IEC 61131-3

Edition 2.0 2003-01

INTERNATIONAL STANDARD

Programmable controllers – Part 3: Programming languages

IEC 61131-3:2003(E)

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.

- Catalogue of IEC publications: www.iec.ch/searchpub
  The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.
- IEC Just Published: www.iec.ch/online_news/justpub
  Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.
- Electropedia: www.electropedia.org
  The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.
- Customer Service Centre: www.iec.ch/webstore/custserv
  If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:
  Email: csc@iec.ch
  Tel.: +41 22 919 02 11
  Fax: +41 22 919 03 00

IEC 61131-3

Edition 2.0 2003-01

INTERNATIONAL STANDARD

Programmable controllers –
Part 2: Equipment requirements and tests

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX XH

ICS 25.040; 35.240.50

ISBN 2-8318-6653-7

CONTENTS

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, *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*.

1.3.40 instance: individual, named copy of the data structure associated with a *function block type* or *program type*, which persists from one *invocation* of the associated operations to the next.

1.3.41 instance name: *identifier* associated with a specific *instance*.

1.3.42 instantiation: the creation of an *instance*.

1.3.43 integer literal: *literal* which directly represents a value of type `SINT`, `INT`, `DINT`, `LINT`, `BOOL`, `BYTE`, `WORD`, `DWORD`, or `LWORD`, as defined in 2.3.1.

1.3.44 invocation: the process of initiating the execution of the operations specified in a *program organization unit*.

1.3.45 keyword: lexical unit that characterizes a *language element*, for example, "`IF`".

1.3.46 label: language construction naming an instruction, network, or group of networks, and including an *identifier*.

1.3.47 language element: any item identified by a symbol on the left-hand side of a production rule in the formal specification given in annex B of this standard.

1.3.48 literal: lexical unit that directly represents a value. (ISO)

1.3.49 local scope: the *scope* of a *declaration* or *label* applying only to the *program organization unit* in which the declaration or label appears.

1.3.50 logical location: the location of a *hierarchically addressed* variable in a schema which may or may not bear any relation to the physical structure of the programmable controller's inputs, outputs, and memory.

1.3.51 long real: real number represented in a *long word*.

1.3.52 long word: 64-bit data element.

1.3.53 memory (user data storage): functional unit to which the user program can store data and from which it can retrieve the stored data.

1.3.54 named element: element of a *structure* which is named by its associated *identifier*.

1.3.55 network: arrangement of nodes and interconnecting branches.

1.3.56 off-delay (on-delay) timer function block: *function block* which delays the *falling (rising) edge* of a Boolean *input* by a specified duration.

1.3.57 operand: *language element* on which an operation is performed.

1.3.58 operator: symbol that represents the action to be performed in an operation.

1.3.59 output variable (output): *variable* which is used to return the result(s) of the *evaluation* of a *program organization unit*.

1.3.60 overloaded: with respect to an operation or *function*, capable of operating on data of different types, as specified in 2.5.1.4.

1.3.61 power flow: the symbolic flow of electrical power in a ladder diagram, used to denote the progression of a logic solving algorithm.