Standards Compliance according to IEC 61131-3

1. Introduction:

The IEC 61131 standard is applicable for the programmable logic controllers (PLC). In accordance with the rules of the European Union, this international standard has been accepted in Germany as DIN EN 6 1131, in France as NF EN 6 1131, and in England as BS EN 6 1131. The most important parts of the standard are quoted below. Quotes are in italics.

Part 3 of this standard defines the application area in Section 1.1:
"This Part of IEC 61131 specifies syntax and semantics of programming languages for programmable controller 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."

Section 1.4 explains the overview and general requirements.
"This Part of IEC 61131 specifies the syntax and semantics of a unified suite of programming languages for programmable controllers (PCs). These consist of two textual languages, IL (Instruction List) and ST (Structured Text), and two graphical languages, LD (Ladder Diagram) and FBD (Function Block Diagram).
Sequential Function Chart (SFC) elements are 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. The programming language elements defined in this part may be used in an interactive programming environment. The specification of such environments is beyond the scope of this standard; however, such an environment shall be capable of producing textual or graphic program documentation in the formats specified in this standard."

Section 1.5 of the standards fulfillment specifies:
"A programmable controller system, as defined in IEC 61131-1, which claims to comply, wholly or partially, with the requirements of this Part of IEC 61131 shall do so only as described below. A compliance statement shall be included in the documentation accompanying the system, or shall be produced by the system itself. The form of the compliance statement shall be: "This system complies with the requirements of IEC 61131-3, for the following language features:"; followed by a set of compliance tables …".

Die Tabellennummern müssen hierbei denen der zugehörigen Normeigenschaften entsprechen.

2. Standards Compliance in STEP 7

The SIMATIC STEP 7 system complies with the requirements of IEC 61131-3 for the following programming languages

- Ladder Logic KOP/LAD (corresponds to the IEC 61131-3 language "LAD/LD")
- Function Block Diagram FUP/FBD (corresponds to the IEC 61131-3 language "FUP/FBD")

in the characteristics described in the following Chapter 4 and in the appendices.
3. Substitutes and Additional Language Elements

In addition, the standard stipulates that a standardized PLC system

a) may not include any substitute or additional language elements to attain a standardized characteristic.
b) has specified all implementation-dependent parameters according to Annex D.
c) reports user errors from Annex E; (for a partial program check, reference must be made to incompleteness).
d) reports user errors during converting and/or during start-up, and specifies or introduces appropriate measures.
e) All characteristics not permissible or not present in the standard must be described as "expansions." in a document
f) treats these expansions in the same way as is specified for errors (as a test that can be used as an option).
g) All implementation-independent characteristics from Appendix D must be handled as is specified for errors (as a test that can be used as an option).
h) No standardized names with meanings that vary can be used for manufacturer-defined characteristics.
i) The formal syntax of the text languages is described in a document according to Annex A.
j) shall be capable of reading and writing files containing any of the language elements defined as alternatives in the production library_element_declaration in B.0, in the syntax defined in requirement (i) above, encoded according to the "ISO-646 IRV" given as Table 1 - Row 00 of ISO/IEC 10646.

The STEP 7 programming software fulfills the requirements of the standard in points b), c), d), e), h), i). In respect to a), there exist language elements for compatibility reasons with STEP 5, which might be taken for additional elements.
The f) and g) requirements are not used for STEP 7.

4. Elements Realized According to the Standard

The standard defines all standardized language elements in tables, the rows of which reference the realized feature by number. The language elements which are realized in STEP 7 according to the standard are specified below.

A good knowledge of the norm mentioned is a prerequisite for understanding the following tables.
The German version of DIN IEC 61131-3 : 1994-08 can be obtained from
Beuth Verlag GmbH, 10772 Berlin, Germany. Fax (.30) 2601-1231.
The International Standard IEC 61131-3 : First edition 1993-03 (English/French) can be obtained from
Central Office of IEC, 3 rue de Varembe, Geneve, Switzerland.
### 4.1 Common Elements

<table>
<thead>
<tr>
<th>Table</th>
<th>No.</th>
<th>Language Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Character set features</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5b</td>
</tr>
<tr>
<td>2</td>
<td>Identifier features</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Comment features</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: STL only line comments starting with // and ending with new line.</td>
</tr>
<tr>
<td>4</td>
<td>Numeric literals</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note *): bit length required: W#16#ADAC, DW#16#ADAC_4711</td>
</tr>
<tr>
<td>5</td>
<td>Character string literals</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: according Datentyp char#</td>
</tr>
<tr>
<td>6</td>
<td>Two-character combinations in character strings</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>Duration literal features</td>
<td>1a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2b</td>
</tr>
</tbody>
</table>
Date and time of day literals
1 Date literals (long prefix)
2 Date literals (short prefix)
3 Time of day literals (long prefix)
4 Time of day literals (short prefix)
5 Date and time literals (long prefix)
6 Date and time literals (short prefix)

Keywords
1 BOOL
3 INT
4 DINT
10 REAL
12 TIME
13 DATE
14 TIME_OF_DAY or TOD
15 DATE_AND_TIME or DT
16 STRING *)
17 BYTE
18 WORD
19 DWORD

*): STRING n with length n; otherwise 254 bytes.

Memory Location and size prefix features for directly represented variables
1 I or E depending in the language setting
2 Q or A depending in the language setting
3 M
4 X *)
5 none
6 B
7 W
8 D
*): for DB only.

Keywords for variable declarations
VAR, VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR_TEMP according to IEC

Assignment of types to variables
5 Automatic memory allocation of symbolic variables
* see note in table 10
6 Array declaration
7 Declaration of retentive array declaration

Assignment of initial values for variables
5 Initialization of symbolic variables *) see note in table 10
Graphical negation of Boolean signals
1 negated Input (only FUP)
2 negated Output (only FUP)

Textual invocation of functions for formal and non-formal argument list
1 formal
2 non formal (only with a single argument)

Use of EN input and ENO output
1 Use of "EN" and "ENO " with LAD/FBD if needed for graphical signalflow
2 Use without "EN" and "ENO " with LAD/FBD if not needed for graphical signalflow

Function features
1 IN_OUT variable declaration (textual)

Typed and overloaded functions
2 Typed functions

Note: Only 2 parameters with FDB/LAD; EN + ENO additional.

Type conversion function features
1 *_TO_**
2 TRUNC
3 BCD_TO_** (not for SCL)
4 *_TO_BCD (not for SCL)

Standard functions of one numeric variable
1 ABS
2 SQRT
3 LN *)
5 EXP
6 SIN *)
7 COS *)
8 TAN *)
9 ASIN *)
10 ACOS *)
11 ATAN *)

*): The implementation of these functions is CPU specific.

Standard arithmetic functions
12 ADD +
13 MUL *
14 SUB -
15 DIV /
16 MOD
18 MOVE :=

Note: All functions with FBD/LAD are typed (e.g. integer).

Standard bit shift functions
1 SHL
2 SHR
3 ROR
4 ROL

Note: All functions are typed (e.g. word).
26 **Standard bitwise Boolean functions**
5 AND
6 OR
7 XOR
8 NOT

27 **Standard selection functions**
1 SEL
2a MAX
2b MIN
3 LIMIT

28 **Standard compare functions**
5 GT >
6 GE >=
7 EQ =
8 LE <=
9 LT <
10 NE <>

Note: All functions are typed with FBD/LAD (e.g. Word).

29 **STRING data type functions**
1 LEN
2 LEFT
3 RIGHT
4 MID
5 CONCAT
6 INSERT
7 DELETE
8 REPLACE
9 FIND

Note: the implementation of these functions are CPU-specific.

30 **TIME data type functions**
1b ADD_Time )
3b ADD_DT_T
4b SUB_Time
8b SUB_DT_T
9b SUB_DT_DT

Note: the implementation of these functions is CPU-specific.
Function block declaration features
1a Retain internal variables
1b Non-Retain internal variables
2a Retain output variables
2b Retain input variables
2c Non-Retain output variables
2d Non-Retain input variables
4a Input/output declaration (textual)
5b Funktionsblock-instancename as input (grafical)
11 VAR_TEMP Deklaration

Standard bistable function blocks
1 SR
2 RS

Note: SR_FF is reset dominant; RS_FF is set dominant.

Standard edge detection function blocks
1 R_TRIG Rising edge detector (P_TRIG)
2 F_TRIG Falling edge detector (N_TRIG)

Standard Counter function blocks
1a CTU (Up-counter)
1b CTU_DINT (Up-counter)
1d CTU_UDINT (Up-counter)
2a CTD (Down-counter)
2b CTD_DINT (Down-counter)
2d CTD_UDINT (Down-counter)
3a CTUD (Up/Down-counter)
3b CTUD_DINT (Up/Down-counter)
3d CTUD_UDINT (Up/Down-counter)

Standard timer function blocks
1 TP (Pulse)
2a TON (On-delay)
3a TOF (Off-delay)

Tasks

STEP7 offers tasks as Organisationblocks (OB)
4.6 Common Graphical Elements

Table  No. Language Elements

57 Representation of lines and blocks
Horizontal lines:
1 ISO/IEC 646 "minus" character
2 Graphic or semigraphic
Vertical lines:
3 ISO/IEC 646 "vertical line" character
4 Graphic or semigraphic
Horizontal/vertical connection:
5 ISO/IEC 646 "plus" character
6 Graphic or semigraphic
Blocks with connecting lines:
11 ISO/IEC 646 characters
12 Graphic or semigraphic

58 Graphic execution control elements

2 Unconditional jump LAD
3 Conditional jump FBD
4 Conditional jump LAD
5 Conditional return LAD
6 Conditional jump FBD
7 Unconditional return

Note: in LAD represented as coils.

4.7 Ladder Diagram Language (LD)

Table  No. Language Elements

59 Power rail symbols
1 Left power rail
2 Right power rail

60 Link element symbols
1 Horizontal link
2 Vertical link

61 Contacts
1 Normally open contact
3 Normally closed contact
5 Contact for positive edge detection
7 Contact for negative edge detection

62 Coils
1 Coil
2 Negated coil
3 SET (latch) coil
4 RESET (unlatch) coil
8 Coil for positive edge detection
9 Coil for negative edge detection

ANNEX A - Syntax
only needed for textual languages
ANNEX D - Implementation-dependent Parameters

Below, you will find the parameters for the language elements defined in the standard and the limits realized in STEP 7.

<table>
<thead>
<tr>
<th>IEC-Reference</th>
<th>Parameters</th>
<th>STEP 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 General</td>
<td>Error handling procedures</td>
<td>see Annex E</td>
</tr>
<tr>
<td>2 Common elements</td>
<td>National characters used: # or &quot; pounds Sterling&quot; sign, $ or &quot;currency&quot; sign,</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maximum length of identifiers</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>Maximum comment length</td>
<td>Network comment max. 64 kB, Operand comment &gt; 2000 Zeichen</td>
</tr>
<tr>
<td></td>
<td>Range of values of duration</td>
<td>- 24D_20H_31m_23.648s to + 24D_20H_31m_23.647s</td>
</tr>
<tr>
<td>2.3 Data types</td>
<td>Range of values for variables of TIME type</td>
<td>- 24D_20H_31m_23.648s to + 24D_20H_31m_23.647s</td>
</tr>
<tr>
<td></td>
<td>Precision of representation of seconds in TIME_OF_DAY and DATE_AND_TIME</td>
<td>milliseconds</td>
</tr>
<tr>
<td></td>
<td>Maximum number of array subscripts</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Maximum array size</td>
<td>Max. 65536 elements dependant on existing memory and data type</td>
</tr>
<tr>
<td></td>
<td>Maximum number of structure elements</td>
<td>not available</td>
</tr>
<tr>
<td></td>
<td>Maximum structure size</td>
<td>Ca. 2000 (estimated)</td>
</tr>
<tr>
<td></td>
<td>Maximum number of variables per declaration</td>
<td>not available</td>
</tr>
<tr>
<td></td>
<td>Default maximum length of STRING-variables</td>
<td>254</td>
</tr>
<tr>
<td></td>
<td>Maximum allowed length of STRING-variables</td>
<td>254</td>
</tr>
<tr>
<td>2.4 Variables</td>
<td>Maximum number of hierarchical levels</td>
<td>2 physical mapping</td>
</tr>
<tr>
<td></td>
<td>Logical or physical mapping</td>
<td>physical mapping</td>
</tr>
<tr>
<td></td>
<td>Maximum number of subscripts</td>
<td>1 65536 entries</td>
</tr>
<tr>
<td></td>
<td>Maximum range of subscript values</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Maximum number of levels of structures</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Initialization of system inputs</td>
<td>System: 0, User: definable initial values</td>
</tr>
<tr>
<td></td>
<td>Maximum number of variables per declaration</td>
<td>Ca. 2000 (geschätzt)</td>
</tr>
<tr>
<td>2.5 Programorganisationunits</td>
<td>Information to determine execution times of program organization units</td>
<td>not available</td>
</tr>
<tr>
<td></td>
<td>Maximum number of function specifications</td>
<td>dependant on operation and PLC</td>
</tr>
<tr>
<td></td>
<td>Maximum number of inputs of extensible functions</td>
<td>&gt; 32</td>
</tr>
<tr>
<td></td>
<td>Effects of type conversions on accuracy</td>
<td>not available</td>
</tr>
<tr>
<td></td>
<td>Accuracy of functions of one variable</td>
<td>IEEE-Gleitpunkt up to 64 Bit</td>
</tr>
<tr>
<td>IEC-Reference</td>
<td>Parameters</td>
<td>STEP 7</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td>Implementation of arithmetic functions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maximum number of function block specifications and instantiations</td>
<td>dependant on PLC from 128 to 65536</td>
</tr>
<tr>
<td></td>
<td>PVmin, PVmax of counters</td>
<td>dependant on counter datatype (up to UDINT)</td>
</tr>
<tr>
<td></td>
<td>Program size limitations</td>
<td>dependant on the PLC memory and used operations</td>
</tr>
<tr>
<td>2.7</td>
<td>Contents of RESOURCE libraries</td>
<td>integrated functions /FB und FC</td>
</tr>
<tr>
<td>Configuration</td>
<td>Maximum number of tasks</td>
<td>not available</td>
</tr>
<tr>
<td>elements</td>
<td>Task- interval resolution</td>
<td></td>
</tr>
</tbody>
</table>
### ANNEX E - Error Conditions

Below, you will find the error conditions named in the standard and their occurrence in STEP 7.

<table>
<thead>
<tr>
<th>Error conditions</th>
<th>LAD/FBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value of a variable exceeds the specified subrange</td>
<td>at run time</td>
</tr>
<tr>
<td>Length of initialization list does not match number of array entries</td>
<td>at compilation time</td>
</tr>
<tr>
<td>Improper use of directly represented or external variables in functions</td>
<td>at compilation time</td>
</tr>
<tr>
<td>Type conversion errors</td>
<td></td>
</tr>
<tr>
<td>Numerical result exceeds range for data type</td>
<td>at run time</td>
</tr>
<tr>
<td>Division by zero</td>
<td>system flags</td>
</tr>
<tr>
<td>Mixed input data types to a selection function</td>
<td>at compilation time</td>
</tr>
<tr>
<td>Selector (K) out of range for MUX function</td>
<td></td>
</tr>
<tr>
<td>Invalid character position specified</td>
<td>at run time</td>
</tr>
<tr>
<td>Result exceeds maximum string length</td>
<td></td>
</tr>
<tr>
<td>Result exceeds range for data type</td>
<td>request of a system flag</td>
</tr>
<tr>
<td>Data type conflict in VAR_ACCESS</td>
<td>not available</td>
</tr>
<tr>
<td>Tasks require too many processor resources</td>
<td>not available</td>
</tr>
<tr>
<td>Execution deadline not met Other task scheduling conflicts</td>
<td></td>
</tr>
<tr>
<td>Numerical result exceeds range for data type</td>
<td>request of a system flag</td>
</tr>
<tr>
<td>Division by zero</td>
<td>request of a system flag</td>
</tr>
<tr>
<td>Invalid data type for operation</td>
<td>at compilation time</td>
</tr>
<tr>
<td>Return from function without value assigned</td>
<td>at run time</td>
</tr>
<tr>
<td>Iteration fails to terminate</td>
<td>not possible, compiler error message</td>
</tr>
<tr>
<td>Same identifier used as connector label and element name</td>
<td></td>
</tr>
<tr>
<td>Un-initialized feedback variable</td>
<td>not available r</td>
</tr>
</tbody>
</table>