



IEC 61131-3

Edition 2.0 2003-01

INTERNATIONAL STANDARD

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE **XK**

ICS 25.040; 35.240.50

ISBN 2-8318-6653-7

CONTENTS

FOREWORD	8
1 General	9
1.1 Scope	9
1.2 Normative references	9
1.3 Definitions	9
1.4 Overview and general requirements	14
1.4.1 Software model	14
1.4.2 Communication model	16
1.4.3 Programming model	18
1.5 Compliance	19
1.5.1 System compliance	20
1.5.2 Program compliance	22
2 Common elements	23
2.1 Use of printed characters	23
2.1.1 Character set	23
2.1.2 Identifiers	23
2.1.3 Keywords	24
2.1.4 Use of white space	24
2.1.5 Comments	24
2.1.6 Pragmas	25
2.2 External representation of data	25
2.2.1 Numeric literals	25
2.2.2 Character string literals	26
2.2.3 Time literals	28
2.2.3.1 Duration	28
2.2.3.2 Time of day and date	29
2.3 Data types	29
2.3.1 Elementary data types	30
2.3.2 Generic data types	31
2.3.3 Derived data types	32
2.3.3.1 Declaration	32
2.3.3.2 Initialization	33
2.3.3.3 Usage	35
2.4 Variables	36
2.4.1 Representation	36
2.4.1.1 Single-element variables	36
2.4.1.2 Multi-element variables	38
2.4.2 Initialization	38
2.4.3 Declaration	39
2.4.3.1 Type assignment	41
2.4.3.2 Initial value assignment	42
2.5 Program organization units	45
2.5.1 Functions	45
2.5.1.1 Representation	46
2.5.1.2 Execution control	49
2.5.1.3 Declaration	50
2.5.1.4 Typing, overloading, and type conversion	52
2.5.1.5 Standard functions	55
2.5.1.5.1 Type conversion functions	55
2.5.1.5.2 Numerical functions	56
2.5.1.5.3 Bit string functions	59
2.5.1.5.4 Selection and comparison functions	59
2.5.1.5.5 Character string functions	62
2.5.1.5.6 Functions of time data types	64
2.5.1.5.7 Functions of enumerated data types	66

2.5.2	Function blocks	66
2.5.2.1	Representation	67
2.5.2.1a)	Use of EN and ENO in function blocks	68
2.5.2.2	Declaration	69
2.5.2.3	Standard function blocks	77
2.5.2.3.1	Bistable elements	77
2.5.2.3.2	Edge detection	78
2.5.2.3.3	Counters	78
2.5.2.3.4	Timers	81
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	84
2.6.3	Transitions	86
2.6.4	Actions	89
2.6.4.1	Declaration	89
2.6.4.2	Association with steps	91
2.6.4.3	Action blocks	92
2.6.4.4	Action qualifiers	93
2.6.4.5	Action control	94
2.6.5	Rules of evolution	99
2.6.6	Compatibility of SFC elements	107
2.6.7	SFC Compliance requirements	108
2.7	Configuration elements	108
2.7.1	Configurations, resources, and access paths	110
2.7.2	Tasks	114
3	Textual languages	123
3.1	Common elements	123
3.2	Instruction list (IL)	123
3.2.1	Instructions	123
3.2.2	Operators, modifiers and operands	124
3.2.3	Functions and function blocks	126
3.3	Structured Text (ST)	129
3.3.1	Expressions	129
3.3.2	Statements	131
3.3.2.1	Assignment statements	133
3.3.2.2	Function and function block control statements	133
3.3.2.3	Selection statements	133
3.3.2.4	Iteration statements	134
4	Graphic languages	135
4.1	Common elements	135
4.1.1	Representation of lines and blocks	135
4.1.2	Direction of flow in networks	135
4.1.3	Evaluation of networks	136
4.1.4	Execution control elements	138
4.2	Ladder diagram (LD)	139
4.2.1	Power rails	139
4.2.2	Link elements and states	139
4.2.3	Contacts	140
4.2.4	Coils	140
4.2.5	Functions and function blocks	140
4.2.6	Order of network evaluation	141
4.3	Function Block Diagram (FBD)	143
4.3.1	General	143
4.3.2	Combination of elements	143
4.3.3	Order of network evaluation	143

ANNEX A (normative) Specification method for textual languages	144
A.1 Syntax	144
A.1.1 Terminal symbols	144
A.1.2 Non-terminal symbols	144
A.1.3 Production rules	145
A.2 Semantics	145
ANNEX B (normative) Formal specifications of language elements	146
B.0 Programming model	146
B.1 Common elements	146
B.1.1 Letters, digits and identifiers	146
B.1.2 Constants	147
B.1.2.1 Numeric literals	147
B.1.2.2 Character strings	147
B.1.2.3 Time literals	148
B.1.2.3.1 Duration	148
B.1.2.3.2 Time of day and date	148
B.1.3 Data types	149
B.1.3.1 Elementary data types	149
B.1.3.2 Generic data types	149
B.1.3.3 Derived data types	149
B.1.4 Variables	151
B.1.4.1 Directly represented variables	151
B.1.4.2 Multi-element variables	151
B.1.4.3 Declaration and initialization	152
B.1.5 Program organization units	154
B.1.5.1 Functions	154
B.1.5.2 Function blocks	155
B.1.5.3 Programs	156
B.1.6 Sequential function chart elements	156
B.1.7 Configuration elements	157
B.2 Language IL (Instruction List)	159
B.2.1 Instructions and operands	159
B.2.2 Operators	159
B.3 Language ST (Structured Text)	160
B.3.1 Expressions	160
B.3.2 Statements	160
B.3.2.1 Assignment statements	161
B.3.2.2 Subprogram control statements	161
B.3.2.3 Selection statements	161
B.3.2.4 Iteration statements	161
ANNEX C (normative) Delimiters and keywords	162
ANNEX D (normative) Implementation-dependent parameters	165
ANNEX E (normative) Error conditions	167
ANNEX F (informative) Examples	169
F.1 Function WEIGH	169
F.2 Function block CMD_MONITOR	170
F.3 Function block FWD_REV_MON	173
F.4 Function block STACK_INT	178
F.5 Function block MIX_2_BRIX	183
F.6 Analog signal processing	186
F.6.1 Function block LAG1	187
F.6.2 Function block DELAY	187

F.6.3	Function block AVERAGE	188
F.6.4	Function block INTEGRAL.....	188
F.6.5	Function block DERIVATIVE	189
F.6.6	Function block HYSTERESIS	189
F.6.7	Function block LIMITS_ALARM.....	190
F.6.8	Structure ANALOG_LIMITS.....	190
F.6.9	Function block ANALOG_MONITOR.....	191
F.6.10	Function block PID.....	192
F.6.11	Function block DIFFEQ	193
F.6.12	Function block RAMP	194
F.6.13	Function block TRANSFER.....	195
F.7	Program GRAVEL.....	195
F.8	Program AGV	203
F.9	Use of enumerated data types	206
F.10	Function block RTC (Real Time Clock).....	206
F.11	Function block ALRM_INT.....	206
ANNEX G (informative) Reference character set.....		208
Index		210
Table 1	- Character set features.....	23
Table 2	- Identifier features.....	24
Table 3	- Comment feature.....	25
Table 3a	- Pragma feature.....	25
Table 4	- Numeric literals.....	26
Table 5	- Character string literal features	27
Table 6	- Two-character combinations in character strings	28
Table 7	- Duration literal features	29
Table 8	- Date and time of day literals.....	29
Table 9	- Examples of date and time of day literals	29
Table 10	- Elementary data types	30
Table 11	- Hierarchy of generic data types	32
Table 12	- Data type declaration features	33
Table 13	- Default initial values of elementary data types.....	34
Table 14	- Data type initial value declaration features	35
Table 15	- Location and size prefix features for directly represented variables.....	37
Table 16a	- Variable declaration keywords	39
Table 16b	- Usages of VAR_GLOBAL, VAR_EXTERNAL and CONSTANT declarations	41
Table 17	- Variable type assignment features.....	41
Table 18	- Variable initial value assignment features.....	43
Table 19	- Graphical negation of Boolean signals	47
Table 19a	- Textual invocation of functions for formal and non-formal argument list	49
Table 20	- Use of EN input and ENO output	50
Table 20a	- Function features.....	51
Table 21	- Typed and overloaded functions	53
Table 22	- Type conversion function features	55
Table 23	- Standard functions of one numeric variable.....	57
Table 24	- Standard arithmetic functions.....	58

Table 25 - Standard bit shift functions	59
Table 26 - Standard bitwise Boolean functions	60
Table 27 - Standard selection functions ^d	61
Table 28 - Standard comparison functions	62
Table 29 - Standard character string functions	63
Table 30 - Functions of time data types	64
Table 31 - Functions of enumerated data types	66
Table 32 - Examples of function block I/O variable usage	68
Table 33 - Function block declaration and usage features	71
Table 34 - Standard bistable function blocks ^a	77
Table 35 - Standard edge detection function blocks	78
Table 36 - Standard counter function blocks	79
Table 37 - Standard timer function blocks	81
Table 38 - Standard timer function blocks - timing diagrams	82
Table 39 - Program declaration features	83
Table 40 - Step features	85
Table 41 - Transitions and transition conditions	87
Table 42 - Declaration of actions ^{a,b}	90
Table 43 - Step/action association	92
Table 44 - Action block features	93
Table 45 - Action qualifiers	94
Table 45a - Action control features	98
Table 46 - Sequence evolution	101
Table 47 - Compatible SFC features	108
Table 48 - SFC minimal compliance requirements	108
Table 49 - Configuration and resource declaration features	112
Table 50 - Task features	116
Table 51a - Examples of instruction fields	124
Table 51b - Parenthesized expression features for IL language	125
Table 52 - Instruction List operators	125
Table 53 - Function Block invocation and Function invocation features for IL language	127
Table 54 - Standard Function Block input operators for IL language	129
Table 55 - Operators of the ST language	131
Table 56 - ST language statements	132
Table 57 - Representation of lines and blocks	136
Table 58 - Graphic execution control elements	138
Table 59 - Power rails	139
Table 60 - Link elements	140
Table 61 - Contacts ^a	141
Table 62 - Coils	142
Table C.1 - Delimiters	162
Table C.2 - Keywords	163
Table D.1 - Implementation-dependent parameters	165
Table E.1 - Error conditions	167

Table G.1 - Character representations	208
Table G.2 - Character encodings	209
Figure 1 - Software model	15
Figure 2 a) - Data flow connection within a program.....	16
Figure 2 b) - Communication via GLOBAL variables.....	16
Figure 2 c) - Communication function blocks	17
Figure 2 d) - Communication via access paths	17
Figure 3 - Combination of programmable controller language elements	19
Figure 4 - Examples of function usage	45
Figure 5 - Use of formal argument names.....	48
Figure 6 - Examples of function declarations and usage	52
Figure 7 - Examples of explicit type conversion with overloaded functions	54
Figure 8 - Examples of explicit type conversion with typed functions	54
Figure 9 - Function block instantiation examples	67
Figure 10 - Examples of function block declarations.....	70
Figure 11 a) - Graphical use of a function block name as an input variable	73
Figure 11 b) - Graphical use of a function block name as an in-out variable	74
Figure 11 c) - Graphical use of a function block name as an external variable	75
Figure 12 - Declaration and usage of in-out variables in function blocks.....	76
Figure 14 - ACTION_CONTROL function block - External interface (Not visible to the user)	95
Figure 15 a) - ACTION_CONTROL function block body with “final scan” logic	96
Figure 15 b) - ACTION_CONTROL function block body without “final scan” logic	97
Figure 16 a) - Action control example - SFC representation.....	98
Figure 16 b) - Action control example - functional equivalent	99
Figure 17 - Examples of SFC evolution rules.....	105
Figure 18 a) - Examples of SFC errors: an “unsafe” SFC	106
Figure 18 b) - Examples of SFC errors: an “unreachable” SFC.....	107
Figure 19 a) - Graphical example of a configuration	109
Figure 19 b) - Skeleton function block and program declarations for configuration example	110
Figure 20 - Examples of CONFIGURATION and RESOURCE declaration features.....	113
Figure 21 a) - Synchronization of function blocks with explicit task associations	120
Figure 21 b) - Synchronization of function blocks with implicit task associations	121
Figure 21 c) - Explicit task associations equivalent to figure 21 b)	122
Figure 22 - EXIT statement example	134
Figure 23 - Feedback path example.....	137
Figure 24 - Boolean OR examples	143

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.

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 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.

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, *GRAFNET 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*