Actual PLC Programming Standard and Worldwide Acceptance

IEC 61131-3 and PLCopen Activity

Ralf Wohlschlaeger

General Manager Factory Automation
Matsushita Electric Works (Europe) AG

Chairman of PLCopen PC1 committee (promotion)
Contents :

- What is IEC 61131-3 ?
  - History
  - Advantages
  - Explanation
  - Applications

- World-wide popularity ?
  - Europe/USA/Asia

- What is PLCopen ?
  - Organisation
  - Current topics
  - Conformity Level, Reusability Level, Motion Control FBs, XML etc.

- Future prospect
The Way to IEC61131-3 Programming

NEMA Programmable Controllers Committee formed (USA)
GRAFCET (France)
DIN 40719, Function Charts (Germany)
NEMA ICS-3-304, Programmable Controllers (USA)
IEC SC65A/WG6 formed
DIN 19 239, Programmable Controller (Germany)
IEC 65A(Sec)38, Programmable Controllers
MIL-STD-1815 Ada (USA)
IEC SC65A(Sec)49, PC Languages
IEC SC65A(Sec)67
IEC 848, Function Charts
IEC 64A(Sec)90
IEC 1131-3
Type 3 report recommendation
IEC 61131-3 name change

Source: Dr. J. Christensen
### Conventional styled software

- Direct hardware address:
  - X0, X1, Y0, Y1, DT0, DT1, ...
- Each variable has a name
- Each variable has a data type
- Global and local variables
- 1 Program from start to end
- No structure

### IEC 61131-3 styled software

- IEC address:
  - %IX0, %QX0, %MW5.0
  - Each variable has a name
  - Each variable has a data type
  - Global and local variables
- POU concept:
  - 1 program or more programs
  - Function Blocks and Functions
- Well structured
Why IEC 61131-3?

IEC 61131-3
An internationally accepted standard

• Unified rules in systems worldwide, reduces misunderstandings and shortens training

• Reuse of ready-made Functions and Function Blocks, saves time for programming and debugging

• Better overview through structure and modularity

• Fewer errors through defined data types and encapsulation

• Safe investment due to standardisation
Examples of IEC 61131-3 
advantages

Variables:
- better documentation --> programming by names / symbols
- I/O connection list already stored in the project
- Base for the re-use of software

POUs:
- structured programming
- well defined interface --> other variables can be used in other projects
- re-use of Function Blocks saves time and debugging

SFC:
- flowchart on the monitor
- divide big programs into small and easy parts
- top down development / bottom up --> well structured
- different languages in the program
- easy debugging and error locating - only the current step is active

General:
- save training time for programmers
- enables parallel software development by more programmers
- certification ensures users to protect their investments for the future
Use Variable Names instead of Addresses
Flexible IEC instructions: 1 function instead of several

Conventional programming requires different functions for e.g.:

16-bit

32-bit

4-digit
BCD data

8-digit
BCD data

Floating point data

Input data must be of the same data type!
Part 1  General overview, definitions
Part 2  Hardware
  ● I/O signals, safety requirements, environment
Part 3  Programming Languages
Part 4  User Guidelines
Part 5  Communication
Part 6  Reserved
Part 7  Fuzzy control
Part 8  Guidelines for the application and implementation

International Standard
Configuration A

Resource L

Task1
Program P1

Task2
Program P2

FB1
FB2

Task3
Program P3

Task4
Program P4

FB3
FB4

global and direct addressed variables

access paths

Task association
Access path association
### IEC 61131-3: The 5 Programming Languages and The Common Elements

- **Character set**: (English...........)
- **Data types**: (BOOL, WORD, INTEGER.................)
- **Variables**: (VAR, VAR_input, VAR_output............)
- **POUs, Program Organisation Units**: (Function, Function Block...)  
- **SFC Elements**: (Steps, Transitions........................)
- **Configuration elements**: (Tasks)

- **Basis for software re-use**
# PLCopen

Creating the foundation for your efficiency

## IEC 61131-3 Elementary Data Types

<table>
<thead>
<tr>
<th>No.</th>
<th>Keyword</th>
<th>Data Type</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BOOL</td>
<td>Boolean</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>SINT</td>
<td>Short integer</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>INT</td>
<td>Integer</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>DINT</td>
<td>Double integer</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>LINT</td>
<td>Long integer</td>
<td>64</td>
</tr>
<tr>
<td>6</td>
<td>USINT</td>
<td>Unsigned short integer</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>UINT</td>
<td>Unsigned integer</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>UDINT</td>
<td>Unsigned double integer</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>ULINT</td>
<td>Unsigned long integer</td>
<td>64</td>
</tr>
<tr>
<td>10</td>
<td>REAL</td>
<td>Real numbers</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td>LREAL</td>
<td>Long reals</td>
<td>64</td>
</tr>
<tr>
<td>12</td>
<td>TIME</td>
<td>Duration</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DATE</td>
<td>Date (only)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>TIME_OF_DAY</td>
<td>Time of day (only)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>DATE_AND_TIME</td>
<td>Date and time of day</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>STRING</td>
<td>Character string</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>BYTE</td>
<td>Bit string of length 8</td>
<td>8</td>
</tr>
<tr>
<td>18</td>
<td>WORD</td>
<td>Bit string of length 16</td>
<td>16</td>
</tr>
<tr>
<td>19</td>
<td>DWORD</td>
<td>Bit string of length 32</td>
<td>32</td>
</tr>
<tr>
<td>20</td>
<td>LWORD</td>
<td>Bit string of length 64</td>
<td>64</td>
</tr>
</tbody>
</table>
The 5 Languages of IEC 61131-3

Instruction List

- LD A
- ANDN B
- ST C

Structured Text

C := A AND NOT B

Sequential Function Chart

Step 1
- N FILL

Transition 1

Step 2
- S Empty

Transition 2

Step 3

Ladder Diagram

A B C

- / /-|/|-----------------( )

Function Block Diagram

AND

A B C
POU = Program Organization Unit

• A POU consists of a header (variable declaration) and the body (instructions)

• POUs enable the re-use of software from macro level (Programs) to micro level (FB and Functions)

<table>
<thead>
<tr>
<th>POU Type</th>
<th>Replicated as:</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>Program instance</td>
<td>Main program</td>
</tr>
<tr>
<td>Function Block</td>
<td>FB instance</td>
<td>Subroutine with own memory, several in - and outputs possible</td>
</tr>
<tr>
<td>Function</td>
<td>Function</td>
<td>Subroutine without own memory</td>
</tr>
</tbody>
</table>
Function Blocks can be easily reused

1. Function Block
   - FU_Block
     - OUT_1
     - IN_1
     - IN_2
     - IN_3

2. Variable Interface
   - FU-Block Header
     | Class | Identifier | Type  |
     |-------|------------|-------|
     | 0     | VAR_INPUT | IN_1  | BOOL |
     | 1     | VAR_INPUT | IN_2  | BOOL |
     | 2     | VAR_INPUT | IN_3  | BOOL |
     | 3     | VAR_OUTPUT| OUT_1 | BOOL |

3. Program body
   - Program once
     - LD IN_1
     - AND IN_2
     - OR OUT_1
     - AND IN_3
     - ST OUT_1

Program always

Reuse always
Easy Programming of FBs and FUN
**Easy Programming of FBs and FUN**

**Define inputs and outputs**

<table>
<thead>
<tr>
<th>Class</th>
<th>Identifier</th>
<th>Type</th>
<th>Initial</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VAR_INPUT</td>
<td>In1</td>
<td>BOOL</td>
<td>FALSE</td>
</tr>
<tr>
<td>1</td>
<td>VAR_INPUT</td>
<td>In2</td>
<td>BOOL</td>
<td>FALSE</td>
</tr>
<tr>
<td>2</td>
<td>VAR_INPUT</td>
<td>In3</td>
<td>BOOL</td>
<td>FALSE</td>
</tr>
<tr>
<td>3</td>
<td>VAR_OUTPUT</td>
<td>Out1</td>
<td>BOOL</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

**Program FB contents**

```plaintext
1

Sensor_start
Start_button
Limit_Switch_1

FIRST_SWITCH
In1
In2
In3

Motor_up_down

Out1
```

www.PLCopen.org
## Library Concept

**Libraries:**
- IEC_Standard_Lib
- Vendor_Lib
- Pulsed_Lib
- Communication_Lib
- PID_Lib
- Special_Project_Lib

**Special_Lib**
- TWO_TRIP
- POSITION_2_AX
- POSITION_3_AX
- WAIT_10s
- ELEVAT_4_FL
- ...
- ...

- Self-created FBs can be stored in libraries
- Comfortable structuring and sorting in the libraries
- Know-how protection of FBs and libraries
- Easy reuse of tested software --> saves time
Applications

- Building Automation
- Elevators
- Building Automation
- Light, alarm, window, heating
- Linishing-grinding and polishing machines
- Bakery
- Temperature control
- Telecommunication
Applications

- Water cleaning plant
- Agricultural industry
  - Flower bulbs
- Wrapping Machines
- Medical Equipment
Applications

- Baggage handling in airports
- Food production and packing
- Garage control and traffic remote communication
- Lift cars
- Calibration of taximeters
International Acceptance of IEC 61131-3

PLCopen
Creating the foundation for your efficiency

PLC market share worldwide

Sources: • Frost & Sullivan
• ARC
• ZVEI
Popular Languages Worldwide

1. LD 93%
2. FBD
3. ST

all 5 languages

Outside Europe LD is the most popular PLC programming language
International Acceptance of IEC 61131-3

Europe

- Most PLC suppliers promote IEC 61131-3 in Europe and offer software and training

- Germany, Netherlands, Switzerland, Austria: IEC 61131-3 is well-known and accepted, required by at least 80% of users

- France, Scandinavia: IEC 61131-3 is known by many users, still some require traditionally styled systems

- UK, Italy, Spain: only few small users know IEC 61131-3, most of them require traditionally styled systems
More and more PLC suppliers start to promote IEC 61131-3. Still in the starting phase.

Many users do not use IEC 61131-3 style because it is too difficult for field engineers.

Small increase in IEC 61131-3 step by step

Customers opinions about IEC 61131-3:
- 7%: it is a must
- 41%: somewhat important
- 39%: not important at all
- 13%: no answers

OMAC paper is supporting IEC 61131-3 acceptance in USA
Asia can be divided into 3 parts:

**Japan:**
Very few users are using IEC 61131-3, still difficult to use and to understand. Most Japanese PLC suppliers have examined IEC 61131-3. Some first systems are available, but still no real start/success.

**Australia:**
IEC 61131-3 is well-known. Larger companies are using it already and it is accepted. Small companies still prefer traditional software style.

**Rest of Asia:**
IEC 61131-3 is still not popular. Users prefer easy Ladder Diagram programs. Japan, US and Europe influence the market. China is starting now to promote IEC 61131-3
PLCopen is a World-wide association

Main Office in Europe
Office in North America
Office in Japan
PLCopen was founded on June 15, 1992 in Giessen, Germany. Target was to promote IEC 61131-3, inform customers and give more weight to the IEC 61131-3 standard.
PLCopen *Mission*

We want to be the leading association resolving topics related to control programming to support the use of international standards in this field.
PLCopen is a World-wide association

> 100 members (Nov 2002)

from 21 countries all over the world

Suppliers, institutes and users

See newsletter / website for up-to-date list
Without testing there is no standard

Meanwhile only truly compliant IEC 61131-3 systems are promoted as IEC 61131-3 products

- The IEC 61131 standard gives rules for compliancy
- Certification guides users towards real IEC 61131-3 programming systems (e.g. PLCopen certified products)
TC3: PLCopen Compliance Levels

Application Program

FB

Variable

Code Body

FB

Variable

Code Body

Conformity Level & Reusability Level
Re-usability of Function (Block) libraries

Base Level
Portability of minimal systems
Certification

Certified products can use these logos

Base Level: first step into IEC 61131-3 software

Conformity Level: conforms to IEC 61131-3 based on supported data types

Reusability Level: reuse of IEC 61131-3 Function Blocks based on supported data types

Motion Control: certified Function Blocks according the Motion Control specification
Function Block exchange in ST language is possible:
• if both systems have the Reusability Level ST
• the used instructions are IEC 61131-3 instructions
• the same data types are available
Revolutionizing the industry with a global standard

*Mechanics do not help anymore, a standard with software is possible*

Reduce maintainance and sanitation

*Less hardware parts, more software*

The solution is........

........Software
Advantages of servo driven systems

OMAC / PackAdvantages

- Research done by company RA Jones / Rick Liddington
- In cooperation with Packaging World (magazine)

Traditional Mechanical Design

Over 350 parts packing machine

Targets for a new design:
- ‘Zero’ maintenance
- Sanitation design

Solution
- Aim for a 50% reduction in mechanical parts
Multi Axis Servo Drive

Traditional Mechanical Design

Major part count reduction
- Pulleys - 45 to 0
- Belts - 15 to 0
- Drive sprockets - 15 to 0
- Spline shafts - 2 to 0
- Gearboxes - 16 to 10
- Motors - 1 to 10
- Bearings - 18 to 3
- Line shafts - 6 to 0
- Total - 118 to 23 (81% reduction)

Faster – Better – Cheaper!
Software instead of Hardware
Results:

- Reduction in Size Changeover Time

Side effects:

- Reduction in Size Changeover Time
- Reduce delivery time
Motion Control Standardization means:

- Hardware independent Software Development
- Consistent Development Environment
- Consistent Installation and Maintenance Interface

Same

‘Look and Feel’

IEC 61131-3 is a good base
The PLCopen Task Force Motion Control

- Initiated by Users
- ..to fulfil their requirements

Goal:

To harmonize the access for Motion Control across different platforms during development, installation and maintenance based on the IEC 61131-3 environment

Users:
- Bosch Packaging
- Kuka
- Kloeckner Tevopharm
- Focke EKB
- Hershey Foods
- Tetra Pak

Suppliers:
- Siemens
- Elau
- Beckhoff Industrial Electronics
- SEW Eurodrive
- Mitsubishi Electric Europe
- Cross Hueller
- Lenze
- Parker Hannifin
- ISG Stuttgart
- Control Techniques
- Phoenix Contact
- Keba
- KW Software
- Rockwell Automation
- Nyquist
- Baumueller
- infoteam Software
- Rexroth Indramat
HW Independence via Function Blocks

Software View

Encapsulation / Information Hiding

Hardware View

Name

Inputs

Outputs

I/F

Sercos Drive

Motor

E

I/F

PWM

Drive

Motor

E
Example of a Function Block

MoveAbsolute

<table>
<thead>
<tr>
<th>AXIS_REF</th>
<th>BOOL</th>
<th>REAL</th>
<th>REAL</th>
<th>REAL</th>
<th>REAL</th>
<th>MC_Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis</td>
<td>Execute</td>
<td>Position</td>
<td>Velocity</td>
<td>Acceleration</td>
<td>Deceleration</td>
<td>Direction</td>
</tr>
<tr>
<td>Axis</td>
<td>Done</td>
<td>CommandAborted</td>
<td>Error</td>
<td>Error</td>
<td>ErrorID</td>
<td></td>
</tr>
<tr>
<td>Axis</td>
<td>AXIS_REF</td>
<td>BOOL</td>
<td>BOOL</td>
<td>BOOL</td>
<td>WORD</td>
<td></td>
</tr>
</tbody>
</table>

**FB-Name**: MC_MoveAbsolute

This function block commands a controlled motion at a specified absolute position.
A Motion Control Example...
Drilling of a hole
Solution with Function Block Diagram

**Required functionality**
- Move Absolute
- Move Relative
- Timer
- Move Absolute
Solution with SFC

- **Step 1:** Initialization (power up)
- **Step 2:** Move forward to drilling position and start the drill turning. Check if both actions are completed.
- **Step 3:** Drill the hole.
- **Step 4:** After drilling wait for the step-chain sequence to finish dwelling to free the hole of any debris which might have been stuck in the hole.
- **Step 5:** Move drill back to starting position and shut the spindle off. Combining the completion of moving backwards and stopping the spindle we signal the step-chain to start over.
Conclusion

- PLCopen Motion Control Library provides an independent user interface
- Support for single axis and multiple axes / motion control
- IEC 61131-3 with PLCopen MC provides mechatronic solutions
- User derived FBs and data structures, and multi-tasking are crucial

Future

- Part 1 of the PLCopen Motion Control Library released
- Is implemented now by several suppliers
- Part 2 deals with extensions
- Part 3 will deal with user guidelines
TC6 - XML:

- specification of XML schemes for all IEC 61131-3 languages
- and full projects
- this will provide the basis for exchange (incl. graphical languages)
- and coupling to other software tools
- Start is for FBD languages, as it is the most complicated one
• The basis was provided as a proposal from Schneider Automation and handed over to PLCopen.

• Additional proposals were presented by Rockwell Automation, ABB, and Beckhoff.

• First results (specification) are expected end of 2003
Future direction of PLCopen?
PLCopen and IEC 61499?

Harmonization of new technologies e.g. Ethernet?

Define new FB libraries after Motion Control, which?

Focus on special markets?

Communication proposals? (IEC 61131-5)

Create new advantages for members to increase active support in committees?

Variables / types definition for interfaces to other automation products as SCADA, HMI etc.?

Other issues???

Depend on members activities!!
Thank you very much for your attention!