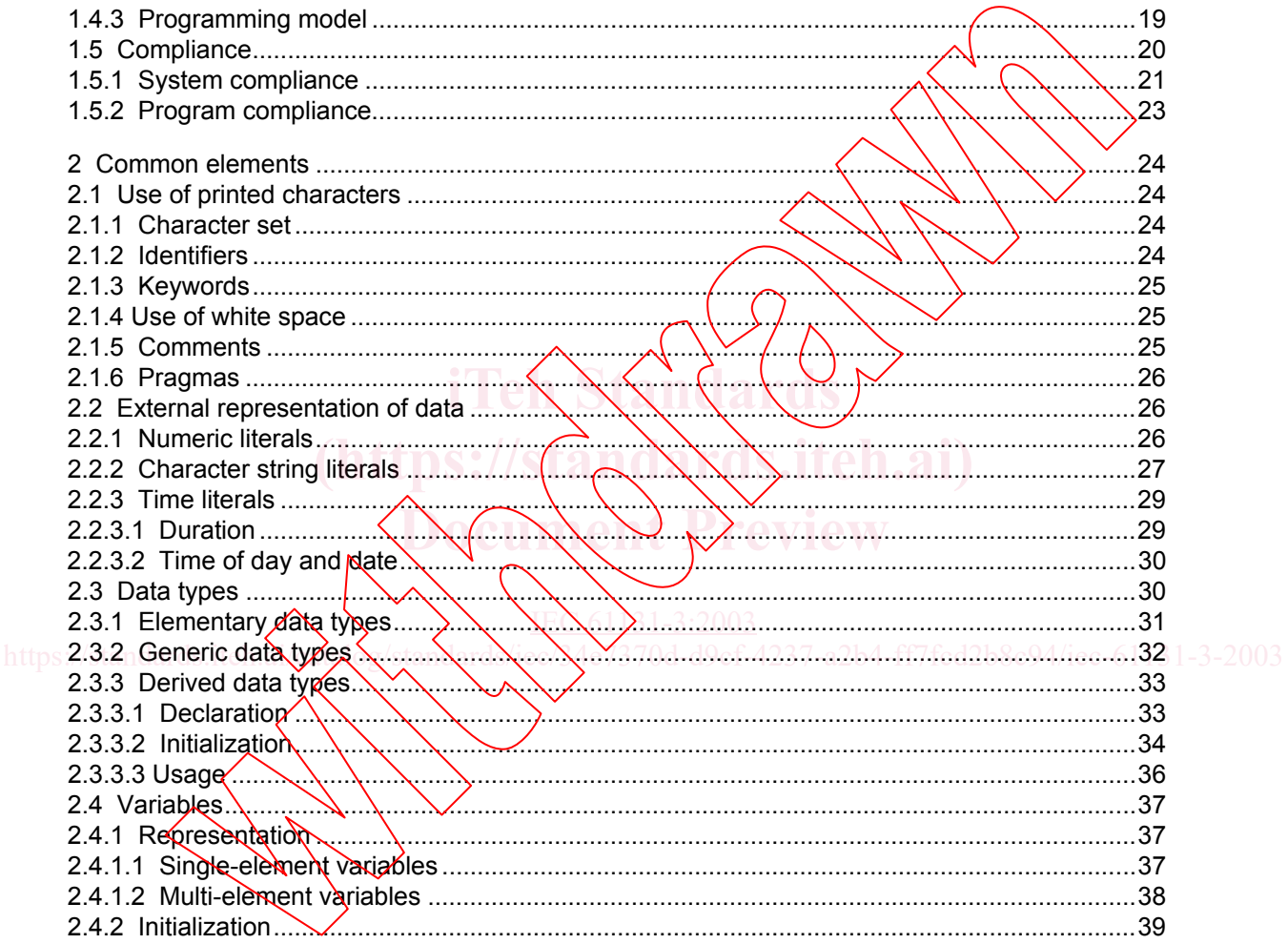
ICS 25.040; 35.240.50

ISBN 978-2-8322-2616-2

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD	8
1 General	9
1.1 Scope	10
1.2 Normative references	10
1.3 Definitions	10
1.4 Overview and general requirements	15
1.4.1 Software model	15
1.4.2 Communication model	17
1.4.3 Programming model	19
1.5 Compliance	20
1.5.1 System compliance	21
1.5.2 Program compliance	23
2 Common elements	24
2.1 Use of printed characters	24
2.1.1 Character set	24
2.1.2 Identifiers	24
2.1.3 Keywords	25
2.1.4 Use of white space	25
2.1.5 Comments	25
2.1.6 Pragmas	26
2.2 External representation of data	26
2.2.1 Numeric literals	26
2.2.2 Character string literals	27
2.2.3 Time literals	29
2.2.3.1 Duration	29
2.2.3.2 Time of day and date	30
2.3 Data types	30
2.3.1 Elementary data types	31
2.3.2 Generic data types	32
2.3.3 Derived data types	33
2.3.3.1 Declaration	33
2.3.3.2 Initialization	34
2.3.3.3 Usage	36
2.4 Variables	37
2.4.1 Representation	37
2.4.1.1 Single-element variables	37
2.4.1.2 Multi-element variables	38
2.4.2 Initialization	39
2.4.3 Declaration	40
2.4.3.1 Type assignment	42
2.4.3.2 Initial value assignment	43
2.5 Program organization units	46
2.5.1 Functions	46
2.5.1.1 Representation	47
2.5.1.2 Execution control	50
2.5.1.3 Declaration	51
2.5.1.4 Typing, overloading, and type conversion	53
2.5.1.5 Standard functions	56
2.5.1.5.1 Type conversion functions	56
2.5.1.5.2 Numerical functions	57
2.5.1.5.3 Bit string functions	60
2.5.1.5.4 Selection and comparison functions	60
2.5.1.5.5 Character string functions	63
2.5.1.5.6 Functions of time data types	65



2.5.1.5.7 Functions of enumerated data types	67
2.5.2 Function blocks	67
2.5.2.1 Representation	68
2.5.2.1a) Use of EN and ENO in function blocks	69
2.5.2.2 Declaration	70
2.5.2.3 Standard function blocks	78
2.5.2.3.1 Bistable elements	78
2.5.2.3.2 Edge detection	79
2.5.2.3.3 Counters	79
2.5.2.3.4 Timers	82
2.5.2.3.5 Communication function blocks	83
2.5.3 Programs	83
2.6 Sequential Function Chart (SFC) elements	84
2.6.1 General	84
2.6.2 Steps	85
2.6.3 Transitions	87
2.6.4 Actions	90
2.6.4.1 Declaration	90
2.6.4.2 Association with steps	92
2.6.4.3 Action blocks	93
2.6.4.4 Action qualifiers	94
2.6.4.5 Action control	95
2.6.5 Rules of evolution	100
2.6.6 Compatibility of SFC elements	108
2.6.7 SFC Compliance requirements	109
2.7 Configuration elements	109
2.7.1 Configurations, resources, and access paths	111
2.7.2 Tasks	115
3 Textual languages	124
3.1 Common elements	124
3.2 Instruction list (IL)	124
3.2.1 Instructions	124
3.2.2 Operators, modifiers and operands	125
3.2.3 Functions and function blocks	127
3.3 Structured Text (ST)	130
3.3.1 Expressions	130
3.3.2 Statements	132
3.3.2.1 Assignment statements	134
3.3.2.2 Function and function block control statements	134
3.3.2.3 Selection statements	134
3.3.2.4 Iteration statements	135
4 Graphic languages	136
4.1 Common elements	136
4.1.1 Representation of lines and blocks	136
4.1.2 Direction of flow in networks	136
4.1.3 Evaluation of networks	137
4.1.4 Execution control elements	139
4.2 Ladder diagram (LD)	140
4.2.1 Power rails	140
4.2.2 Link elements and states	140
4.2.3 Contacts	141
4.2.4 Coils	141
4.2.5 Functions and function blocks	141
4.2.6 Order of network evaluation	142
4.3 Function Block Diagram (FBD)	144
4.3.1 General	144
4.3.2 Combination of elements	144

4.3.3 Order of network evaluation	144
ANNEX A (normative) Specification method for textual languages	145
A.1 Syntax	145
A.1.1 Terminal symbols	145
A.1.2 Non-terminal symbols	145
A.1.3 Production rules	146
A.2 Semantics	146
ANNEX B (normative) Formal specifications of language elements	147
B.0 Programming model	147
B.1 Common elements	147
B.1.1 Letters, digits and identifiers	147
B.1.2 Constants	148
B.1.2.1 Numeric literals	148
B.1.2.2 Character strings	148
B.1.2.3 Time literals	149
B.1.2.3.1 Duration	149
B.1.2.3.2 Time of day and date	149
B.1.3 Data types	150
B.1.3.1 Elementary data types	150
B.1.3.2 Generic data types	150
B.1.3.3 Derived data types	150
B.1.4 Variables	152
B.1.4.1 Directly represented variables	152
B.1.4.2 Multi-element variables	152
B.1.4.3 Declaration and initialization	153
B.1.5 Program organization units	155
B.1.5.1 Functions	155
B.1.5.2 Function blocks	156
B.1.5.3 Programs	157
B.1.6 Sequential function chart elements	157
B.1.7 Configuration elements	158
B.2 Language IL (Instruction List)	160
B.2.1 Instructions and operands	160
B.2.2 Operators	160
B.3 Language ST (Structured Text)	161
B.3.1 Expressions	161
B.3.2 Statements	161
B.3.2.1 Assignment statements	162
B.3.2.2 Subprogram control statements	162
B.3.2.3 Selection statements	162
B.3.2.4 Iteration statements	162
ANNEX C (normative) Delimiters and keywords	163
ANNEX D (normative) Implementation-dependent parameters	166
ANNEX E (normative) Error conditions	168
ANNEX F (informative) Examples	170
F.1 Function WEIGH	170
F.2 Function block CMD_MONITOR	171
F.3 Function block FWD_REV_MON	174
F.4 Function block STACK_INT	179
F.5 Function block MIX_2_BRX	184
F.6 Analog signal processing	187
F.6.1 Function block LAG1	188
F.6.2 Function block DELAY	188

F.6.3 Function block AVERAGE	189
F.6.4 Function block INTEGRAL.....	189
F.6.5 Function block DERIVATIVE	190
F.6.6 Function block HYSTERESIS	190
F.6.7 Function block LIMITS_ALARM.....	191
F.6.8 Structure ANALOG_LIMITS.....	191
F.6.9 Function block ANALOG_MONITOR.....	192
F.6.10 Function block PID.....	193
F.6.11 Function block DIFFEQ.....	194
F.6.12 Function block RAMP	195
F.6.13 Function block TRANSFER.....	196
F.7 Program GRAVEL.....	196
F.8 Program AGV	204
F.9 Use of enumerated data types	207
F.10 Function block RTC (Real Time Clock).....	207
F.11 Function block ALRM_INT.....	207
ANNEX G (informative) Reference character set.....	209
Index	211
Table 1 - Character set features.....	24
Table 2 - Identifier features.....	25
Table 3 - Comment feature.....	26
Table 3a - Pragma feature.....	26
Table 4 - Numeric literals.....	27
Table 5 - Character string literal features.....	28
Table 6 - Two-character combinations in character strings	29
Table 7 - Duration literal features.....	30
Table 8 - Date and time of day literals.....	30
Table 9 - Examples of date and time of day literals	30
Table 10 - Elementary data types	31
Table 11 - Hierarchy of generic data types	33
Table 12 - Data type declaration features	34
Table 13 - Default initial values of elementary data types.....	35
Table 14 - Data type initial value declaration features	36
Table 15 - Location and size prefix features for directly represented variables.....	38
Table 16a - Variable declaration keywords	40
Table 16b - Usages of VAR_GLOBAL, VAR_EXTERNAL and CONSTANT declarations	42
Table 17 - Variable type assignment features.....	42
Table 18 - Variable initial value assignment features.....	44
Table 19 - Graphical negation of Boolean signals	48
Table 19a - Textual invocation of functions for formal and non-formal argument list	50
Table 20 - Use of EN input and ENO output	51
Table 20a - Function features	52
Table 21 - Typed and overloaded functions.....	54
Table 22 - Type conversion function features	56
Table 23 - Standard functions of one numeric variable.....	58
Table 24 - Standard arithmetic functions.....	59

Table 25 - Standard bit shift functions	60
Table 26 - Standard bitwise Boolean functions	61
Table 27 - Standard selection functions ^d	62
Table 28 - Standard comparison functions	63
Table 29 - Standard character string functions	64
Table 30 - Functions of time data types	65
Table 31 - Functions of enumerated data types	67
Table 32 - Examples of function block I/O variable usage	69
Table 33 - Function block declaration and usage features	72
Table 34 - Standard bistable function blocks ^a	78
Table 35 - Standard edge detection function blocks	79
Table 36 - Standard counter function blocks	80
Table 37 - Standard timer function blocks	82
Table 38 - Standard timer function blocks - timing diagrams	82
Table 39 - Program declaration features	84
Table 40 - Step features	86
Table 41 - Transitions and transition conditions	88
Table 42 - Declaration of actions ^{a,b}	91
Table 43 - Step/action association	93
Table 44 - Action block features	94
Table 45 - Action qualifiers	95
Table 45a - Action control features	99
Table 46 - Sequence evolution	102
Table 47 - Compatible SFC features	109
Table 48 - SFC minimal compliance requirements	109
Table 49 - Configuration and resource declaration features	113
Table 50 - Task features	117
Table 51a - Examples of instruction fields	125
Table 51b - Parenthesized expression features for IL language	126
Table 52 - Instruction List operators	126
Table 53 - Function Block invocation and Function invocation features for IL language	128
Table 54 - Standard Function Block input operators for IL language	130
Table 55 - Operators of the ST language	132
Table 56 - ST language statements	133
Table 57 - Representation of lines and blocks	137
Table 58 - Graphic execution control elements	139
Table 59 - Power rails	140
Table 60 - Link elements	141
Table 61 - Contacts ^a	142
Table 62 - Coils	143
Table C.1 - Delimiters	163
Table C.2 - Keywords	164
Table D.1 - Implementation-dependent parameters	166
Table E.1 - Error conditions	168

Table G.1 - Character representations	209
Table G.2 - Character encodings	210
Figure 1 - Software model	16
Figure 2 a) - Data flow connection within a program.....	17
Figure 2 b) - Communication via GLOBAL variables.....	17
Figure 2 c) - Communication function blocks	18
Figure 2 d) - Communication via access paths	18
Figure 3 - Combination of programmable controller language elements	20
Figure 4 - Examples of function usage.....	46
Figure 5 - Use of formal argument names.....	49
Figure 6 - Examples of function declarations and usage	53
Figure 7 - Examples of explicit type conversion with overloaded functions	55
Figure 8 - Examples of explicit type conversion with typed functions	55
Figure 9 - Function block instantiation examples	68
Figure 10 - Examples of function block declarations.....	71
Figure 11 a) - Graphical use of a function block name as an input variable.....	74
Figure 11 b) - Graphical use of a function block name as an in-out variable.....	75
Figure 11 c) - Graphical use of a function block name as an external variable	76
Figure 12 - Declaration and usage of in-out variables in function blocks.....	77
Figure 14 - ACTION_CONTROL function block - External interface (Not visible to the user)	96
Figure 15 a) - ACTION_CONTROL function block body with "final scan" logic.....	97
Figure 15 b) - ACTION_CONTROL function block body without "final scan" logic.....	98
Figure 16 a) - Action control example - SFC representation.....	99
Figure 16 b) - Action control example - functional equivalent	100
Figure 17 - Examples of SFC evolution rules.....	106
Figure 18 a) - Examples of SFC errors: an "unsafe" SFC	107
Figure 18 b) - Examples of SFC errors: an "unreachable" SFC.....	108
Figure 19 a) - Graphical example of a configuration	110
Figure 19 b) - Skeleton function block and program declarations for configuration example	111
Figure 20 - Examples of CONFIGURATION and RESOURCE declaration features.....	114
Figure 21 a) - Synchronization of function blocks with explicit task associations	121
Figure 21 b) - Synchronization of function blocks with implicit task associations	122
Figure 21 c) - Explicit task associations equivalent to figure 21 b)	123
Figure 22 - EXIT statement example	135
Figure 23 - Feedback path example.....	138
Figure 24 - Boolean OR examples	144

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PROGRAMMABLE CONTROLLERS –

Part 3: Programming languages

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61131-3 has been prepared by subcommittee 65B: Devices, of IEC technical committee 65: Industrial-process measurement and control.

<https://standards.iec.ch/standards/iec/61131-3/61131-3-2003>

This bilingual version (2015-06) corresponds to the English version, published in 2003-01.

The text of this standard is based on the following documents:

FDIS	Report on voting
65B/456/FDIS	65B/465/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This second edition of IEC 61131-3 cancels and replaces the first edition, published in 1993, and constitutes a technical revision.

This International Standard has been reproduced without significant modification to its original contents or drafting.

The French version of this standard has not been voted upon.

The committee has decided that the contents of this publication will remain unchanged until 2007. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

Withdawn

iTech Standards
(<https://standards.iteh.ai>)
Document Preview

[IEC 61131-3:2003](https://standards.iteh.ai/standards/iec/61131-3-2003)

<https://standards.iteh.ai/standards/iec/61131-3-2003>

PROGRAMMABLE CONTROLLERS –

Part 3: Programming languages

1 General

1.1 Scope

This part of IEC 61131 specifies syntax and semantics of programming languages for *programmable controllers* as defined in part 1 of IEC 61131.

The functions of program entry, testing, monitoring, operating system, etc., are specified in Part 1 of IEC 61131.

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 60050 (all parts): *International Electrotechnical Vocabulary (IEV)*

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

IEC 60617-12:1997, *Graphical symbols for diagrams – Part 12: Binary logic elements*

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

IEC 60848:2002, *GRAFCET specification language for sequential function charts*

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

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

ISO/AFNOR: 1989, *Dictionary of computer science – The standardised vocabulary*

ISO/IEC 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

1.3 Definitions

For the purposes of this part of IEC 61131, the following definitions apply. Definitions applying to all parts of IEC 61131 are given in part 1.

NOTE 1 Terms defined in this subclause are *italicized* where they appear in the bodies of definitions.

NOTE 2 The notation “(ISO)” following a definition indicates that the definition is taken from the ISO/AFNOR Dictionary of computer science.

NOTE 3 The ISO/AFNOR Dictionary of computer science and the IEC 60050 should be consulted for terms not defined in this standard.

- 1.3.1 absolute time:** the combination of time of day and date information.
- 1.3.2 access path:** the association of a symbolic name with a variable for the purpose of open communication.
- 1.3.3 action:** Boolean variable, or a collection of operations to be performed, together with an associated control structure, as specified in 2.6.4.
- 1.3.4 action block:** graphical language element which utilizes a Boolean input variable to determine the value of a Boolean output variable or the enabling condition for an *action*, according to a predetermined control structure as defined in 2.6.4.5.
- 1.3.5 aggregate:** structured collection of data objects forming a *data type*. (ISO)
- 1.3.6 argument:** synonymous with *input variable*, *output variable* or *in-out variable*.
- 1.3.7 array:** *aggregate* that consists of data objects, with identical attributes, each of which may be uniquely referenced by *subscripting*. (ISO)
- 1.3.8 assignment:** mechanism to give a value to a variable or to an *aggregate*. (ISO)
- 1.3.9 based number:** number represented in a specified base other than ten.
- 1.3.10 bistable function block:** *function block* with two stable states controlled by one or more inputs.
- 1.3.11 bit string:** data element consisting of one or more bits.
- 1.3.12 body:** that portion of a *program organization unit* which specifies the operations to be performed on the declared *operands* of the program organization unit when its execution is *invoked*.
- 1.3.13 call:** language construct for *invoking* the execution of a *function* or *function block*.
- 1.3.14 character string:** *aggregate* that consists of an ordered sequence of characters.
- 1.3.15 comment:** language construct for the inclusion of text in a program and having no impact on the execution of the program. (ISO)
- 1.3.16 compile:** to translate a *program organization unit* or *data type* specification into its machine language equivalent or an intermediate form.
- 1.3.17 configuration:** language element corresponding to a *programmable controller system* as defined in IEC 61131-1.
- 1.3.18 counter function block:** *function block* which accumulates a value for the number of changes sensed at one or more specified *inputs*.
- 1.3.19 data type:** set of values together with a set of permitted operations. (ISO)
- 1.3.20 date and time:** the date within the year and the time of day represented as a single language element.
- 1.3.21 declaration:** the mechanism for establishing the definition of a *language element*. A declaration normally involves attaching an identifier to the language element, and allocating attributes such as *data types* and algorithms to it.

1.3.22 delimiter: character or combination of characters used to separate program *language elements*.

1.3.23 direct representation: means of representing a variable in a programmable controller program from which a manufacturer-specified correspondence to a physical or *logical location* may be determined directly.

1.3.24 double word: data element containing 32 bits.

1.3.25 evaluation: the process of establishing a value for an expression or a *function*, or for the *outputs* of a network or *function block*, during program execution.

1.3.26 execution control element: A *language element* which controls the flow of program execution.

1.3.27 falling edge: the change from 1 to 0 of a Boolean variable.

1.3.28 function (procedure): *program organization unit* which, when executed, yields exactly one data element and possibly additional *output variables* (which may be multi-valued, for example, an *array* or *structure*), and whose *invocation* can be used in textual languages as an *operand* in an expression.

1.3.29 function block instance (function block): *instance* of a *function block type*.

1.3.30 function block type: programmable controller programming *language element* consisting of:

- 1) the definition of a data structure partitioned into input, output, and internal variables; and
- 2) a set of operations to be performed upon the elements of the data structure when an *instance* of the function block type is *invoked*.

1.3.31 function block diagram: *network* in which the nodes are *function block instances*, graphically represented *functions (procedures), variables, literals, and labels*.

1.3.32 generic data type: *data type* which represents more than one type of data, as specified in 2.3.2.

1.3.33 global scope: scope of a declaration applying to all program organization units within a *resource* or *configuration*.

1.3.34 global variable: variable whose *scope* is *global*.

1.3.35 hierarchical addressing: the *direct representation* of a data element as a member of a physical or logical hierarchy, for example, a point within a module which is contained in a rack, which in turn is contained in a cubicle, etc.

1.3.36 identifier: combination of letters, numbers, and underline characters, as specified in 2.1.2, which begins with a letter or underline and which names a *language element*.

1.3.37 in-out variable: *variable* that is declared in a VAR_IN_OUT . . . END_VAR block.

1.3.38 initial value: the value assigned to a variable at system start-up.

1.3.39 input variable (input): variable which is used to supply an argument to a *program organization unit*.