IEC 61131-3 Basics and PLCopen

European Panasonic Headquarter, Germany

created by
Ralf Wohlschlaeger
General Manager Factory Automation
Panasonic Electric Works (Europe) AG

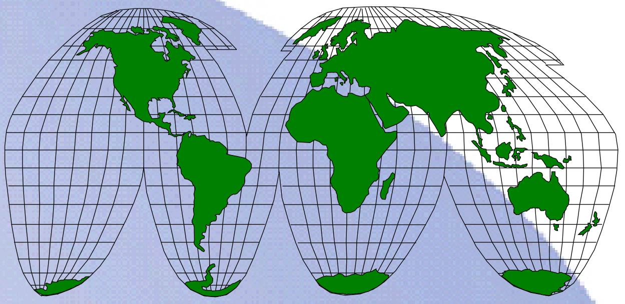
Chairman of PLCopen PC1 committee (promotion)

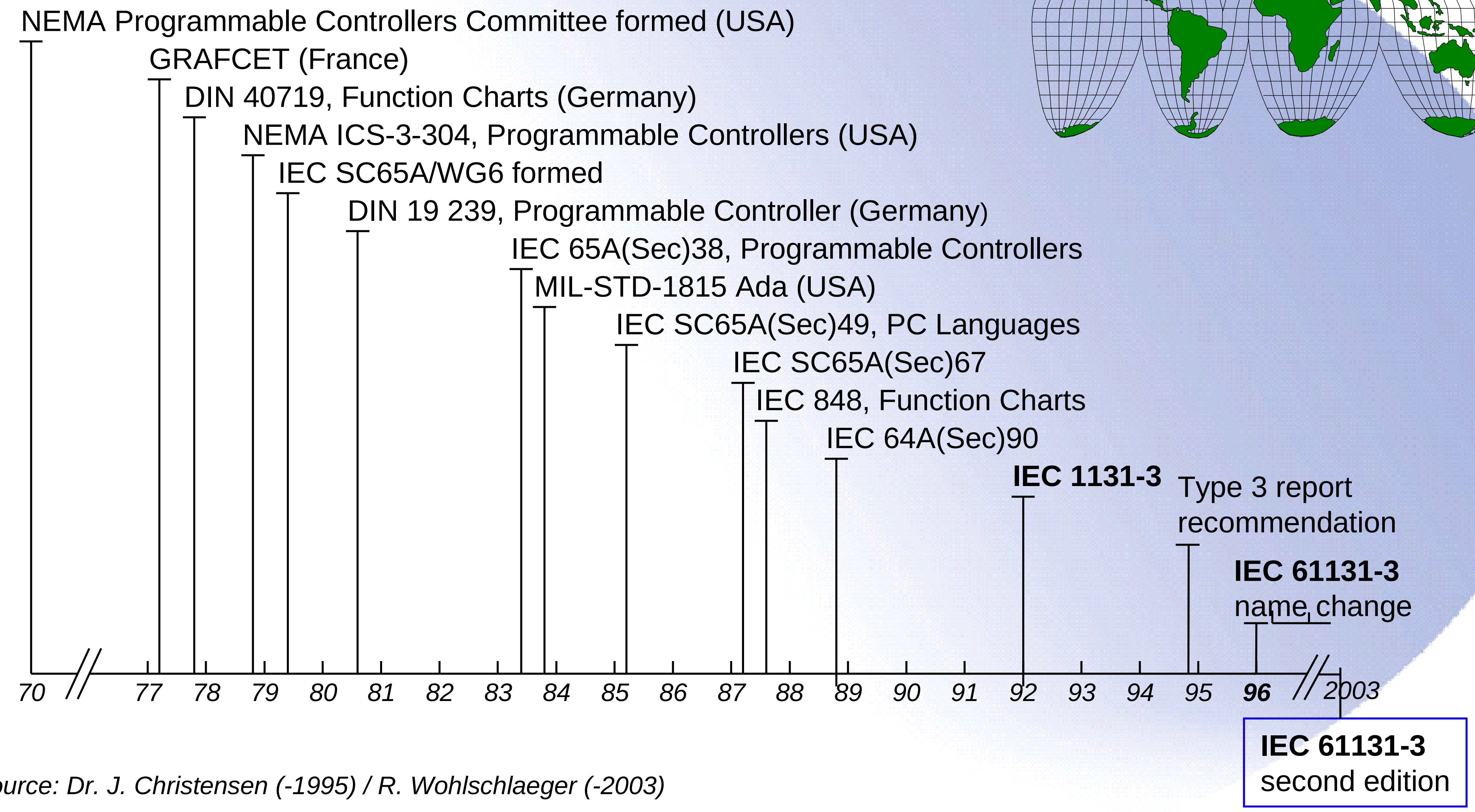
Contents :

- What is IEC 61131-3 ?
  - History
  - Advantages
  - Explanation
- What is PLCopen ?
  - Organisation
  - Current topics

Panasonic
ideas for life

The Way to IEC61131-3 Programming



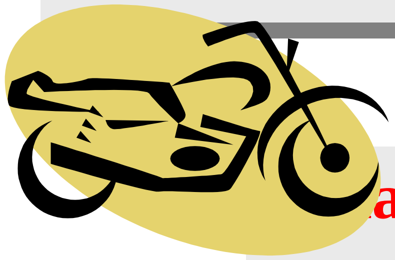


NEMA Programmable Controllers Committee formed (USA)
GRAFCET (France)
DIN 40719, Function Charts (Germany)
NEMA ICS-3-304, Programmable Controllers (USA)
IEC SC65A/WG6 formed
DIN 19 239, Programmable Controller (Germany)
IEC 65A(Sec)38, Programmable Controllers
MIL-STD-1815 Ada (USA)
IEC SC65A(Sec)49, 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
IEC 61131-3 second edition
70
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
2003

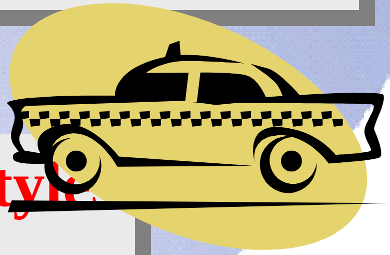
Source: Dr. J. Christensen (-1995) / R. Wohlschlaeger (-2003)

Style of Software

Conventional styled software	IEC 61131-3 styled software
Direct hardware address : - X0, X1...Y0,Y1....DT0,DT1.....	IEC address: - %IX0.....%QX0.....%MW5.0 - each Variable have a name - each Variable have a data type - global and local Variables
1 Program from start to end	POU concept: - 1 program or more programs - Function Blocks and Functions



asonic Control FPWIN Pro can use both sty

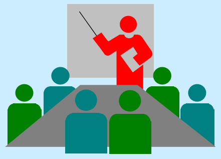


Both styles can be mixed

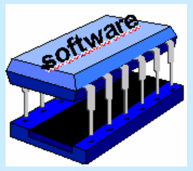
Why IEC 61131-3?

Panasonic
ideas for life

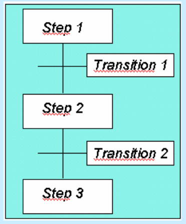
IEC 61131-3
An internationally accepted standard



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

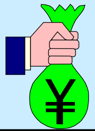


- Reuse of ready-made Functions and Function Blocks,
  saves time for programming and debugging



- Better overview through structure and modularity

- Fewer errors through defined data types and encapsulation



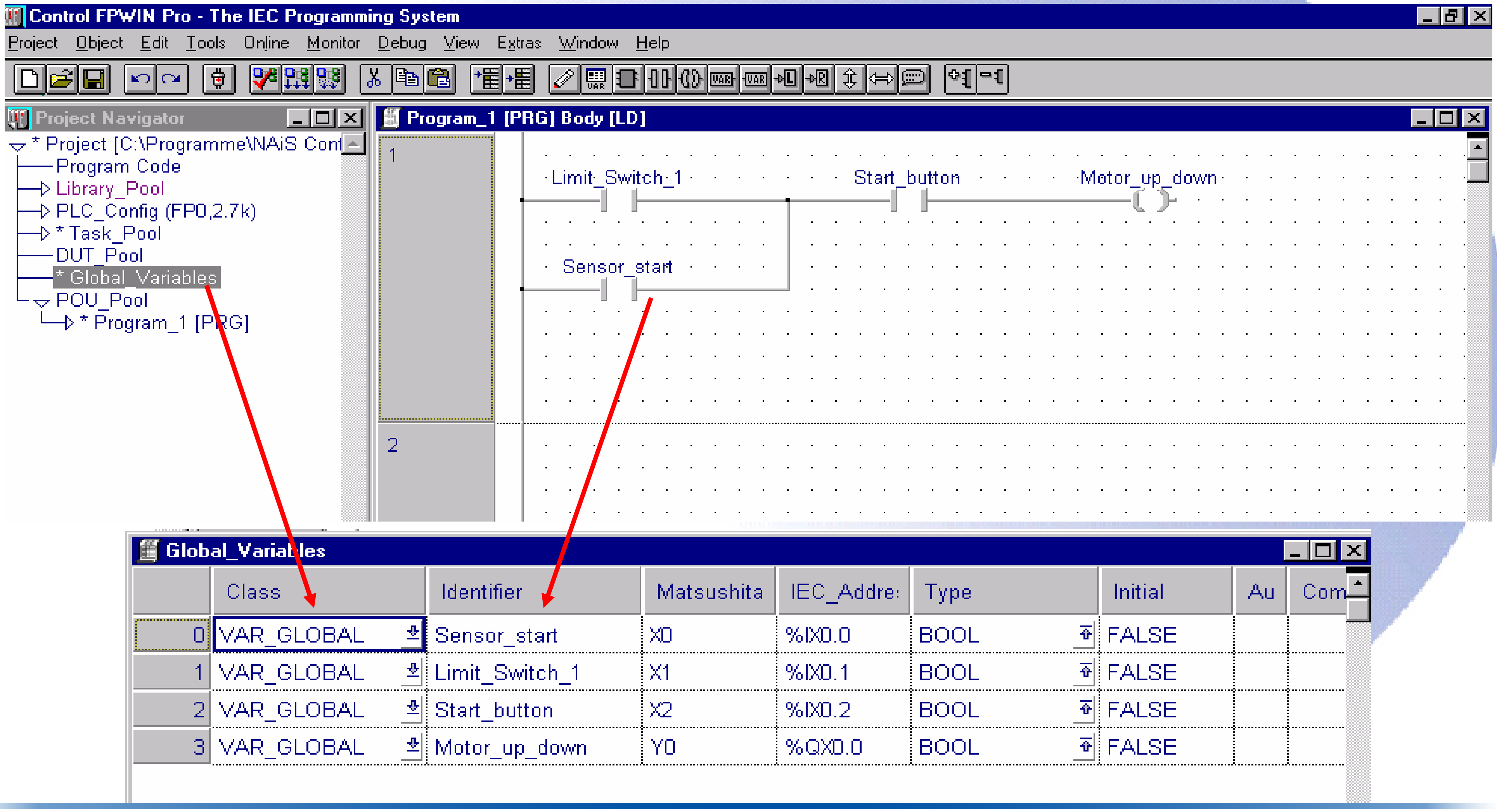
- Safe investment due to standardisation

Examples of IEC 61131-3 advantages

Panasonic
ideas for life

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



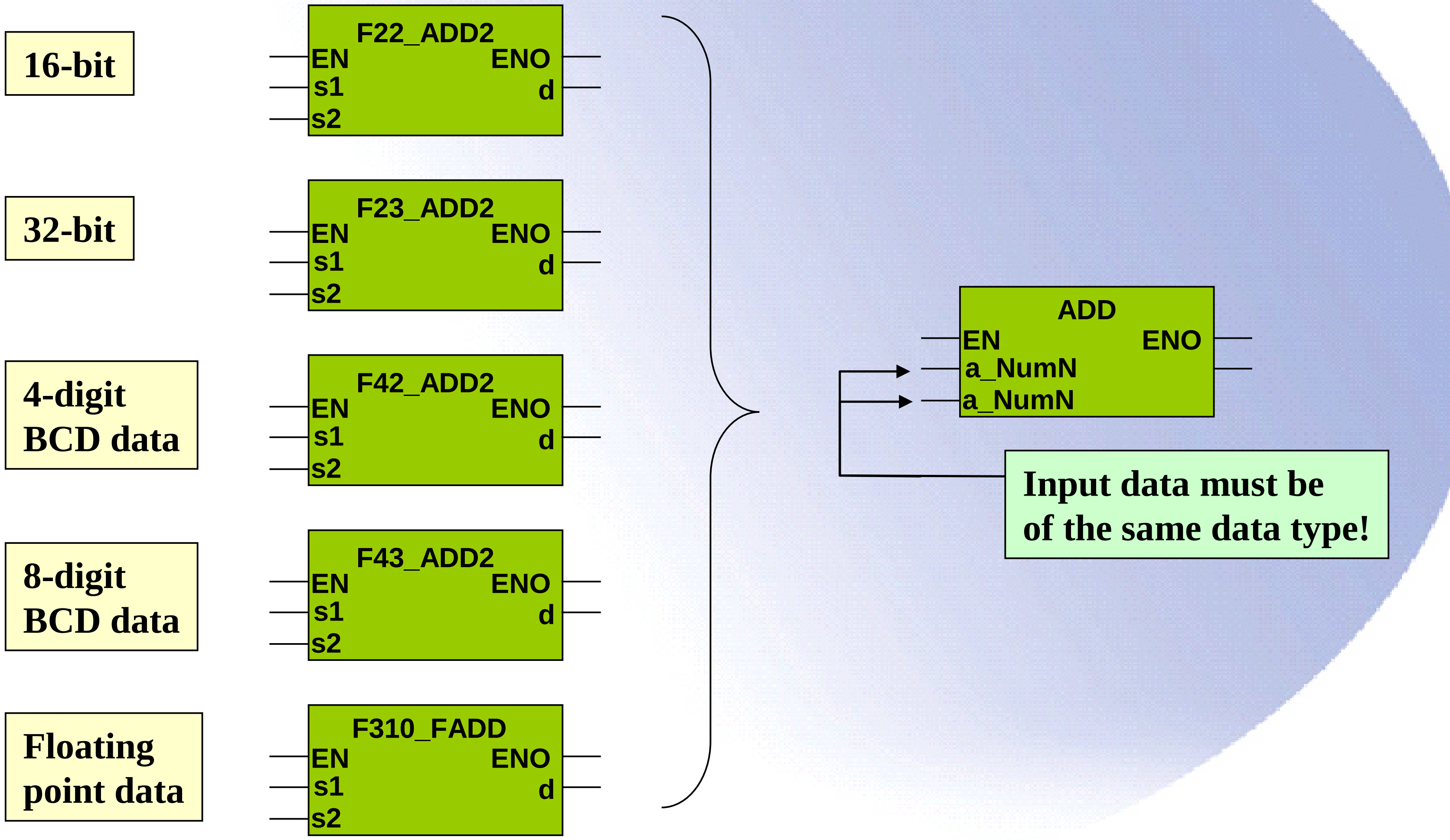
	Class	Identifier	Matsushita	IEC_Addre:	Type	Initial	Au	Com
0	VAR_GLOBAL	Sensor_start	X0	%IX0.0	BOOL	FALSE		
1	VAR_GLOBAL	Limit_Switch_1	X1	%IX0.1	BOOL	FALSE		
2	VAR_GLOBAL	Start_button	X2	%IX0.2	BOOL	FALSE		
3	VAR_GLOBAL	Motor_up_down	Y0	%QX0.0	BOOL	FALSE		

IEC 61131-3 Functions

Panasonic
ideas for life

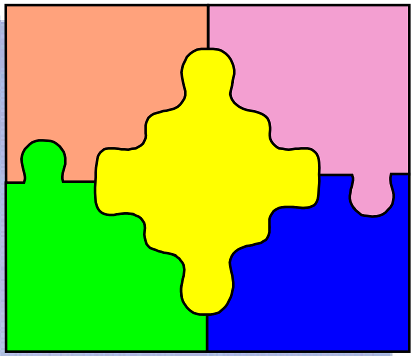
Conventional programming requires different functions for e.g.:

Flexible IEC instructions:
1 function instead of several



Input data must be of the same data type!

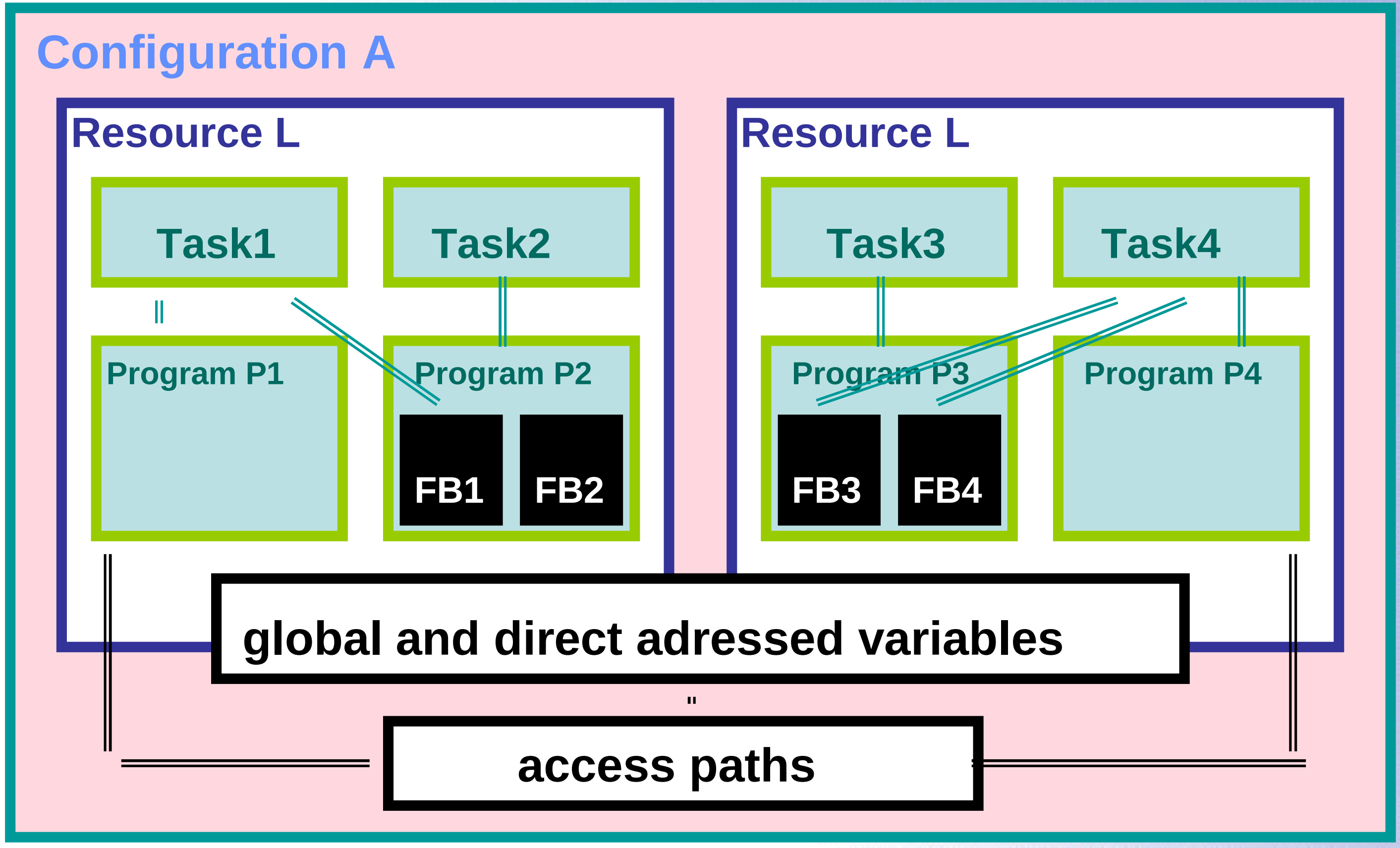
The IEC 61131 Standard - The PLC Standard



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	Technical Report

International Standard

IEC 61131-3 Software Model



Task association
Access path association

Common Elements of the IEC 61131-3

Panasonic
ideas for life

**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**

IEC 61131-3 Elementary Data Types

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

The 5 Languages of IEC 61131-3

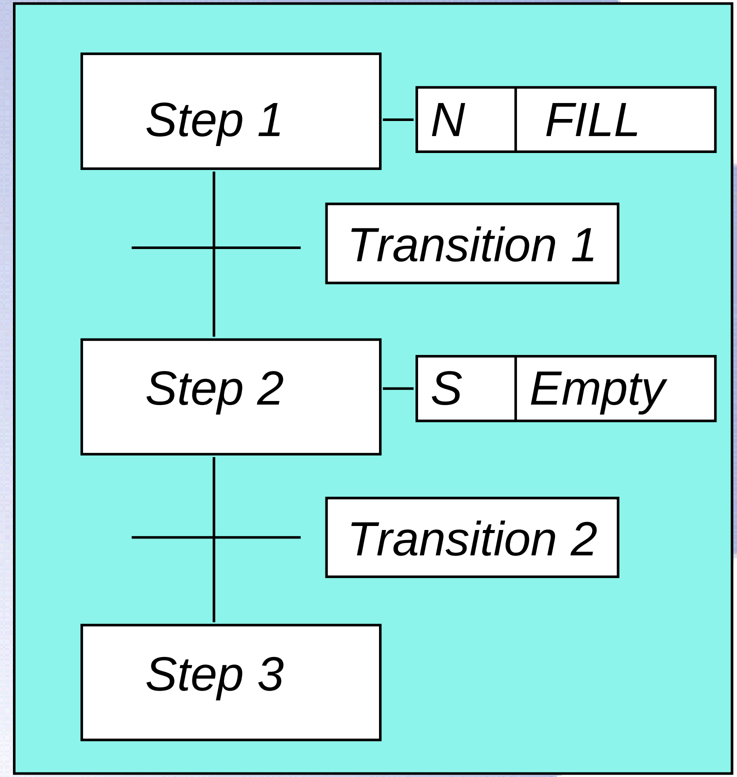
Instruction List

```
LD      A
ANDN    B
ST      C
```

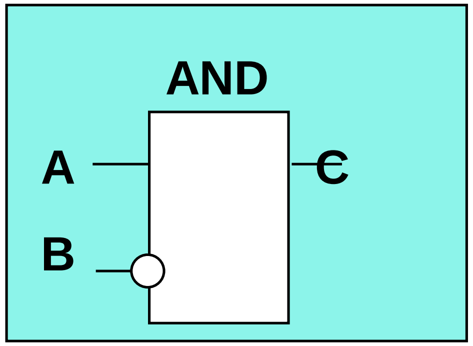
Structured Text

```
C:= A   AND  NOT B
```

Sequential Function Chart



Function Block Diagram

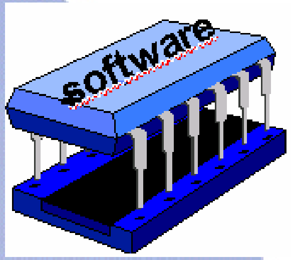


Ladder Diagram

```
 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)**

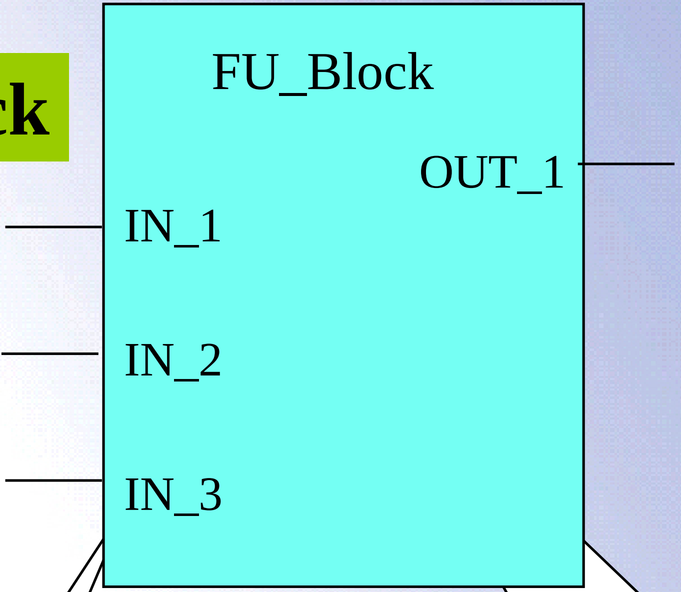


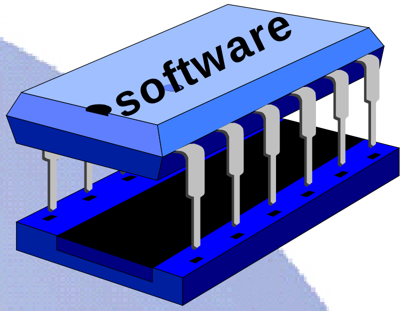
POU Type	Replicated as:	Comments
Program	Program instance	Main program
Function Block	FB instance	Subroutine with own memory, several in - and outputs possible
Function	Function	Subroutine without own memory

Function Blocks can be easily reused

Panasonic
ideas for life

1. Function Block





3. Program body

2. Variable Interface

FU-Block Header

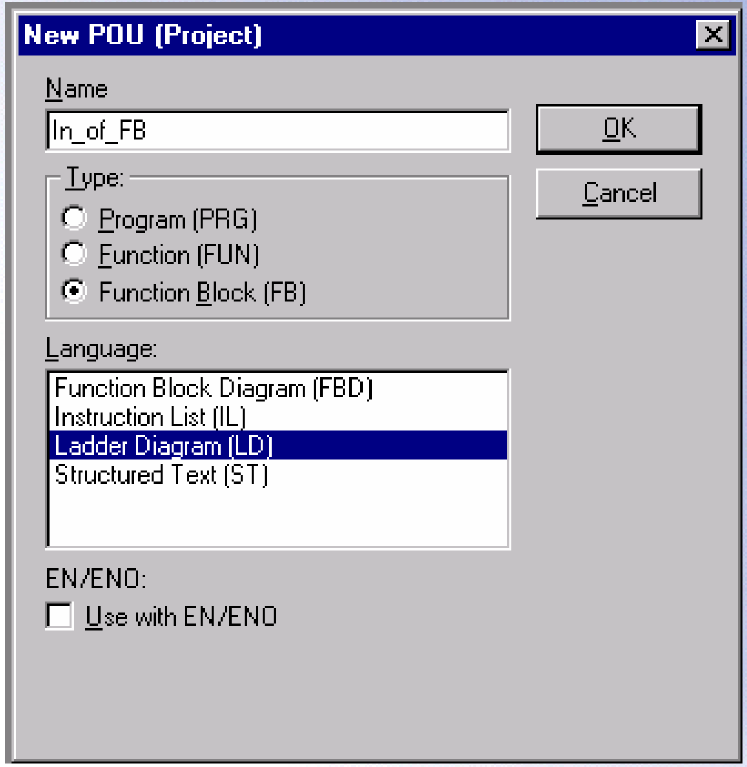
	Class	Identifier	Type
0	VAR_INPUT	IN_1	BOOL
1	VAR_INPUT	IN_2	BOOL
2	VAR_INPUT	IN_3	BOOL
3	VAR_OUPUT	OUT_1	BOOL

Program once

reuse always

```
LD     IN_1
AND    IN_2
OR     OUT_1
AND    IN_3
 :      :
 :      :
 :      :
 :      :
ST     OUT_1
```

Easy Programming of FBs and FUN


New POU (Project)

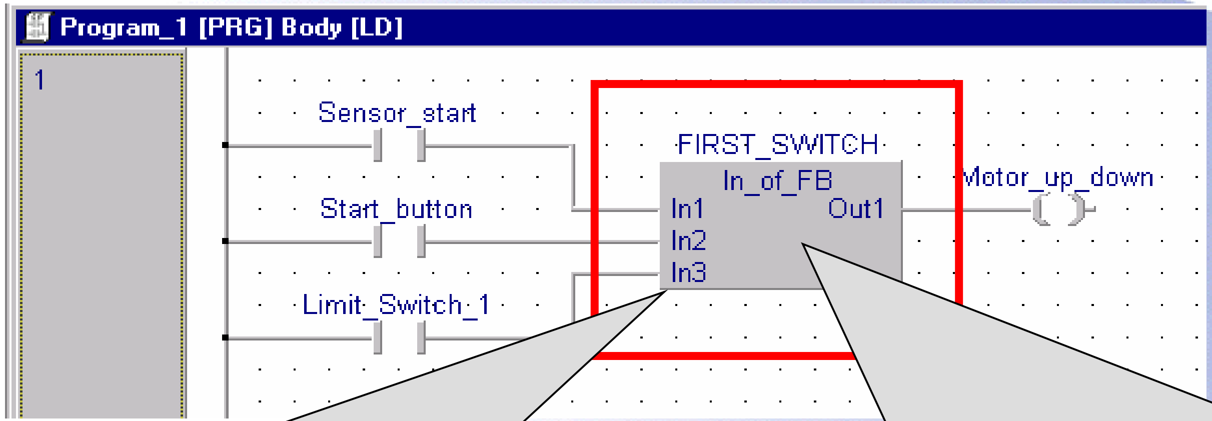
Name: In_of_FB

Type:
- ○ Program (PRG)
- ○ Function (FUN)
- ● Function Block (FB)

Language:
- Function Block Diagram (FBD)
- Instruction List (IL)
- Ladder Diagram (LD) [selected]
- Structured Text (ST)

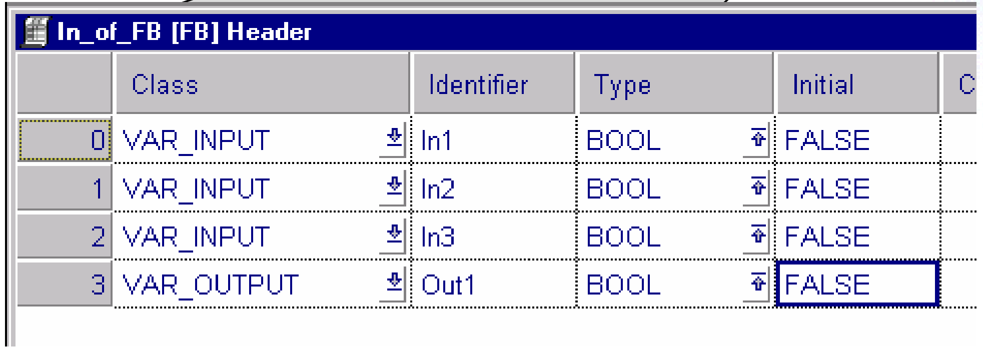
EN/ENO:
- ☐ Use with EN/ENO

Easy Programming of FBs and FUN



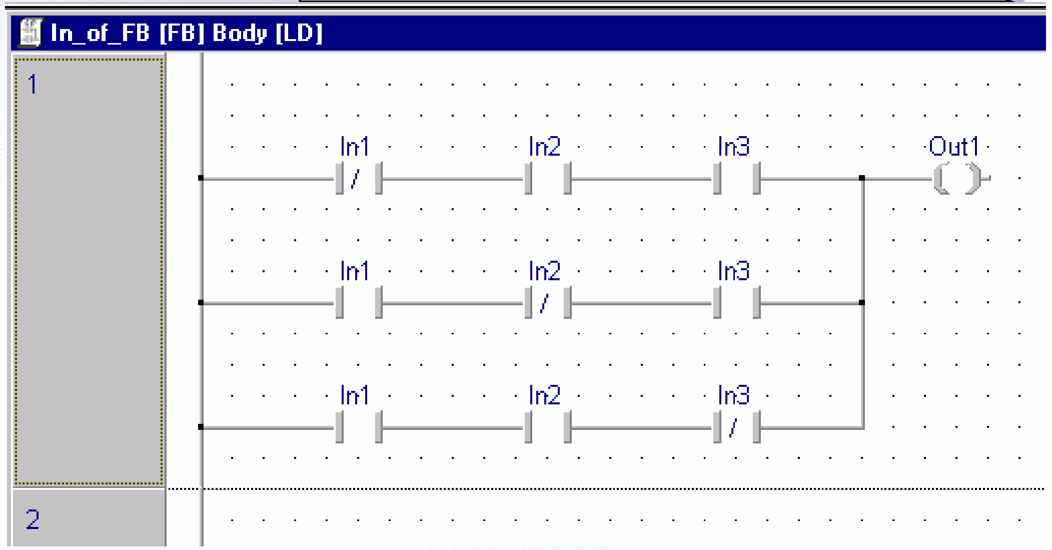
Define inputs and outputs

Program FB contents



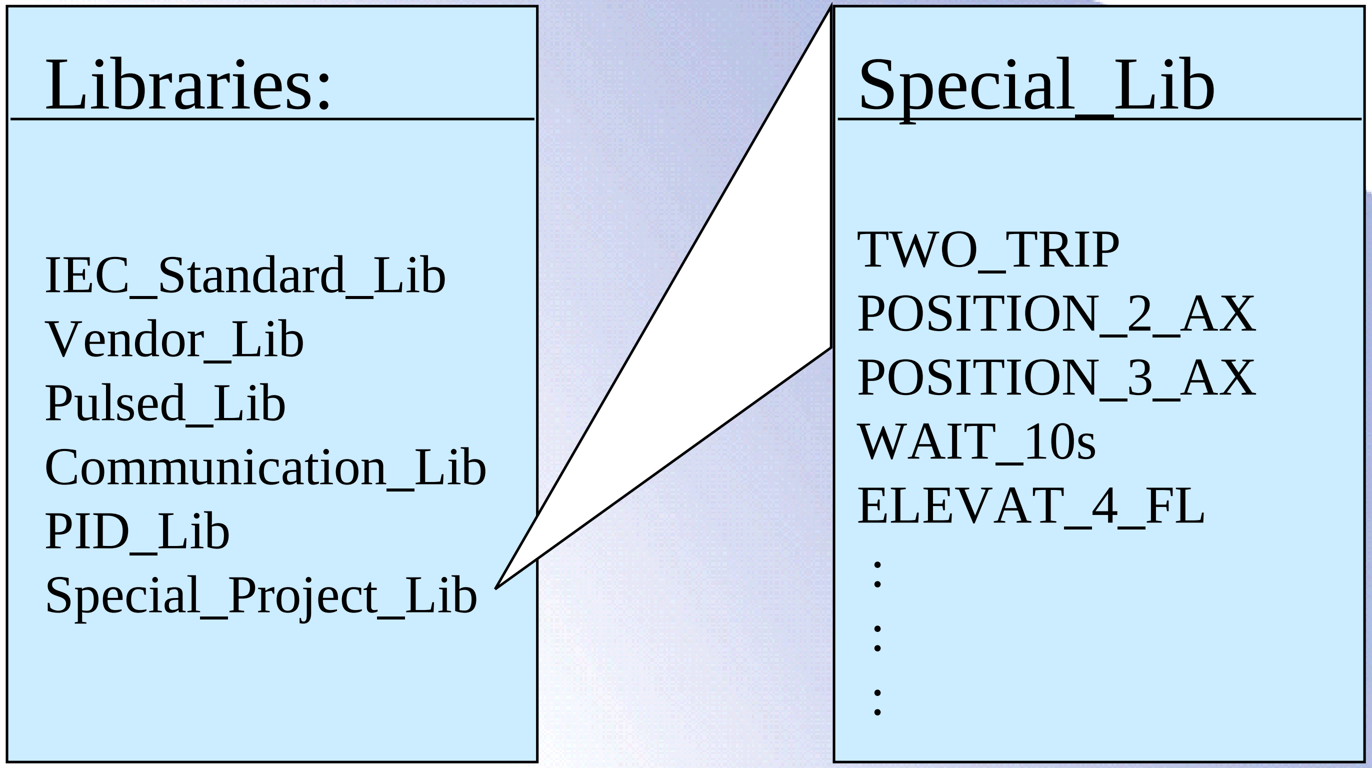
In_of_FB [FB] Header

	Class	Identifier	Type	Initial	C
0	VAR_INPUT	In1	BOOL	FALSE	
1	VAR_INPUT	In2	BOOL	FALSE	
2	VAR_INPUT	In3	BOOL	FALSE	
3	VAR_OUTPUT	Out1	BOOL	FALSE	



Panasonic ideas for life

Library Concept



Libraries:

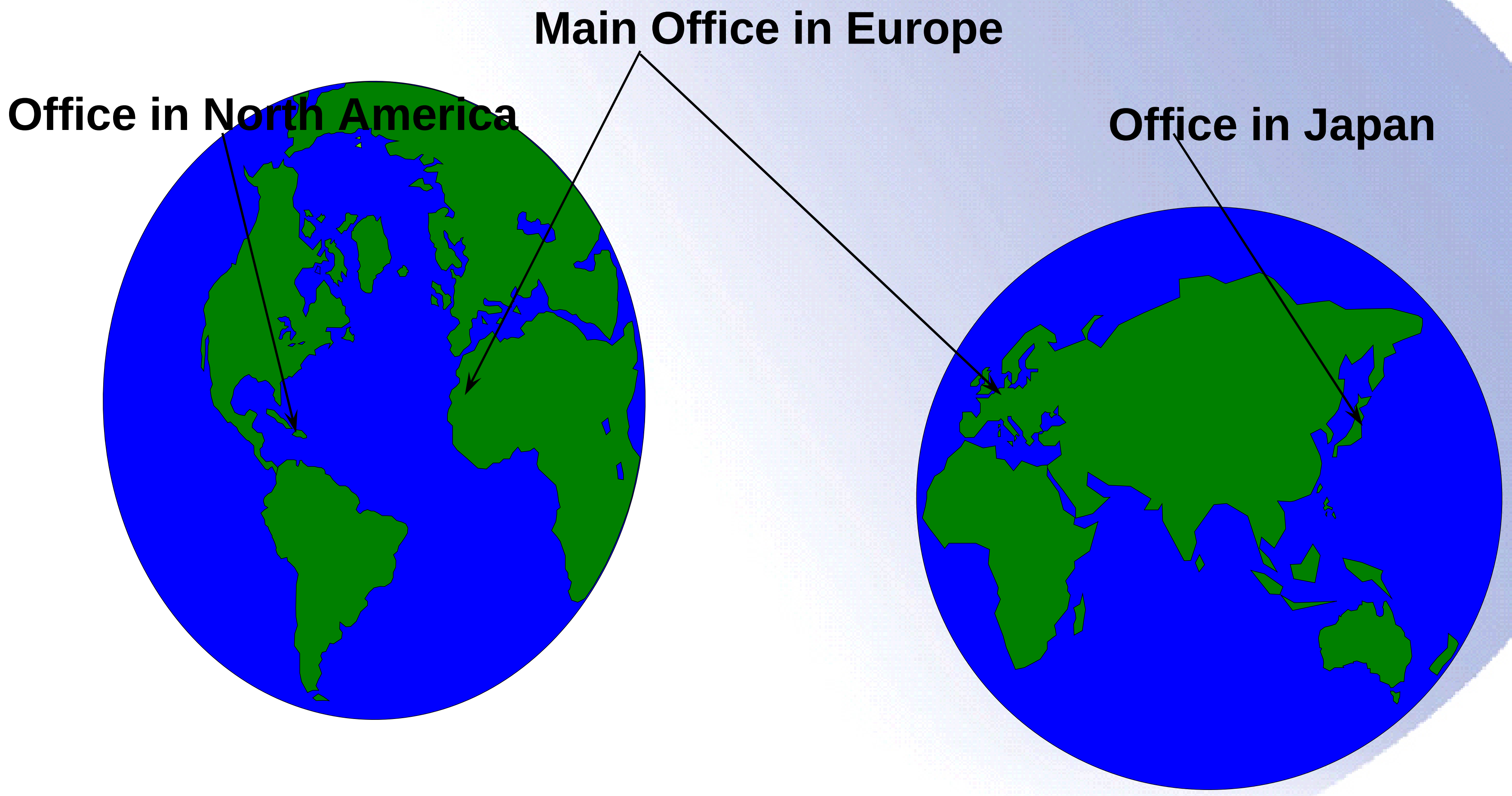
IEC_Standard_Lib
Vendor_Lib
Pulsed_Lib
Communication_Lib
PID_Lib
Special_Project_Lib

Special_Lib

TWO_TRIP
POSITION_2_AX
POSITION_3_AX
WAIT_10s
ELEVAT_4_FL
:
:
:

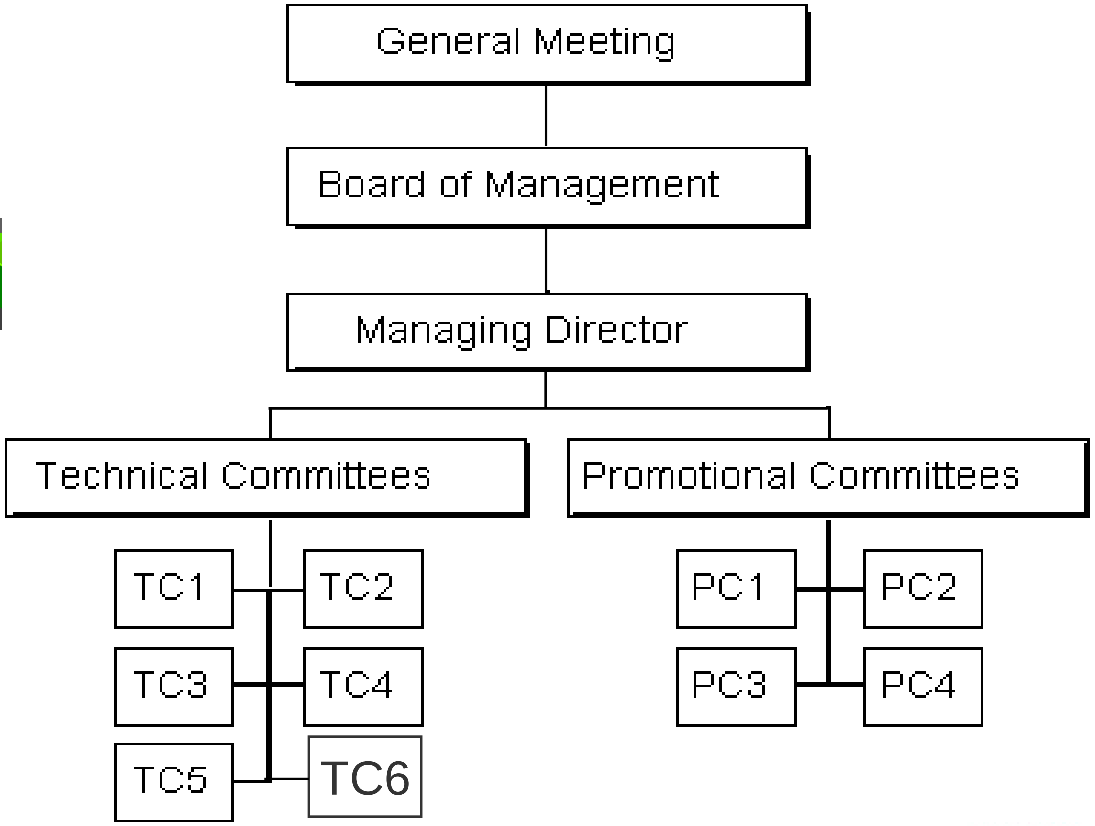
- **Self-created FBs can be stored in libraries**
- **Comfortable structuring and sorting in the libraries**
- **Know-how protection of FBs and libraries**
- **Easy reuse of tested software --> saves time**

PLCopen is a World-wide association

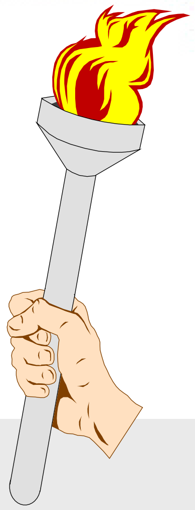


PLCopen

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.



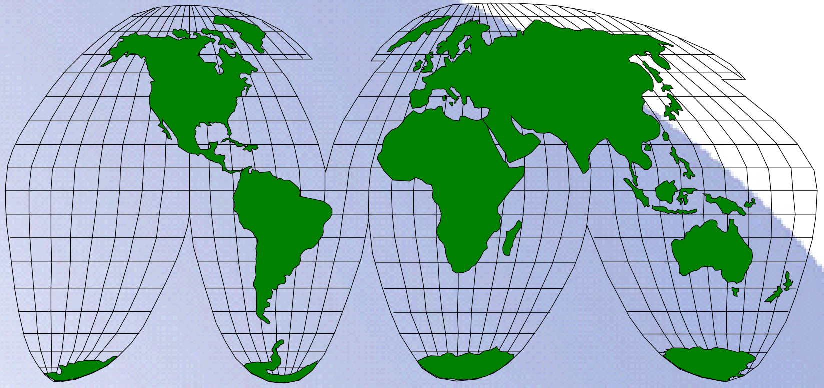
PC1 Main promotion committee
PC2 Common training program
PC3 North-America
PC4 Japan



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



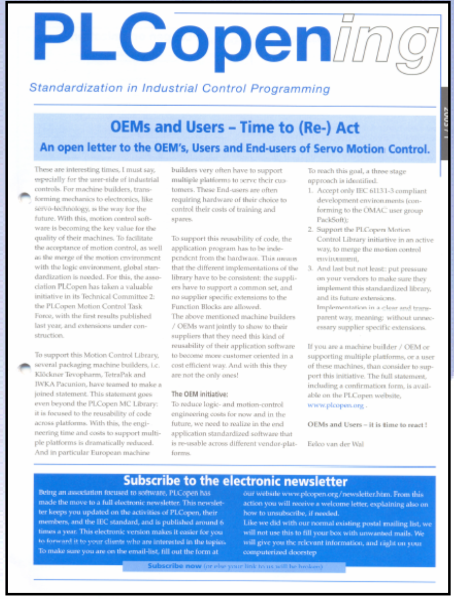
> 80 members (June 2004)

from 19 countries all over the world

Suppliers, institutes and users

See newsletter / website for up-to-date list

Details



PLCopen*ing*

Standardization in Industrial Control Programming

2003 / 1

OEMs and Users – Time to (Re-) Act

An open letter to the OEM's, Users and End-users of Servo Motion Control.

These are interesting times, I must say, especially for the user-side of industrial controls. For machine builders, transforming mechanics to electronics, like servo-technology, is the way for the future. With this, motion control software is becoming the key value for the quality of their machines. To facilitate the acceptance of motion control, as well as the merge of the motion environment with the logic environment, global standardization is needed. For this, the association PLCopen has taken a valuable initiative in its Technical Committee 2: the PLCopen Motion Control Task Force, with the first results published last year, and extensions under construction.

To support this Motion Control Library, several packaging machine builders, i.e. Klöckner Tevopharm, TetraPak and IWKA Pacunion, have teamed to make a joined statement. This statement goes even beyond the PLCopen MC Library: it is focused to the reusability of code across platforms. With this, the engineering time and costs to support multiple platforms is dramatically reduced. And in particular European machine builders very often have to support multiple platforms to serve their customers. These End-users are often requiring hardware of their choice to control their costs of training and spares.

To support this reusability of code, the application program has to be independent from the hardware. This means that the different implementations of the library have to be consistent: the suppliers have to support a common set, and no supplier specific extensions to the Function Blocks are allowed.
The above mentioned machine builders / OEMs want jointly to show to their suppliers that they need this kind of reusability of their application software to become more customer oriented in a cost efficient way. And with this they are not the only ones!

The OEM initiative:
To reduce logic- and motion-control engineering costs for now and in the future, we need to realize in the end application standardized software that is re-usable across different vendor-platforms.

To reach this goal, a three stage approach is identified.

1. Accept only IEC 61131-3 compliant development environments (conforming to the OMAC user group PackSoft);
2. Support the PLCopen Motion Control Library initiative in an active way, to merge the motion control environment.
3. And last but not least: put pressure on your vendors to make sure they implement this standardized library, and its future extensions. Implementation in a clear and transparent way, meaning: without unnecessary supplier specific extensions.

If you are a machine builder / OEM or supporting multiple platforms, or a user of these machines, than consider to support this initiative. The full statement, including a confirmation form, is available on the PLCopen website, www.plcopen.org .

OEMs and Users – it is time to react !

Eelco van der Wal

Subscribe to the electronic newsletter

Being an association focused to software, PLCopen has made the move to a full electronic newsletter. This newsletter keeps you updated on the activities of PLCopen, their members, and the IEC standard, and is published around 6 times a year. This electronic version makes it easier for you to forward it to your clients who are interested in the topics. To make sure you are on the email-list, fill out the form at our website www.plcopen.org/newsletter.htm. From this action you will receive a welcome letter, explaining also on how to unsubscribe, if needed.
Like we did with our normal existing postal mailing list, we will not use this to fill your box with unwanted mails. We will give you the relevant information, and right on your computerized doorstep

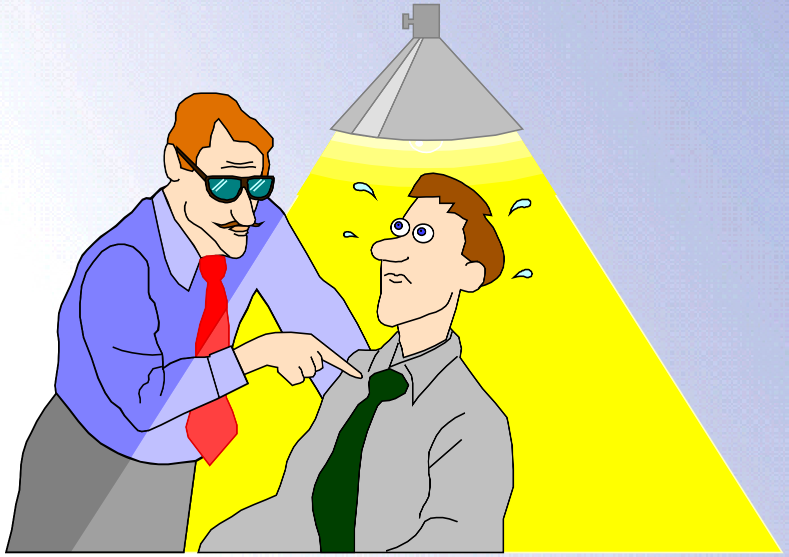
Subscribe now (or else your link to us will be broken)

The Essence of Compliancy

- The IEC 61131 standard gives rules for compliancy

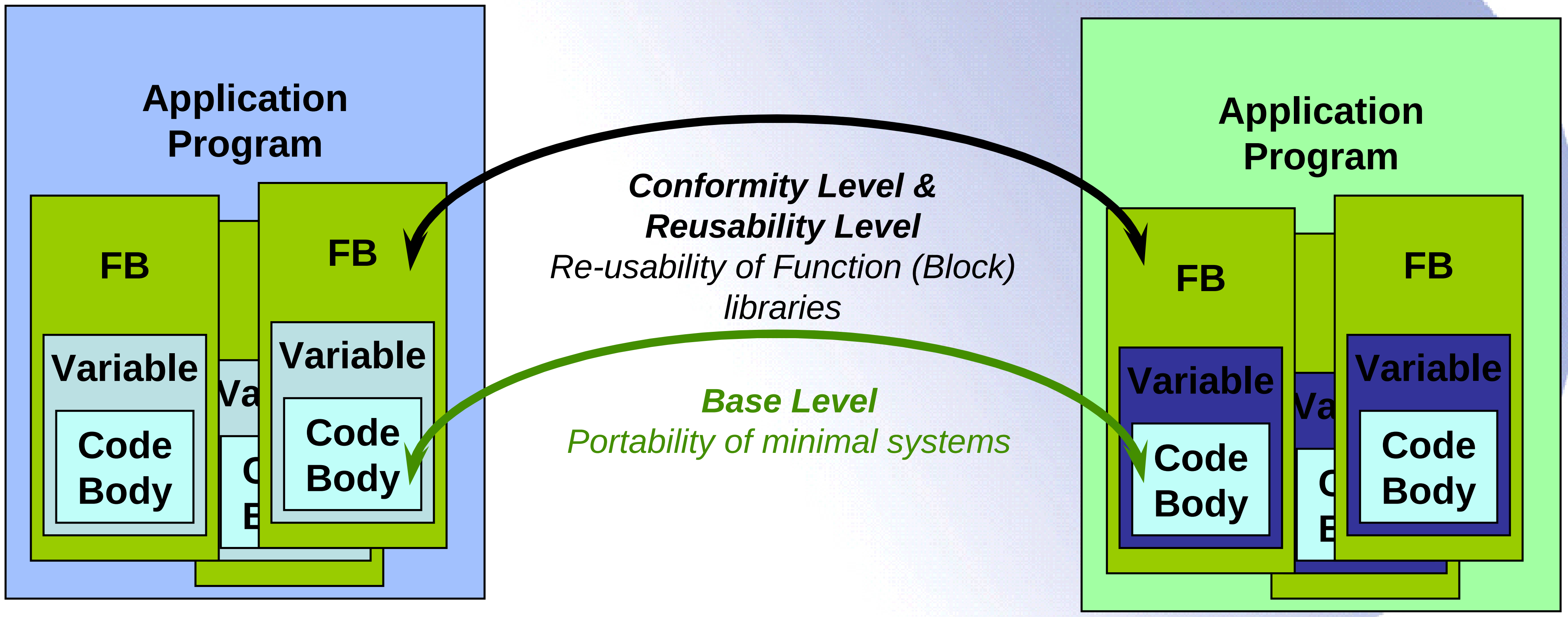
- Certification guides users towards real IEC 61131-3 programming systems (e.g. PLCopen certified products)

Without testing
there is no standard



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

TC3: PLCopen Compliance Levels



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

Base Level
Portability of minimal systems

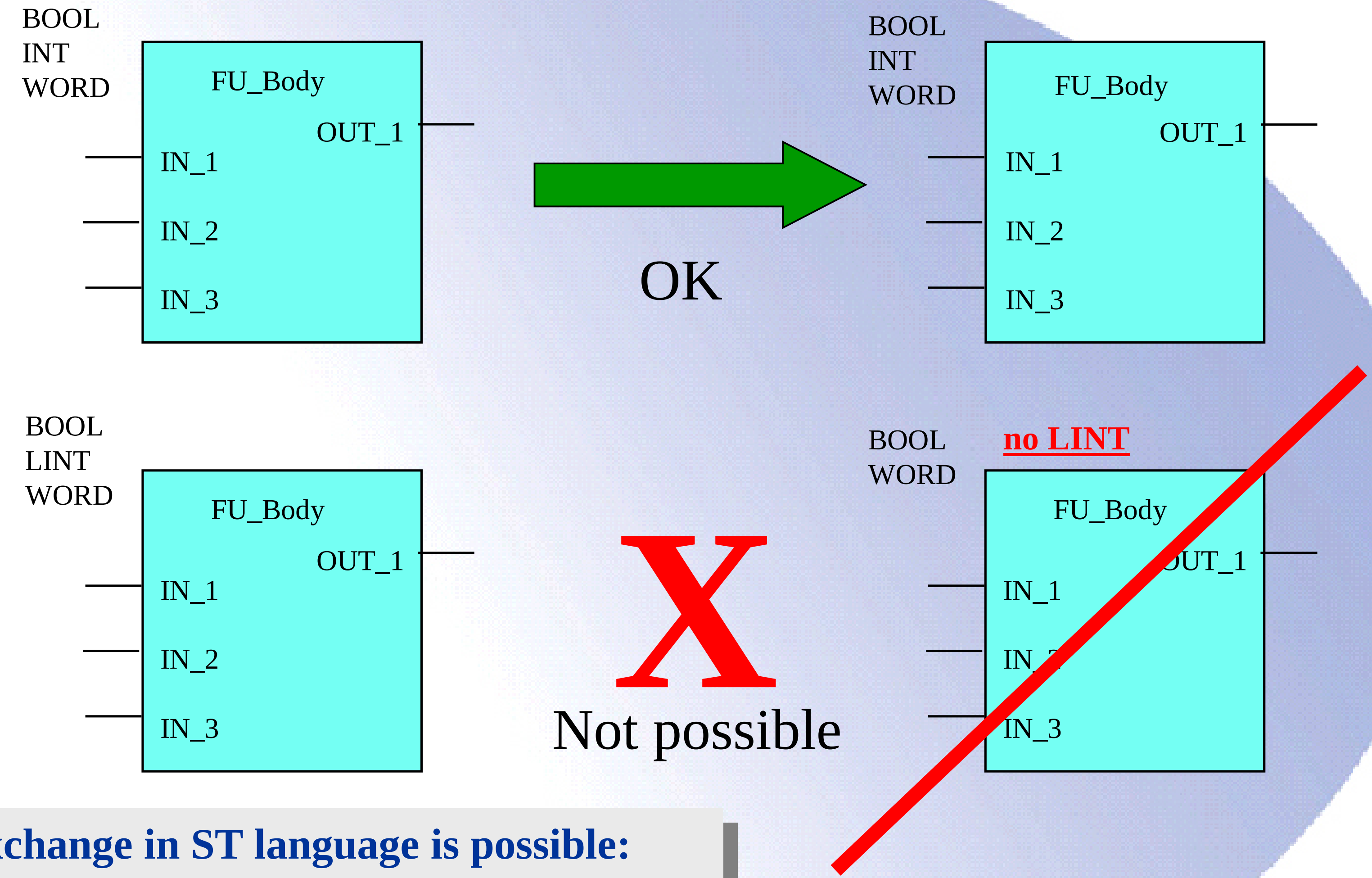
Certification

Panasonic
ideas for life

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
XML:	Opening up the development environments by specifying XML formats for IEC 61131-3

Reusability Level ST



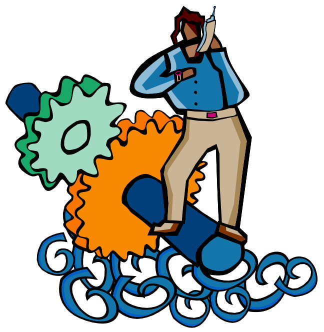
Function Block exchange in ST language is possible:

- **if both systems have the Reusability Level ST**
- **the used instructions are IEC 61131-3 instructions**
- **the same data types are available**

Why Motion Control FBs ?

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

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

- ü **Initiated by Users**
- ü **..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

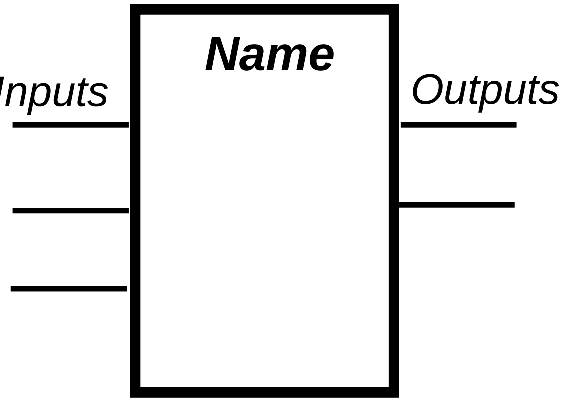
Users:
- Bosch Packaging
- Kuka
- Kloeckner Tevopharm
- Focke EKB
- Hershey Foods
- Tetra Pak

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

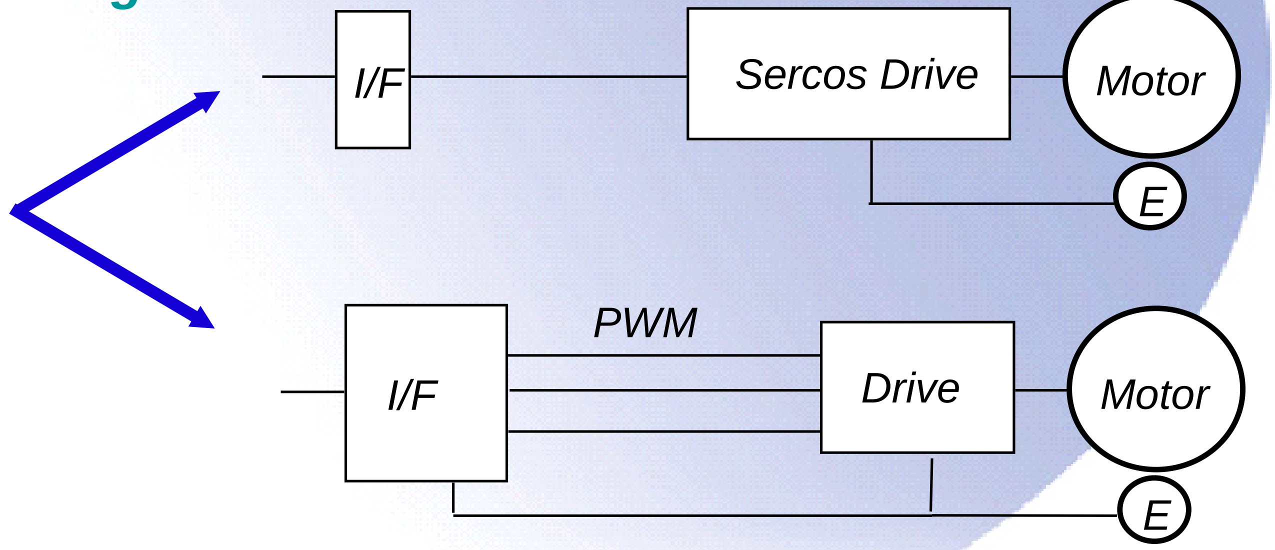
HW Independence via Function Blocks

Software View

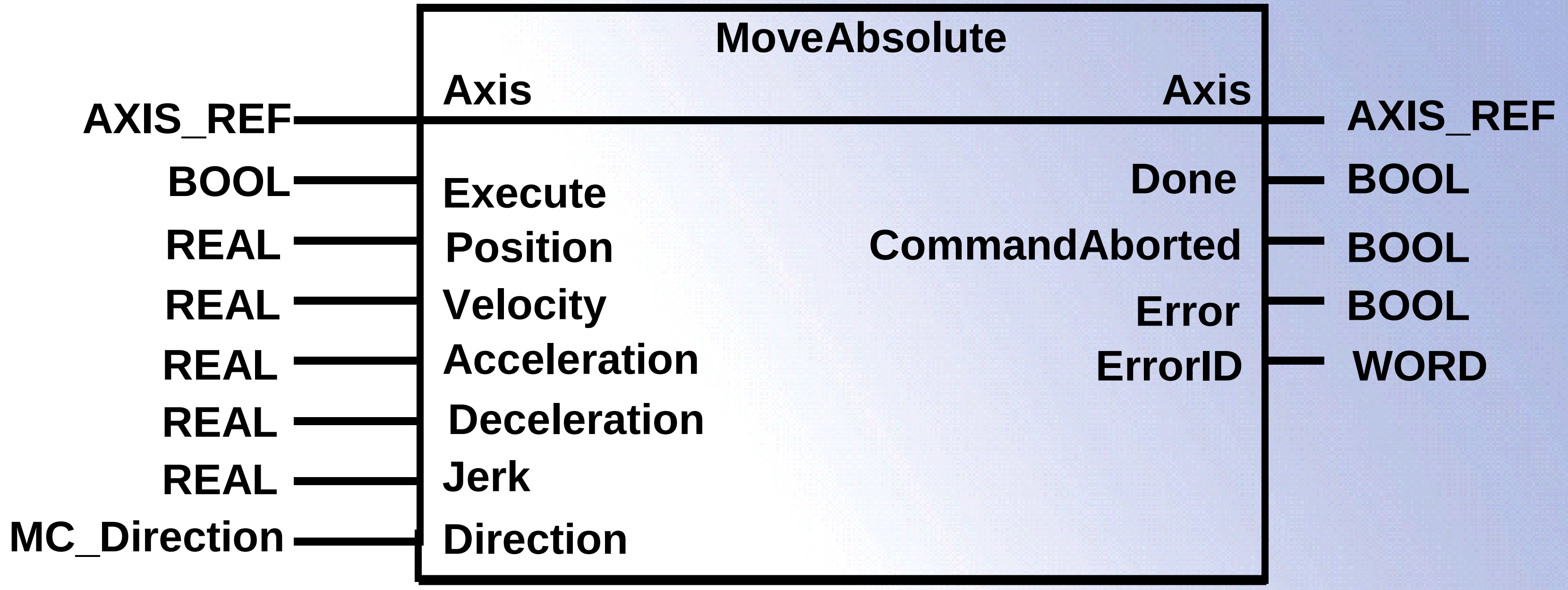
Encapsulation / Information Hiding



Hardware View



Example of a Function Block



FB-Name	**MC_MoveAbsolute**
This function block commands a controlled motion at a specified absolute position.	

Panasonic
ideas for life

Thank you !