
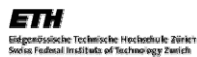


# Programming a Programmable Logic Controller (PLC)

Second part

Jens Bathelt



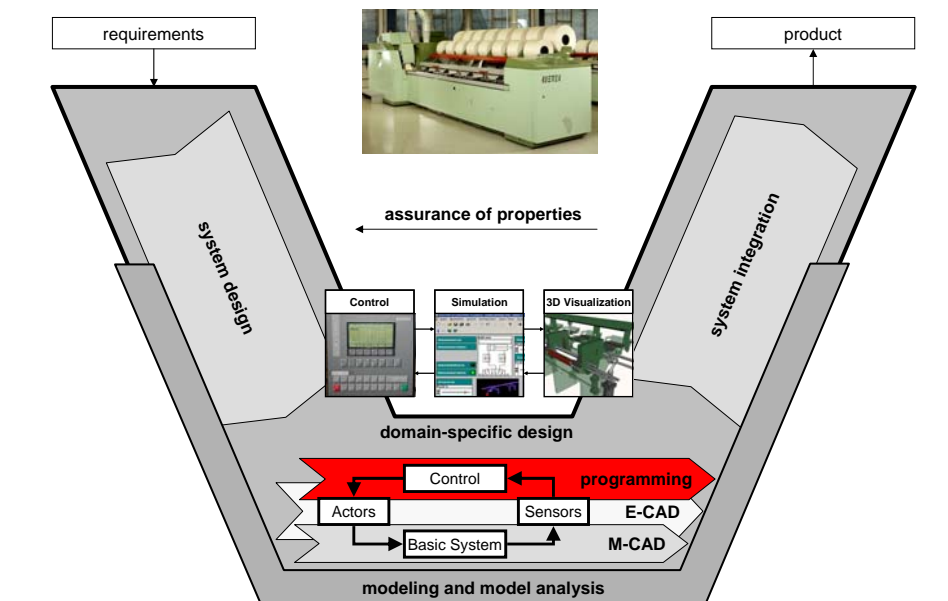
## Content of Module 3

- Lesson 3.1, 1h: deepening Programmable Logical Controllers (PLC) and explaining Sequential Function Charts (SFC).
- Case 3.1, 2h: Programming SFC using SIMATIC S7-Graph
- Lesson 3.2, 1h: The remaining four IEC-1131 languages and State Transition Diagrams (STD)
- Case 3.2, 2h: Deepening SFC using SIMATC S7-Graph

## Objectives

- Able to arrange all five IEC-1131 languages and STD (State Transition Diagram) according to the software development process
- Classification of PLC programming language properties like graphical language, close to machine language and high level language like C.
- Explanation of Basic principles and procedures while using STD as a programming language.
- Difference between a PLC and a Soft-PLC
- Getting a grip of the IEC 61499 (decentralized intelligence)

## Leadslide



## The PLC software development versus IEC 1131-3

— **STD, State Transition Diagram** (SIEMENS: HiGraph)

— **SFC, Sequential Function Chart** (AS, Ablaufsprache)

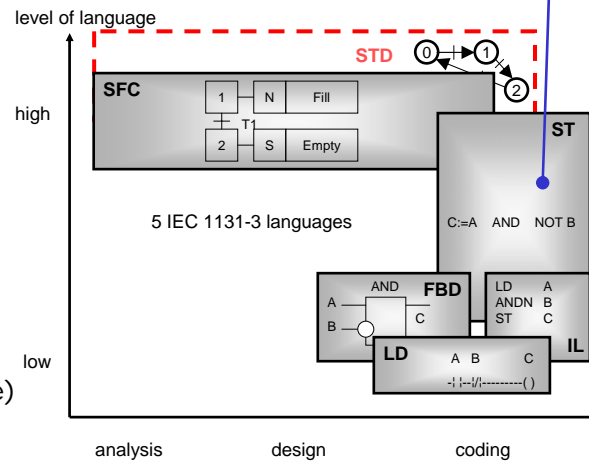
— **ST, Structured Text**

— **FBD, Function Block Diagram** (FUP, Funktionsplan)

— **IL, Instruction List** (AWL, Anweisungsliste)

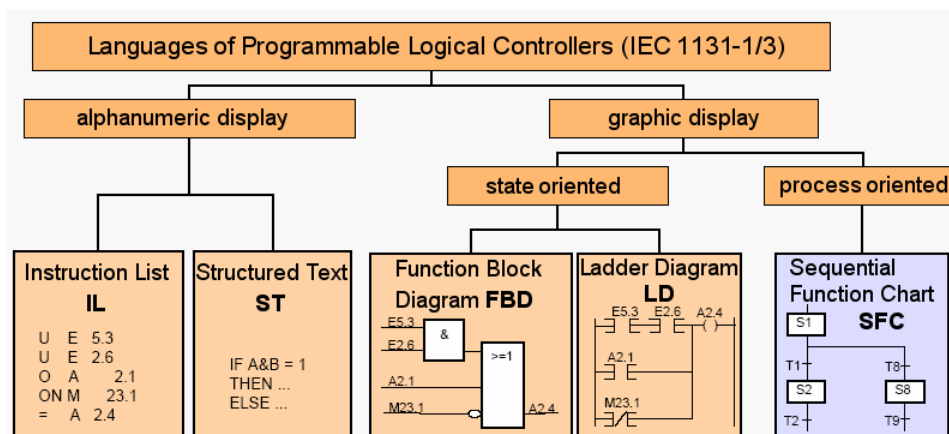
— **LD, Ladder Diagram** (KOP, Kontaktplan)

Other languages (like C, Basic, ..) are possible on many PLC's



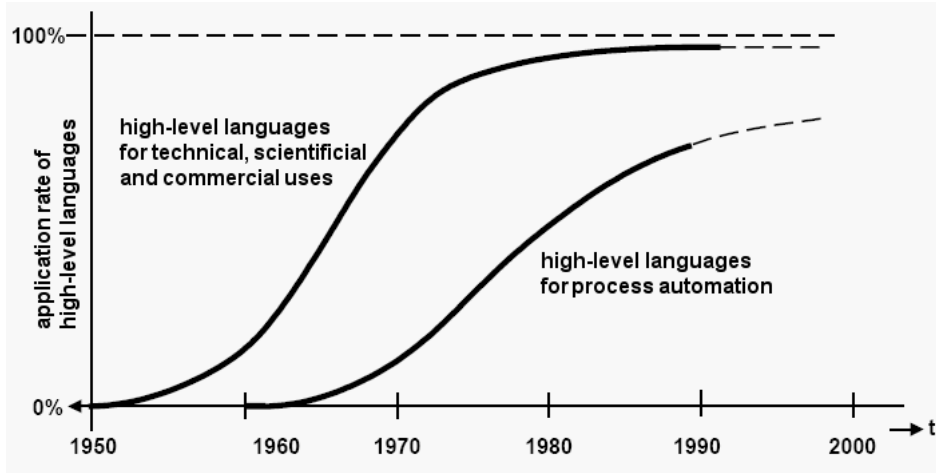
Bonfatti F., Monari P. D. and Sampieri U.: IEC 1131-3 Programming Methodology, CJ International, France, 1997

## IEC 1131 programming languages classification



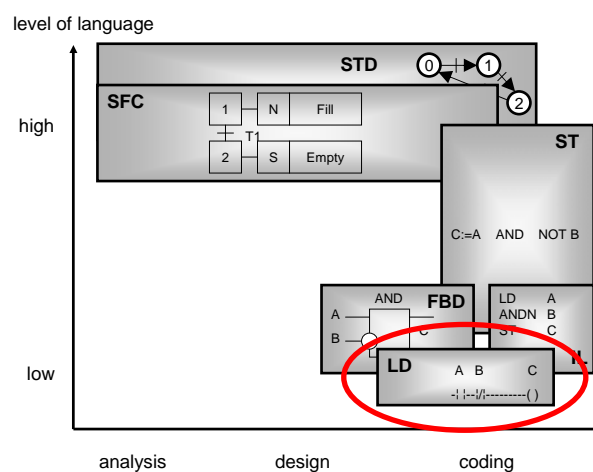
## Programming language level

- Real time requirement to control a machine
- Robust and not the newest hardware in PLC's (CPU, memory)



IAS Universität Stuttgart, 2003

## LD, Ladder Diagram (KOP, Kontaktplan)



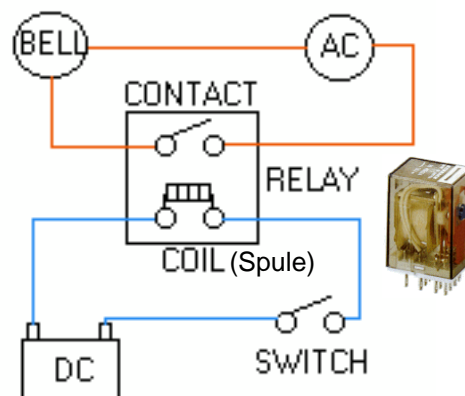
## LD, facts

- Derived from the pre-PLC relay based controls
- low level language
- graphical language
- SIEMENS: LAD/KOP
- Ladder: 'Leiter'

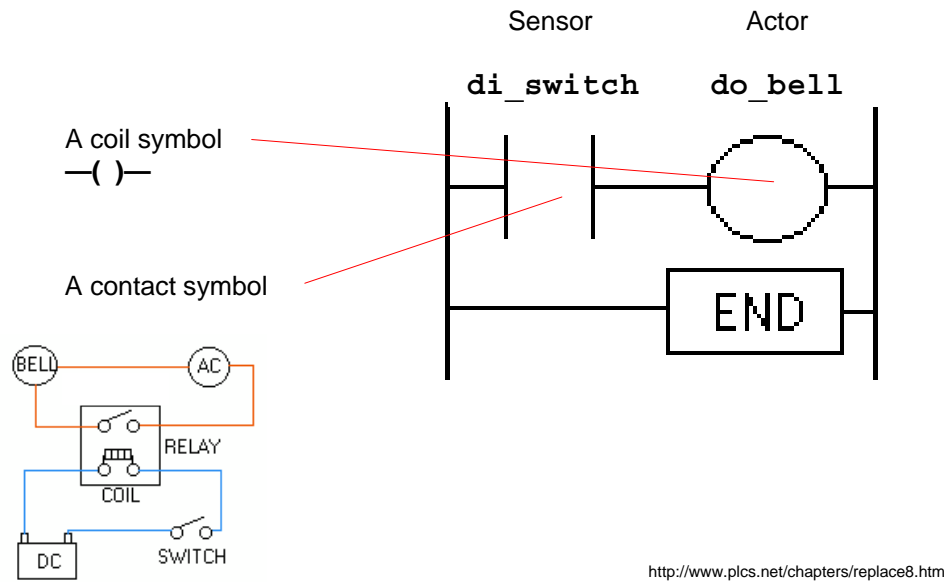


## LD, Relays

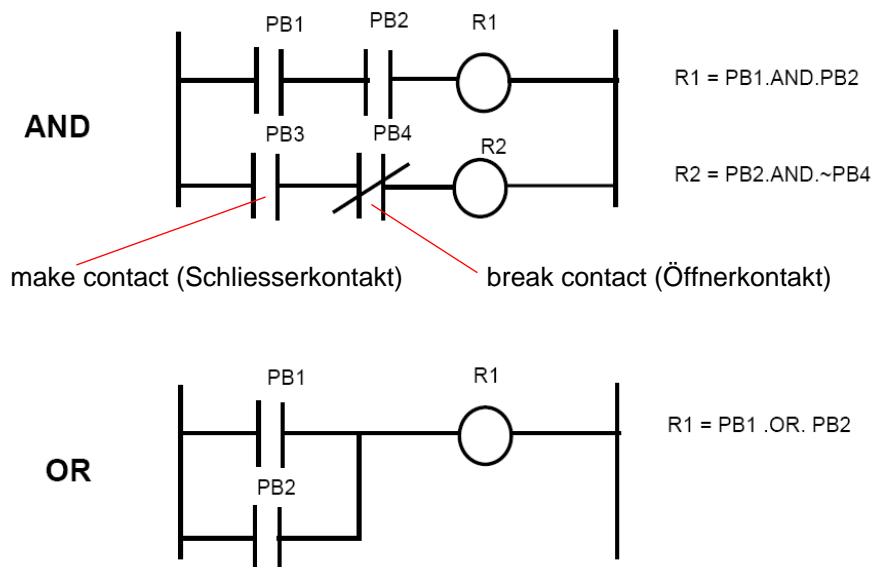
- Relays were used to control the digital machine events in the past
- PLC's were invented to increase the control power



## LD, a simple program based on the former relay example



## LD, logic basics

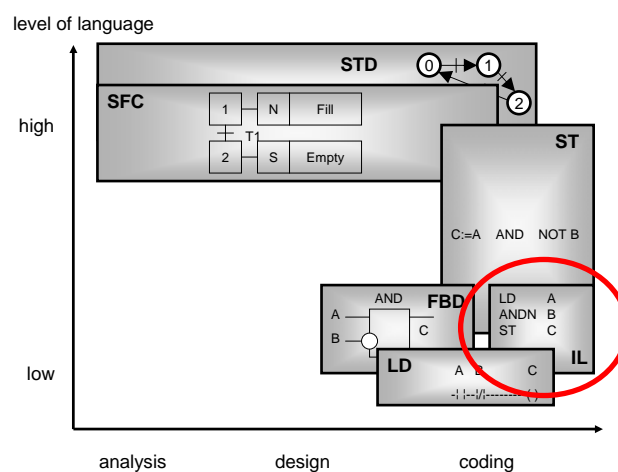


## LD, ongoing example



The outputs (Ausgang) A1 **and** A2 are only set, if either the input (Eingang) E3 is set **or** E1 **and** E2 are set simultaneously.

## IL, Instruction List (AWL, Anweisungsliste)



## IL, facts

- Close to assembler (Maschinensprache)
- low level language
- textual language
- SIEMENS: AWL
- Old language, many experienced users, a lot of generated code in use in industry, hard to maintain, hard to read for externals, hard to handle in larger projects, fast, minimal memory usage, no programming structure.
- The internal representation of all languages in the SIMATIC standard edition is IL!

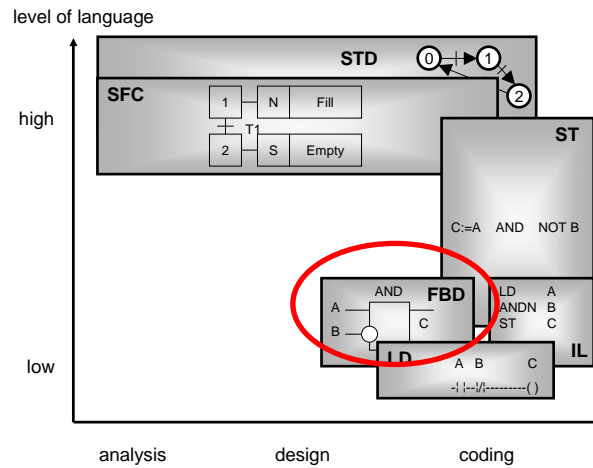
## IL, ongoing example

and (und)	U (	E1
	U	E2
	)	
or (oder)	O	E3
	=	A1
	=	A2

The outputs (Ausgang) A1 **and** A2 are only set, if either the input (Eingang) E3 is set **or** E1 **and** E2 are set simultaneously.



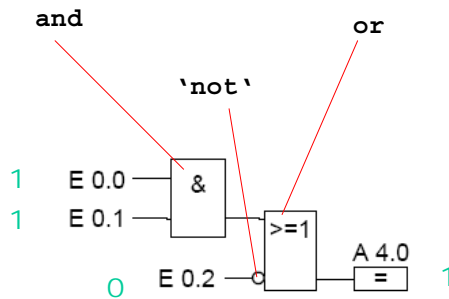
## FBD, Function Block Diagram (FBS, Funktionsbausteinsprache)



## FBD, facts

- Drawing functions blocks to express logics (like and/or/not) according to DIN 40 700 and DIN 19 239 analog to signal flows observed in electronic circuit diagrams (Stromlaufplan).
- low level language
- graphical language
- SIEMENS: FUP (Funktionsplan)

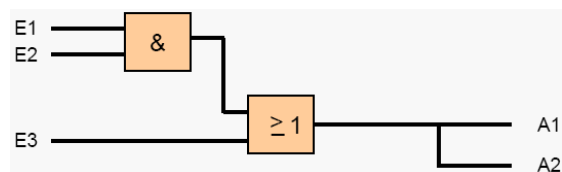
## FBD, logic basics



A 4.0 is set if

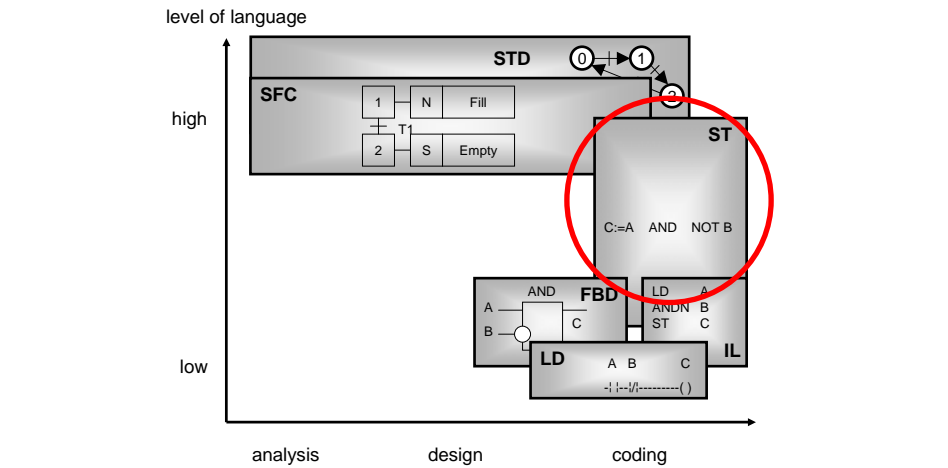
- E 0.0 AND E 0.1 are set
- OR E 0.2 is NOT set

## FBD, ongoing example



The outputs (Ausgang) A1 **and** A2 are only set, if either the input (Eingang) E3 is set **or** E1 **and** E2 are set simultaneously.

ST, Structured Text (Strukturierter Text)



ST, facts

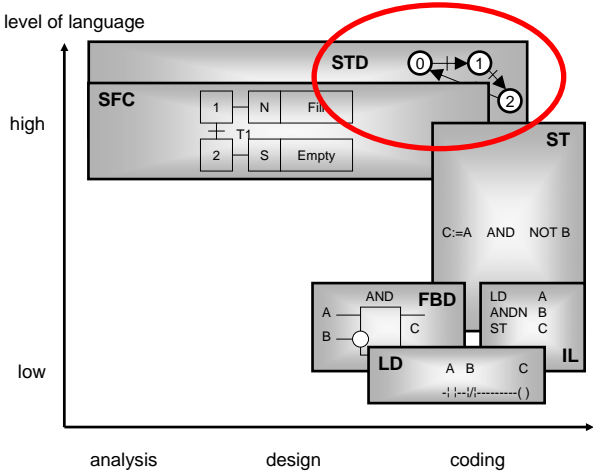
- Close to Modula-2, Pascal, Basic and C.
- High level language (Hochsprache)
- Textual language
- SIEMENS: SCL (Structured Control Language)
- Like all PLC languages embedded in the PLC cycle loop. (That is the major difference to the ordinary use of C, Pascal, ..)

ST, ongoing example

```
A1 :=      E3 OR (E1 AND E2) ;
A2 :=      A1 ;
```

The outputs (Ausgang) A1 **and** A2 are only set, if either the input (Eingang) E3 is set **or** E1 **and** E2 are set simultaneously.

## STD, State Transition Diagram (Zustandsgraph)



## STD, facts

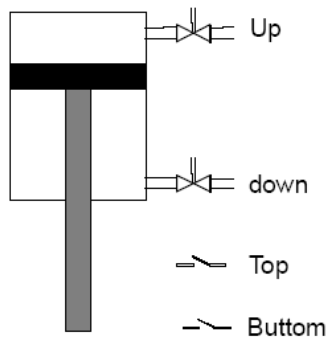
- STD was/is used as a sketching language to describe the high-level states of a mechatronic system on a piece of paper.
- high level language
- graphical language
- SIEMENS: HiGraph, better for the high-level structure design than Graph (SFC) when designing komplex control software
- Useful as well for mechanical engineers, initial operators (Inbetriebsetzer), service engineers,.. → interdisciplinary language like SFC.

## STD, elements

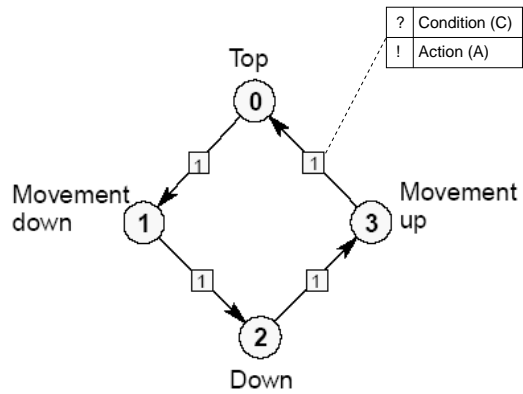
- States (NOT functions like in SFC)
- Transitions (PLC-input)
  - Between the states
- Actions (PLC-output)
  - After fulfilled transition condition
  - SIEMENS:
    - Entering the state
    - During the state
    - Exiting the state
- SIEMENS: IL (AWL) is used to describe the transition conditions and the actions

## STD, easy example

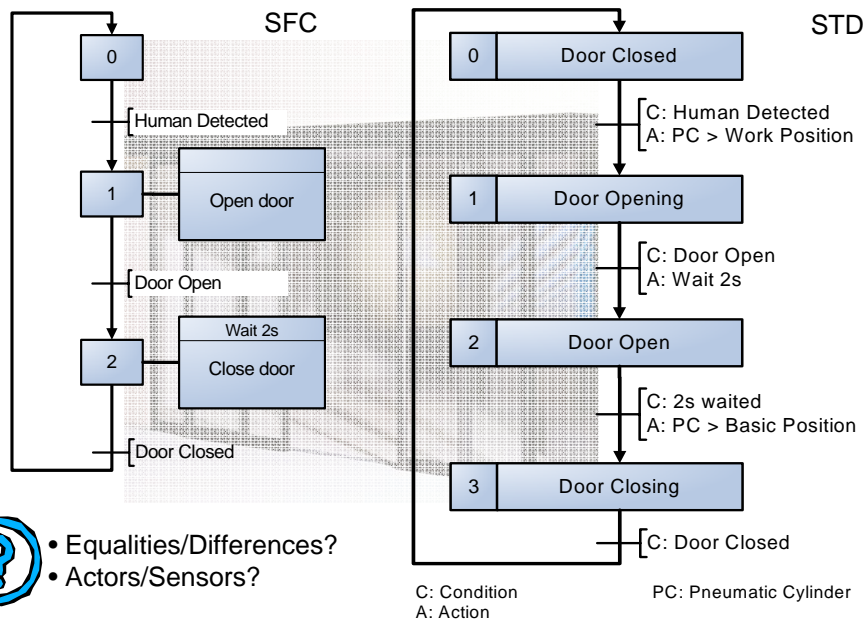
Valves with the states  
"Top" and "Bottom" and  
the movements "Up" and "Down"



Representation of states  
in a state graph

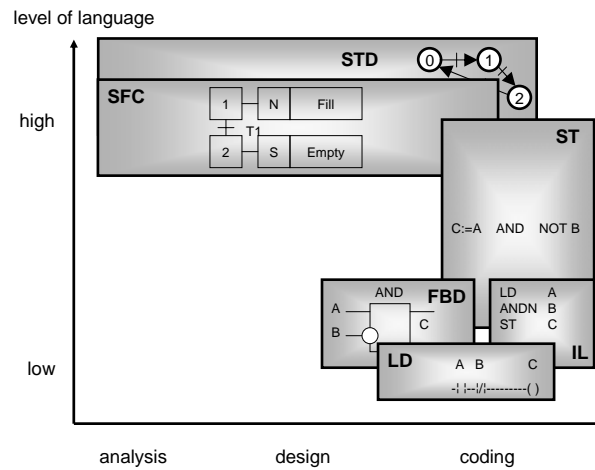


## SFC versus STD



## Summarizing all discussed languages

	LD	IL	FBD	ST	SFC	STD
Low level	X	X	X			
High level				X	X	X
Textual		X		X		
Graphical	X		X		X	X
SIEMENS	LAD/KOP	AWL	FUP	SCL	Graph	HiGraph
German	KOP	AWL	FBS	ST	AS	Zustandsdiagram



## Soft PLC versus PLC



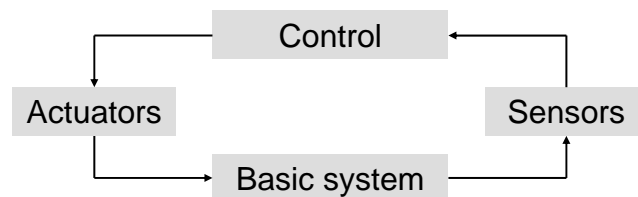
## PLC (Programmable Logic Controller)

- A PLC (i.e. Programmable Logic Controller) is a device that was invented to replace the necessary sequential relay circuits for machine control. The PLC works by looking at its inputs and depending upon their state, turning on/off its outputs.



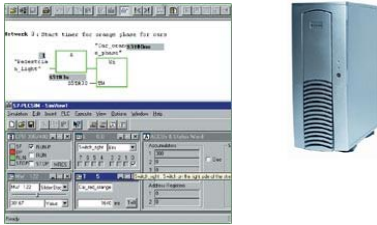
<http://www.plcs.net/>

## Machine controlled by a PLC

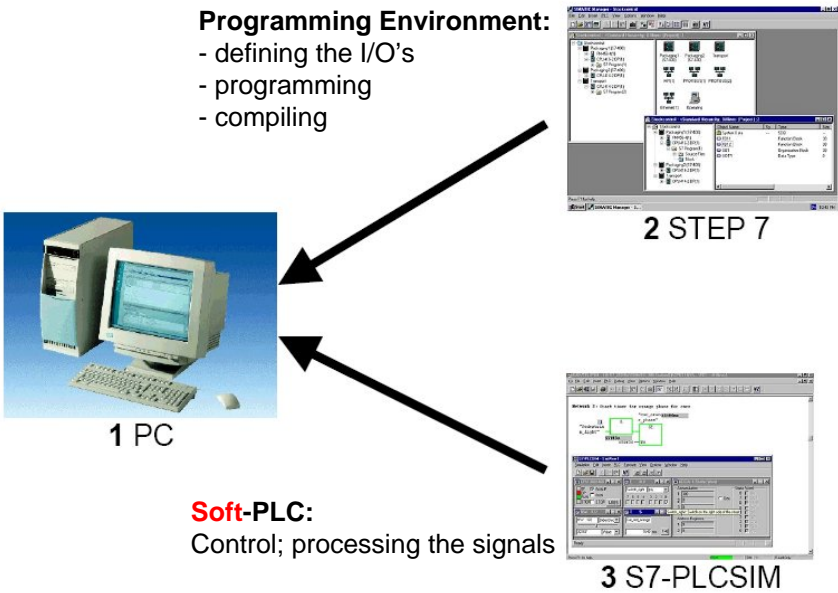




## Machine controlled by a Soft-PLC



## SIEMENS Soft-PLC: PLCSim

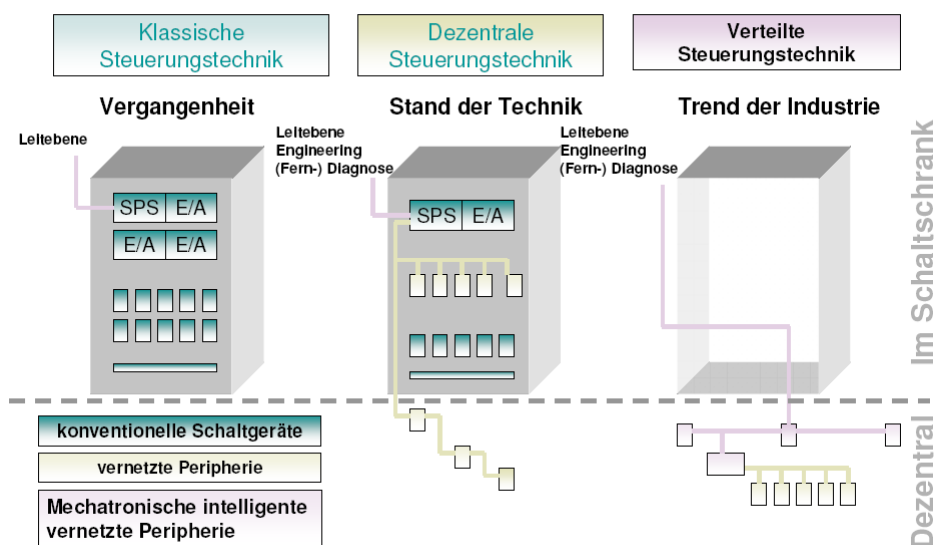


## Trends in automation

- Wireless Automation
- Combining PLC & CNC  
(SIEMENS: SIMATIC, SINUMERIK → SIMOTION)
- Decentralized intelligence

## Decentralized intelligence

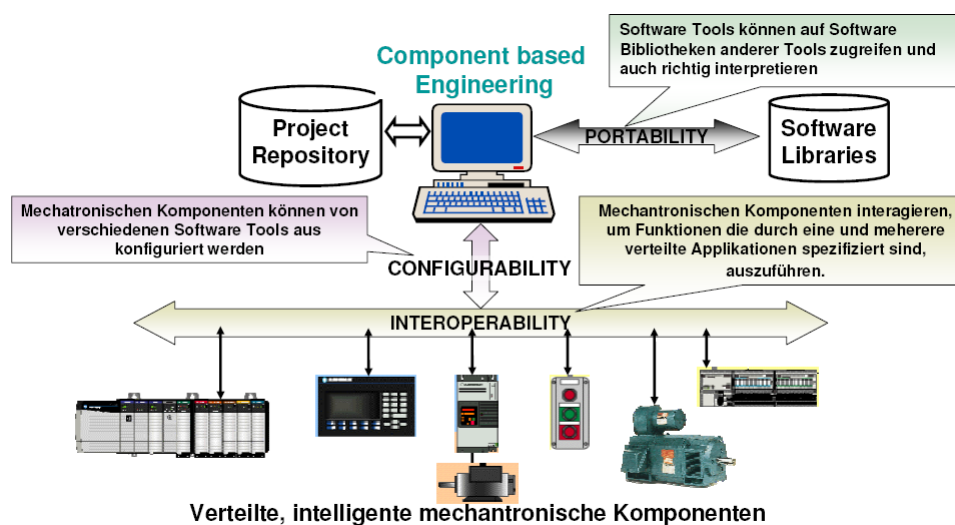
### Vom zentralen Ansatz zu verteilten Architekturen



## Why decentralized intelligence?

- Verified and reliable modules can be reused
- It is easier to extend HW/SW due to the encapsulation of intelligence
- Faster initial operation due to envisioned “plug&play”
- Manufacturer can enhance their components by SW (intelligent actuators, ..)
  - ➔ The Interface has to be standardized

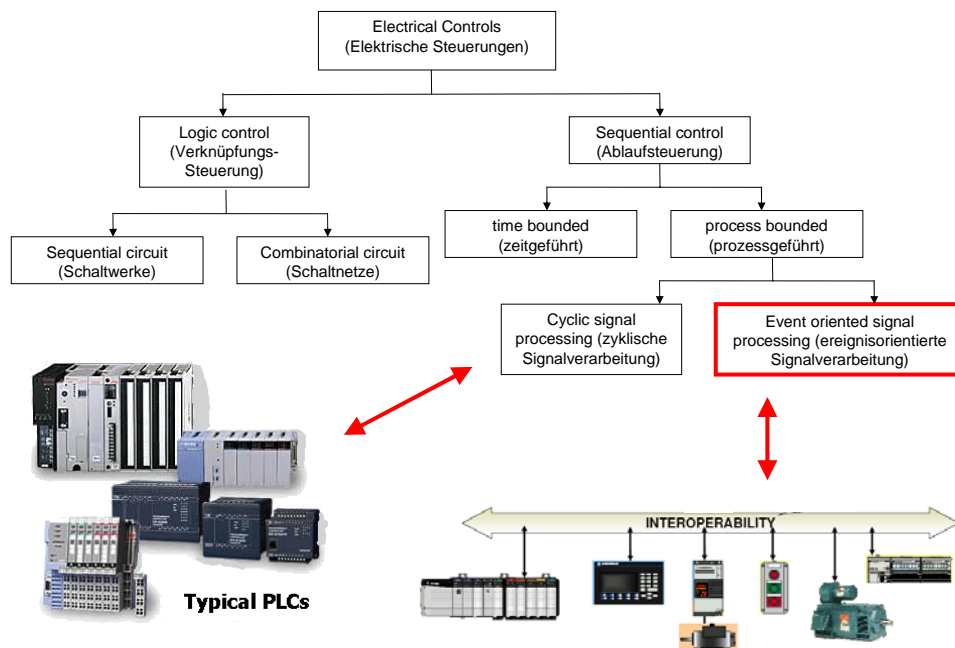
## Standardized Interfaces enabling distributed intelligence



## New Standard: IEC 61499

- Initial work on the IEC 61499 started in 1992
- Official standard since 2005
- Reference model for distributed automation
  - No inherent hierarchy for field devices (“Gleichberechtigung”)
  - The control logic is distributed and executed on the field devices
  - No central PLC “on the top”
- Extending function blocks as known from:
  - IEC 61131-3 Function Blocks (PLC)
  - IEC 61804 Function Blocks (Distributed Control Systems)
- Object (component/module/device) oriented
  - Data encapsulation/interfaces & reusability..
- Event oriented signal processing
  - As classified in the following slide..

## Overview of control types

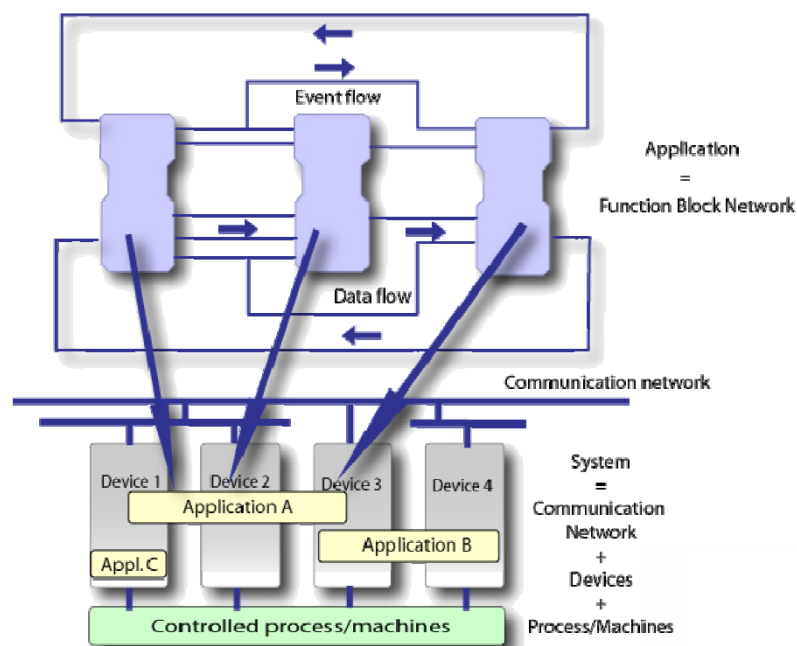


## Object oriented aspects in IEC61449 function blocks

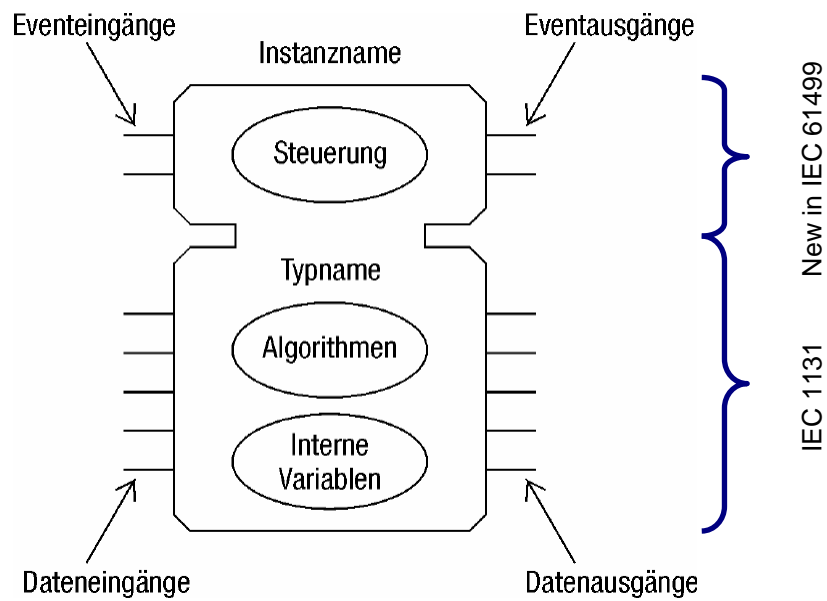
Feature	Objects	Function Blocks	Comment
<b>Encapsulated data</b> (Kapselung)	✓	✓	Objects may contain data that is also instances of other objects. Function blocks may contain instances of other function blocks.
<b>External interface</b>	✓	✓	In IEC 61499 function block, there is not distinction between private and public interfaces.
<b>Invocation</b>	Objects have methods with arguments and returned values	Function blocks use input and output variables and events.	With function blocks, data can be synchronised with an event.
<b>Inheritance</b> (Vererbung)	✓	✗	Currently in IEC 61499 there is no mechanism for a function block to inherit behaviour.
<b>Polymorphism</b> (Polymorphismus "Vielgestaltigkeit")	✓	✓	IEC 61499 introduces a new 'selector' concept that allows function blocks to share common interfaces.
<b>Instantiated from a class</b>	A object class and function block type are synonymous.	Function block instances are defined from function block type.	

<http://www.searcheng.co.uk/selection/control/Articles/IEC61449/page3.htm>

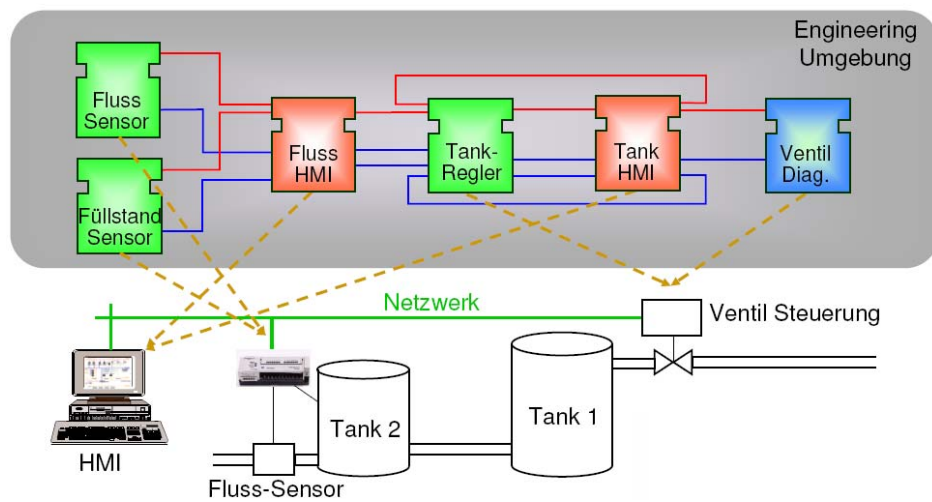
## IEC 61499: concept



## Function block according to IEC 61499

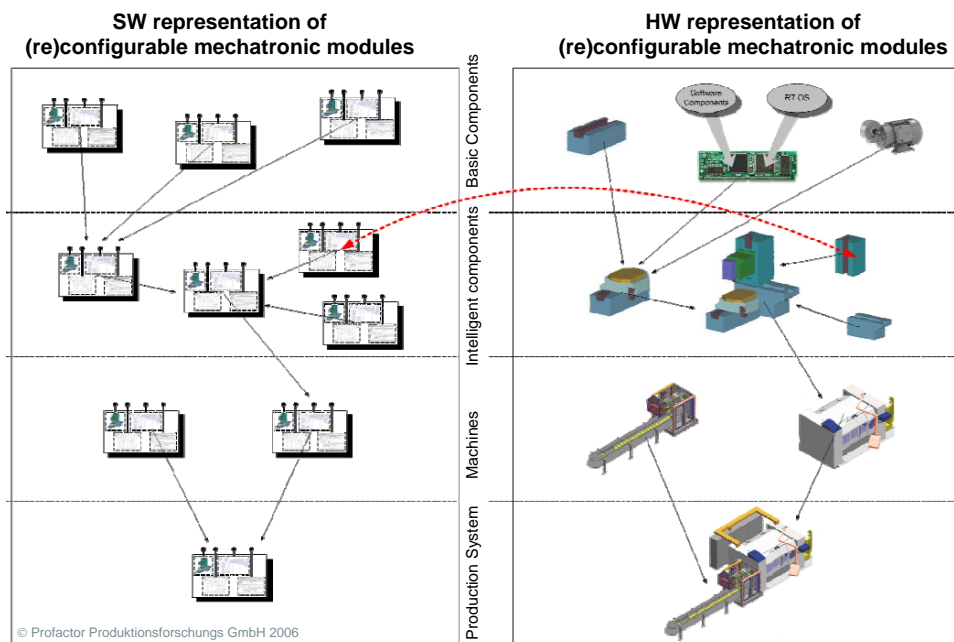


## Simple Example: flow control



— Amount of intelligent components?

## Research topic: (Re)configurable mechatronic modules



## Objectives

- Able to arrange all five IEC-1131 languages and STD (State Transition Diagram) according to the software development process
- Classification of PLC programming language properties like graphical language, close to machine language and high level language like C.
- Explanation of Basic principles and procedures while using STD as a programming language.
- Difference between a PLC and a Soft-PLC
- Getting a grip of the IEC 61499 (decentralized intelligence)