



IEC 61131-3

Edition 3.0 2013-02

INTERNATIONAL STANDARD

NORME INTERNATIONALE

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

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 25.040; 35.240.50

ISBN 978-2-83220-661-4

**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.....	7
1 Scope.....	9
2 Normative references	9
3 Terms and definitions	9
4 Architectural models	18
4.1 Software model	18
4.2 Communication model	19
4.3 Programming model	20
5 Compliance	22
5.1 General	22
5.2 Feature tables	22
5.3 Implementer's compliance statement	22
6 Common elements	24
6.1 Use of printed characters	24
6.1.1 Character set.....	24
6.1.2 Identifiers	24
6.1.3 Keywords	24
6.1.4 Use of white space	25
6.1.5 Comments	25
6.2 Pragma	26
6.3 Literals – External representation of data	26
6.3.1 General	26
6.3.2 Numeric literals and string literals.....	26
6.3.3 Character string literals	28
6.3.4 Duration literal	29
6.3.5 Date and time of day literal	30
6.4 Data types.....	30
6.4.1 General	30
6.4.2 Elementary data types (BOOL, INT, REAL, STRING, etc.).....	30
6.4.3 Generic data types	33
6.4.4 User-defined data types.....	34
6.5 Variables	47
6.5.1 Declaration and initialization of variables	47
6.5.2 Variable sections	49
6.5.3 Variable length ARRAY variables	51
6.5.4 Constant variables.....	53
6.5.5 Directly represented variables (%)	54
6.5.6 Retentive variables (RETAIN, NON_RETAIN).....	56
6.6 Program organization units (POUs)	58
6.6.1 Common features for POU's	58
6.6.2 Functions.....	70
6.6.3 Function blocks	99
6.6.4 Programs.....	117
6.6.5 Classes	118

6.6.6	Interface	137
6.6.7	Object oriented features for function blocks	146
6.6.8	Polymorphism	152
6.7	Sequential Function Chart (SFC) elements	155
6.7.1	General	155
6.7.2	Steps	155
6.7.3	Transitions	157
6.7.4	Actions	160
6.7.5	Rules of evolution	168
6.8	Configuration elements	176
6.8.1	General	176
6.8.2	Tasks	180
6.9	Namespaces	186
6.9.1	General	186
6.9.2	Declaration	186
6.9.3	Usage	192
6.9.4	Namespace directive USING	192
7	Textual languages	195
7.1	Common elements	195
7.2	Instruction list (IL)	195
7.2.1	General	195
7.2.2	Instructions	195
7.2.3	Operators, modifiers and operands	196
7.2.4	Functions and function blocks	198
7.3	Structured Text (ST)	201
7.3.1	General	201
7.3.2	Expressions	201
7.3.3	Statements	203
8	Graphic languages	208
8.1	Common elements	208
8.1.1	General	208
8.1.2	Representation of variables and instances	209
8.1.3	Representation of lines and blocks	211
8.1.4	Direction of flow in networks	212
8.1.5	Evaluation of networks	213
8.1.6	Execution control elements	214
8.2	Ladder diagram (LD)	215
8.2.1	General	215
8.2.2	Power rails	216
8.2.3	Link elements and states	216
8.2.4	Contacts	216
8.2.5	Coils	218
8.2.6	Functions and function blocks	219
8.2.7	Order of network evaluation	219
8.3	Function Block Diagram (FBD)	219
8.3.1	General	219
8.3.2	Combination of elements	219
8.3.3	Order of network evaluation	220
Annex A (normative)	Formal specification of the languages elements	221

Annex B (informative) List of major changes and extensions of the third edition	228
Bibliography	229
Figure 1 – Software model	18
Figure 2 – Communication model	20
Figure 3 – Combination of programmable controller language elements	21
Figure 4 – Implementer's compliance statement (Example)	23
Figure 5 – Hierarchy of the generic data types	34
Figure 6 – Initialization by literals and constant expressions (Rules)	35
Figure 7 – Variable declaration keywords (Summary)	50
Figure 8 – Usage of <code>VAR_GLOBAL</code> , <code>VAR_EXTERNAL</code> and <code>CONSTANT</code> (Rules)	51
Figure 9 – Conditions for the initial value of a variable (Rules)	57
Figure 10 – Formal and non-formal representation of call (Examples)	63
Figure 11 – Data type conversion rules – implicit and/or explicit (Summary)	67
Figure 12 – Supported implicit type conversions	68
Figure 13 – Usage of function block input and output parameters (Rules)	108
Figure 14 – Usage of function block input and output parameters (Illustration of rules)	109
Figure 15 – Standard timer function blocks – timing diagrams (Rules)	116
Figure 16 – Overview of inheritance and interface implementation	119
Figure 17 – Inheritance of classes (Illustration)	128
Figure 18 – Interface with derived classes (Illustration)	138
Figure 19 – Inheritance of interface and class (Illustration)	143
Figure 20 – Function block with optional body and methods (Illustration)	149
Figure 21 – Inheritance of function block body with <code>SUPER()</code> (Example)	151
Figure 22 – <code>ACTION_CONTROL</code> function block – External interface (Summary)	165
Figure 23 – <code>ACTION_CONTROL</code> function block body (Summary)	166
Figure 24 – Action control (Example)	168
Figure 25 – SFC evolution (Rules)	174
Figure 26 – SFC errors (Example)	175
Figure 27 – Configuration (Example)	177
Figure 28 – <code>CONFIGURATION</code> and <code>RESOURCE</code> declaration (Example)	180
Figure 29 – Accessibility using namespaces (Rules)	189
Figure 30 – Common textual elements (Summary)	195
Table 1 – Character set	24
Table 2 – Identifiers	24
Table 3 – Comments	25
Table 4 – Pragma	26
Table 5 – Numeric literals	27
Table 6 – Character string literals	28
Table 7 – Two-character combinations in character strings	29
Table 8 – Duration literals	29
Table 9 – Date and time of day literals	30

Table 10 – Elementary data types	31
Table 11 – Declaration of user-defined data types and initialization	35
Table 12 – Reference operations	46
Table 13 – Declaration of variables	48
Table 14 – Initialization of variables	49
Table 15 – Variable-length <code>ARRAY</code> variables	52
Table 16 – Directly represented variables	54
Table 17 – Partial access of <code>ANY_BIT</code> variables	60
Table 18 – Execution control graphically using <code>EN</code> and <code>ENO</code>	65
Table 19 – Function declaration	72
Table 20 – Function call	74
Table 21 – Typed and overloaded functions	76
Table 22 – Data type conversion function	78
Table 23 – Data type conversion of numeric data types	80
Table 24 – Data type conversion of bit data types	82
Table 25 – Data type conversion of bit and numeric types	83
Table 26 – Data type conversion of date and time types	85
Table 27 – Data type conversion of character types	86
Table 28 – Numerical and arithmetic functions	87
Table 29 – Arithmetic functions	88
Table 30 – Bit shift functions	89
Table 31 – Bitwise Boolean functions	89
Table 32 – Selection functions ^d	90
Table 33 – Comparison functions	91
Table 34 – Character string functions	92
Table 35 – Numerical functions of time and duration data types	93
Table 36 – Additional functions of time data types <code>CONCAT</code> and <code>SPLIT</code>	94
Table 37 – Function for endianness conversion	98
Table 38 – Functions of enumerated data types	98
Table 39 – Validate functions	99
Table 40 – Function block type declaration	100
Table 41 – Function block instance declaration	104
Table 42 – Function block call	105
Table 43 – Standard bistable function blocks ^a	112
Table 44 – Standard edge detection function blocks	113
Table 45 – Standard counter function blocks	113
Table 46 – Standard timer function blocks	115
Table 47 – Program declaration	117
Table 48 – Class	120
Table 49 – Class instance declaration	122
Table 50 – Textual call of methods – Formal and non-formal parameter list	125
Table 51 – Interface	137
Table 52 – Assignment attempt	146

Table 53 – Object oriented function block	147
Table 54 – SFC step	156
Table 55 – SFC transition and transition condition	158
Table 56 – SFC declaration of actions	160
Table 57 – Step/action association	162
Table 58 – Action block.....	163
Table 59 – Action qualifiers.....	163
Table 60 – Action control features.....	168
Table 61 – Sequence evolution – graphical.....	169
Table 62 – Configuration and resource declaration	178
Table 63 – Task.....	182
Table 64 – Namespace	191
Table 65 – Nested namespace declaration options	192
Table 66 – Namespace directive <code>USING</code>	194
Table 67 – Parenthesized expression for IL language	197
Table 68 – Instruction list operators	197
Table 69 – Calls for IL language	199
Table 70 – Standard function block operators for IL language.....	201
Table 71 – Operators of the ST language.....	202
Table 72 – ST language statements.....	203
Table 73 – Graphic execution control elements.....	215
Table 74 – Power rails and link elements	216
Table 75 – Contacts.....	217
Table 76 – Coils.....	218

IECNORM.COM : Click to view the full PDF of IEC 61131-3 ed 3.0:2013

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PROGRAMMABLE CONTROLLERS –

Part 3: Programming languages

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of 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, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). 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. 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 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 IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. 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: Measurement and control devices, of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition of IEC 61131-3 cancels and replaces the second edition, published in 2003. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

This third edition is a compatible extension of the second edition. The main extensions are new data types and conversion functions, references, name spaces and the object oriented features of classes and function blocks. See Annex B.

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

FDIS	Report on voting
65B/858/FDIS	65B/863/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 publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all the parts in the IEC 61131 series, published under the general title *Programmable controllers* can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

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

IECNORM.COM : Click to view the full PDF of IEC 61131-3 ed 3.0:2013

PROGRAMMABLE CONTROLLERS –

Part 3: Programming languages

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.

This part of IEC 61131 specifies the syntax and semantics of a unified suite of programming languages for programmable controllers (PCs). This suite consists of two textual languages, Instruction List (IL) and Structured Text (ST), and two graphical languages, Ladder Diagram (LD) and Function Block Diagram (FBD).

An additional set of graphical and equivalent textual elements named Sequential Function Chart (SFC) is defined for structuring the internal organization of programmable controller programs and function blocks. Also, configuration elements are defined which support the installation of programmable controller programs into programmable controller systems.

In addition, features are defined which facilitate communication among programmable controllers and other components of automated systems.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. 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, *Programmable controllers – Part 1: General information*

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

ISO/IEC 10646:2012, *Information technology – Universal Coded Character Set (UCS)*

ISO/IEC/IEEE 60559, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

SOMMAIRE

AVANT-PROPOS	235
1 Domaine d'application	237
2 Références normatives	237
3 Termes et définitions	237
4 Modèles architecturaux	246
4.1 Modèle logiciel	246
4.2 Modèle de communication	248
4.3 Modèle de programmation	249
5 Conformité	251
5.1 Généralités	251
5.2 Tableaux de caractéristiques	252
5.3 Déclaration de conformité de l'Intégrateur	252
6 Eléments communs	254
6.1 Utilisation des caractères d'impression	254
6.1.1 Jeu de caractères	254
6.1.2 Identificateurs	254
6.1.3 Mots-clés	255
6.1.4 Utilisation de l'espace blanc	255
6.1.5 Commentaires	255
6.2 Pragma	256
6.3 Littéraux – représentation externe de données	256
6.3.1 Généralités	256
6.3.2 Littéraux numériques et littéraux de chaîne	256
6.3.3 Littéraux de chaîne de caractères	258
6.3.4 Littéraux de durée	260
6.3.5 Littéraux de date et heure	261
6.4 Types de données	262
6.4.1 Généralités	262
6.4.2 Types de données élémentaires (BOOL, INT, REAL, STRING, etc.)	262
6.4.3 Types de données génériques	264
6.4.4 Types de données définis par l'utilisateur	265
6.5 Variables	279
6.5.1 Déclaration et initialisation de variables	279
6.5.2 Sections de variables	281
6.5.3 Variables <code>ARRAY</code> de longueur variable	283
6.5.4 Variables constantes	285
6.5.5 Variables directement représentées (%)	286
6.5.6 Variables persistantes (RETAIN, NON_RETAIN)	288
6.6 Unités d'organisation de programme (POU)	290
6.6.1 Caractéristiques communes pour les POU	290
6.6.2 Fonctions	302
6.6.3 Blocs fonctionnels	332
6.6.4 Programmes	352
6.6.5 Classes	353

6.6.6	Interface	372
6.6.7	Caractéristiques orientées objet pour les blocs fonctionnels	381
6.6.8	Polymorphisme	387
6.7	Eléments d'un diagramme fonctionnel séquentiel (SFC)	390
6.7.1	Généralités	390
6.7.2	Etapes	391
6.7.3	Transitions	392
6.7.4	Actions	395
6.7.5	Règles d'évolution	404
6.8	Eléments de configuration	412
6.8.1	Généralités	412
6.8.2	Tâches	416
6.9	Espaces de noms	422
6.9.1	Généralités	422
6.9.2	Déclaration	422
6.9.3	Utilisation	427
6.9.4	Directive d'espace de noms USING	428
7	Langages textuels	430
7.1	Eléments communs	430
7.2	Liste d'instructions (IL)	430
7.2.1	Généralités	430
7.2.2	Instructions	430
7.2.3	Opérateurs, modificateurs et opérandes	431
7.2.4	Fonctions et blocs fonctionnels	433
7.3	Texte structuré (ST)	436
7.3.1	Généralités	436
7.3.2	Expressions	436
7.3.3	Enoncés	438
8	Langages graphiques	444
8.1	Eléments communs	444
8.1.1	Généralités	444
8.1.2	Représentation de variables et d'instances	444
8.1.3	Représentation de traits et de blocs	446
8.1.4	Sens du flux dans les réseaux	447
8.1.5	Évaluation des réseaux	448
8.1.6	Eléments de contrôle d'exécution	449
8.2	Diagramme à contacts (LD)	450
8.2.1	Généralités	450
8.2.2	Rails de puissance	451
8.2.3	Eléments de liaison et états	451
8.2.4	Contacts	451
8.2.5	Bobines	453
8.2.6	Fonctions et blocs fonctionnels	454
8.2.7	Ordre d'évaluation des réseaux	454
8.3	Diagramme de bloc fonctionnel (FBD)	454
8.3.1	Généralités	454
8.3.2	Combinaison d'éléments	454
8.3.3	Ordre d'évaluation des réseaux	455
Annexe A (normative)	Spécification formelle des éléments de langage	456

Annexe B (informative) Liste des modifications et extensions majeures de la troisième édition	463
Bibliographie.....	464
Figure 1 – Modèle logiciel	247
Figure 2 – Modèle de communication	249
Figure 3 – Combinaison d'éléments de langage pour automate programmable	251
Figure 4 – Déclaration de conformité de l'Intégrateur (exemple).....	253
Figure 5 – Hiérarchie des types de données génériques	265
Figure 6 – Initialisation par des littéraux et des expressions constantes (règles)	266
Figure 7 – Mots-clés pour une déclaration de variable (résumé)	282
Figure 8 – Utilisation de <code>VAR_GLOBAL</code> , <code>VAR_EXTERNAL</code> et <code>CONSTANT</code> (règles)	283
Figure 9 – Conditions associées à la valeur initiale d'une variable (règles)	289
Figure 10 – Représentation formelle et informelle d'appel (exemples).....	295
Figure 11 – Règles de conversion d'un type de données – implicite et/ou explicite (résumé)	299
Figure 12 – Conversions de type implicites prises en charge	300
Figure 13 – Utilisation des paramètres d'entrée et de sortie de bloc fonctionnel (règles).....	343
Figure 14 – Utilisation des paramètres d'entrée et de sortie de bloc fonctionnel (illustration des règles)	344
Figure 15 – Blocs fonctionnels normalisés minuteur – diagrammes temporels (règles).....	351
Figure 16 – Présentation de la mise en œuvre d'héritage et d'interface.....	354
Figure 17 – Héritage de classes (illustration)	363
Figure 18 – Interface avec classes dérivées (illustration)	373
Figure 19 – Héritage d'interface et de classe (illustration)	378
Figure 20 – Bloc fonctionnel avec corps et méthodes facultatifs (illustration)	384
Figure 21 – Héritage de corps de bloc fonctionnel avec <code>SUPER()</code> (exemple)	386
Figure 22 – Bloc fonctionnel <code>ACTION_CONTROL</code> – Interface externe (résumé).....	400
Figure 23 – Corps de bloc fonctionnel <code>ACTION_CONTROL</code> (résumé)	402
Figure 24 – Contrôle d'action (exemple).....	404
Figure 25 – Evolution d'un SFC (règles).....	410
Figure 26 – Erreurs d'un SFC (exemple)	411
Figure 27 – Configuration (exemple)	413
Figure 28 – Déclaration de <code>CONFIGURATION</code> et de <code>RESOURCE</code> (exemple)	416
Figure 29 – Accessibilité à l'aide des espaces de noms (règles)	424
Figure 30 – Éléments textuels communs (résumé)	430
Tableau 1 – Jeu de caractères	254
Tableau 2 – Identificateurs.....	254
Tableau 3 – Commentaires	256
Tableau 4 – Pragma	256
Tableau 5 – Littéraux numériques	258
Tableau 6 – Littéraux de chaîne de caractères.....	259
Tableau 7 – Combinaisons de deux caractères dans les chaînes de caractères	260

Tableau 8 – Littéraux de durée	261
Tableau 9 – Littéraux de date et heure	261
Tableau 10 – Types de données élémentaires	262
Tableau 11 – Déclaration des types de données définis par l'utilisateur et initialisation	266
Tableau 12 – Opérations sur les références	278
Tableau 13 – Déclaration de variables	280
Tableau 14 – Initialisation de variables	281
Tableau 15 – Variables <code>ARRAY</code> de longueur variable	284
Tableau 16 – Variables directement représentées	286
Tableau 17 – Accès partiel aux variables <code>ANY_BIT</code>	292
Tableau 18 – Contrôle de l'exécution en utilisant graphiquement <code>EN</code> et <code>ENO</code>	297
Tableau 19 – Déclaration de fonction	304
Tableau 20 – Appel d'une fonction	306
Tableau 21 – Fonctions typées et en surcharge	308
Tableau 22 – Fonction de conversion de type de données	311
Tableau 23 – Conversion de type de données des types de données numériques	312
Tableau 24 – Conversion de type de données des types de données binaires	315
Tableau 25 – Conversion de type de données des types binaires et numériques	316
Tableau 26 – Conversion de type de données des types date et heure	319
Tableau 27 – Conversion de type de données des types caractère	319
Tableau 28 – Fonctions numériques et arithmétiques	320
Tableau 29 – Fonctions arithmétiques	321
Tableau 30 – Fonctions de décalage de bit	322
Tableau 31 – Fonctions booléennes au niveau du bit	322
Tableau 32 – Fonctions de sélection ^d	323
Tableau 33 – Fonctions de comparaison	324
Tableau 34 – Fonctions de chaîne de caractères	326
Tableau 35 – Fonctions numériques des types de données de temps et de durée	327
Tableau 36 – Fonctions additionnelles des types de données de temps <code>CONCAT</code> et <code>SPLIT</code>	328
Tableau 37 – Fonctions de conversion de boutisme	331
Tableau 38 – Fonctions des types de données énumérés	332
Tableau 39 – Fonctions de validation	332
Tableau 40 – Déclaration du type de bloc fonctionnel	334
Tableau 41 – Déclaration d'instance de bloc fonctionnel	338
Tableau 42 – Appel de bloc fonctionnel	340
Tableau 43 – Blocs fonctionnels normalisés bistables ^a	347
Tableau 44 – Blocs fonctionnels normalisés de détection de front	348
Tableau 45 – Blocs fonctionnels normalisés compteur	348
Tableau 46 – Blocs fonctionnels normalisés minuteur	350
Tableau 47 – Déclaration de programme	352
Tableau 48 – Classe	355
Tableau 49 – Déclaration d'instance de classe	357