Technische Hochschule Rosenheim
Technical University of Applied Sciences

Prof. Dr. Florian Künzner
Technical University of Applied Sciences Rosenheim, Computer Science

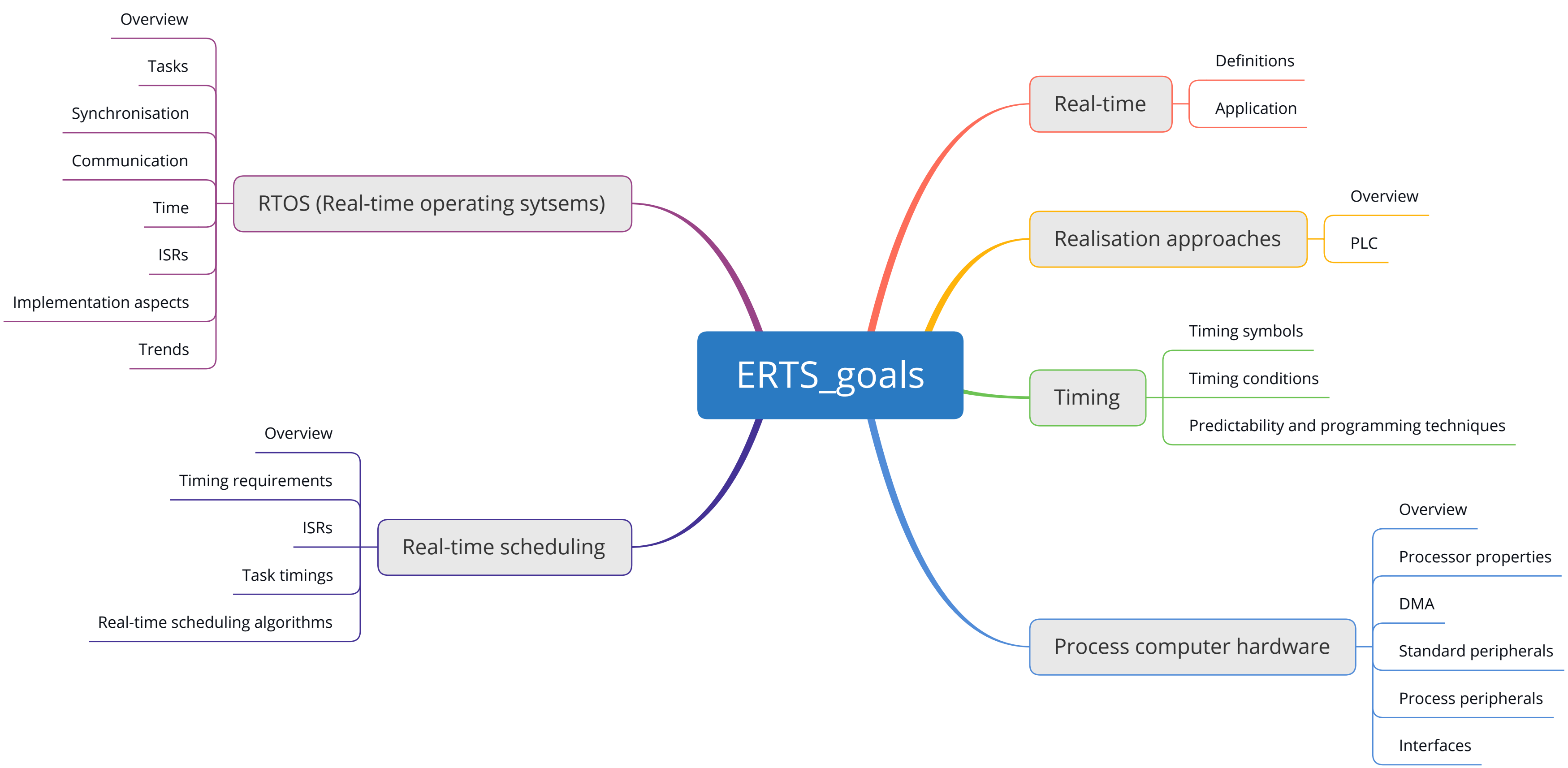
ERTS - Embedded real-time systems

ERTS 2 – Realisation approaches

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal

ERTS::Realisation approaches

- Overview of realisation approaches
- Programmable logic control (PLC)
- Assessment of different approaches

CAMPUS Rosenheim
Computer Science

Overview

How to realise real-time systems?

CAMPUS Rosenheim
Computer Science

Overview

Depending on the complexity (and knowledge), different approaches can be chosen:

- Solution with electromechanics
- Solution with PLC (SPS)
- Solution with simple μC-based software
- Solution with complex software system
- Solution with automation devices

CAMPUS Rosenheim
Computer Science

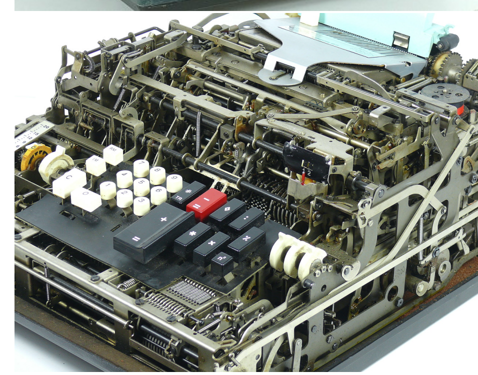
Technische Hochschule Rosenheim
Technical University of Applied Sciences

Solution with electromechanics

Mechanics and simple electrical parts are used to build (simple) solutions.

- For (simple) real-time system solutions
- Electrical parts: relais, resistors, inductors, transistors, ...
- Inflexible: change of logic $\rightarrow$ change of hardware
- Domain of the engineers (mechanical & electrical engineering)

Olivetti Logos 27-2



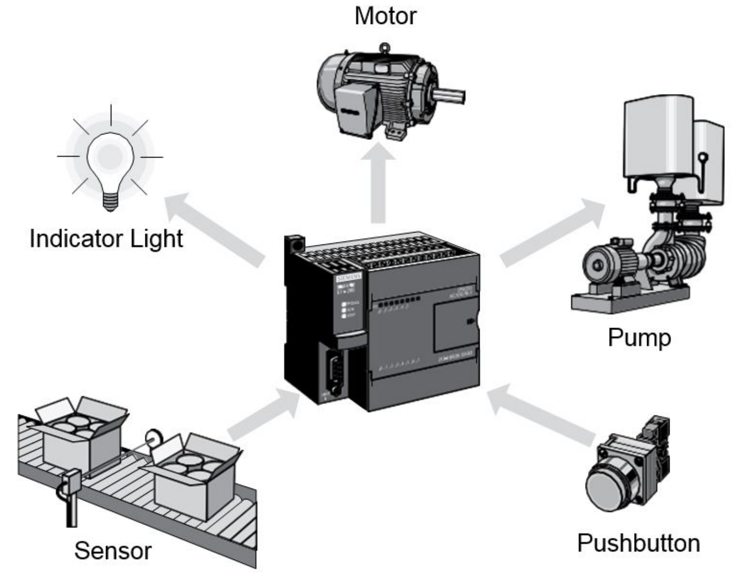
[source: technikum29.de]

CAMPUS Rosenheim
Computer Science

Solution with PLC (SPS)

Programmable logic controls (PLC) are **industrial digital computers**, and are a digital realisation of simple (relais/control) electrical cabinets.

- German: Speicherprogrammierbare Steuerung (SPS)
- Logic: input $\rightarrow$ processing $\rightarrow$ output
- Controls: lights, motors, pumps, ...
- Inputs: buttons, sensors, ...
- Domain of electricians and electrical engineers



[source: isd-soft.com]

$\Rightarrow$ More details in a separate chapter.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Solution with simple μC-based software

A microcontroller (bare metal) with a simple cyclic control loop.

- Easy to implement and understand
- On arbitrarily high operating speed $\rightarrow$ all time conditions can always be met.
- Used in very high safety critical systems: allows easy verification
- But: difficult to implement parallel logic

Logic:

```
do {
      read_inputs();
      process();
      write_outputs();
} while(true)
```

For complex, parallel logic:

- Introducing major/minor cycles
- major: every nth minor cycle
- minor: every (small) cycle
- May be very hard to implement or maintain

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Solution with complex software system

A process control computer or microcontroller with an operating system and a high level programming language.

- Usually uses a real-time operating system (RTOS)
- Appropriate parallelism techniques: tasks/processes
- Synchronisation and communication mechanisms
- Allows complex, parallel solutions
- More advanced (hard) to verify

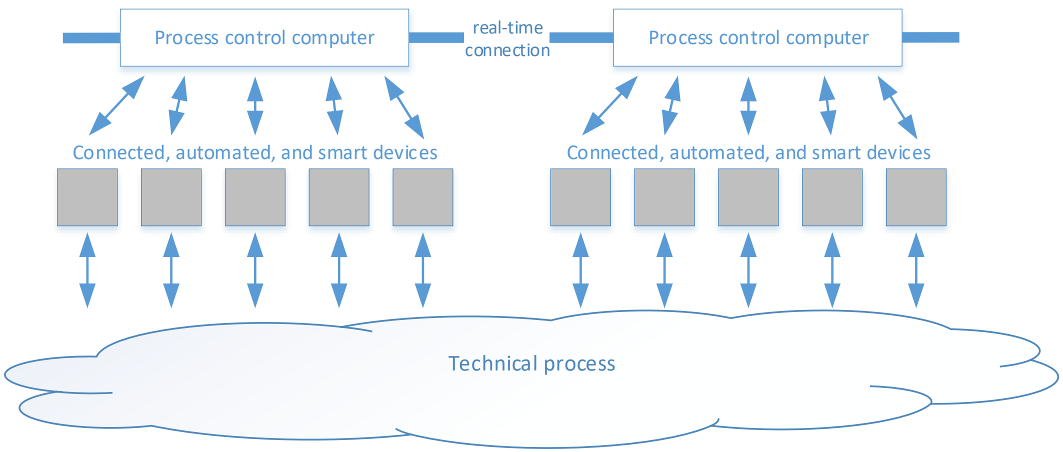
[source: researchgate.net]

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Solution with automation devices

A process control computer with multiple connected, automated, and smart devices.



Connections:

- Digital or analog input/output
- Serial ports: RS232, RS485, ...
- Fieldbus: PROFIBUS, PROFINET, ...
- Usual networks (real-time ethernet)

Examples for smart, automated devices:

- Measurement devices (e.g. scale)
- Robot controls
- Computerised numerical controls (CNC)
- PLCs

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC

PLC: Siemens S7-300, a modular hardware PLC with many I/Os.

[source: sps4you.de]

Structure:

- Power supply
- CPU module
- Digital and analog I/O modules
- Extension modules
- Additional programmable ext. modules

Extension modules:

- Digital and analog inputs/outputs
- Counter and positioning modules
- Stepdrive (stepper motors)
- Closed-loop control module
- Connection: RS232, RS422, RS485, PROFIBUS, Ethernet, ...

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC: internal operation

Internal mode of operation:

```
loop {
    read_inputs();
    process();
    write_outputs();
}
```

Limiting factors:

- Cylce time
- Maximum inputs/outputs

CAMPUS Rosenheim
Computer Science

PLC: types
Hardware vs software PLC

Hardware PLC (as discussed):

- CPU module
- I/O plug-in cards

Software PLC (Soft PLC)

- Process control computer
- "Normal" or real-time operating system
- PLC is a process
- I/Os very individual (depending on system)
- e.g. Raspberry Pi

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC: internetworking

Internetworking of the process control computer with the PLC.

Transfer of *n* data words from/to PLC	$\Rightarrow$	no problems
Start program from network (digital I/O)	$\Rightarrow$	mostly no problems
Start program with parameters	$\Rightarrow$	often not available
Start, stop of PLC over network	$\Rightarrow$	sometimes problematic
Network semaphores	$\Rightarrow$	often not available

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC prog

The IEC 61131-3 describes 5 different programming languages for PLCs.

IL	Instruction list	Comparable to assembler
LD	Ladder diagram	Comparable to an electric circuit diagram (90 degrees rotated)
FBD	Function block diagram	Graphical language with logic blocks (often with programmable sub routines possible)
SFC	Sequential function chart	Graphical language for sequential flow charts
ST	Structured text	Programming language similar to PASCAL

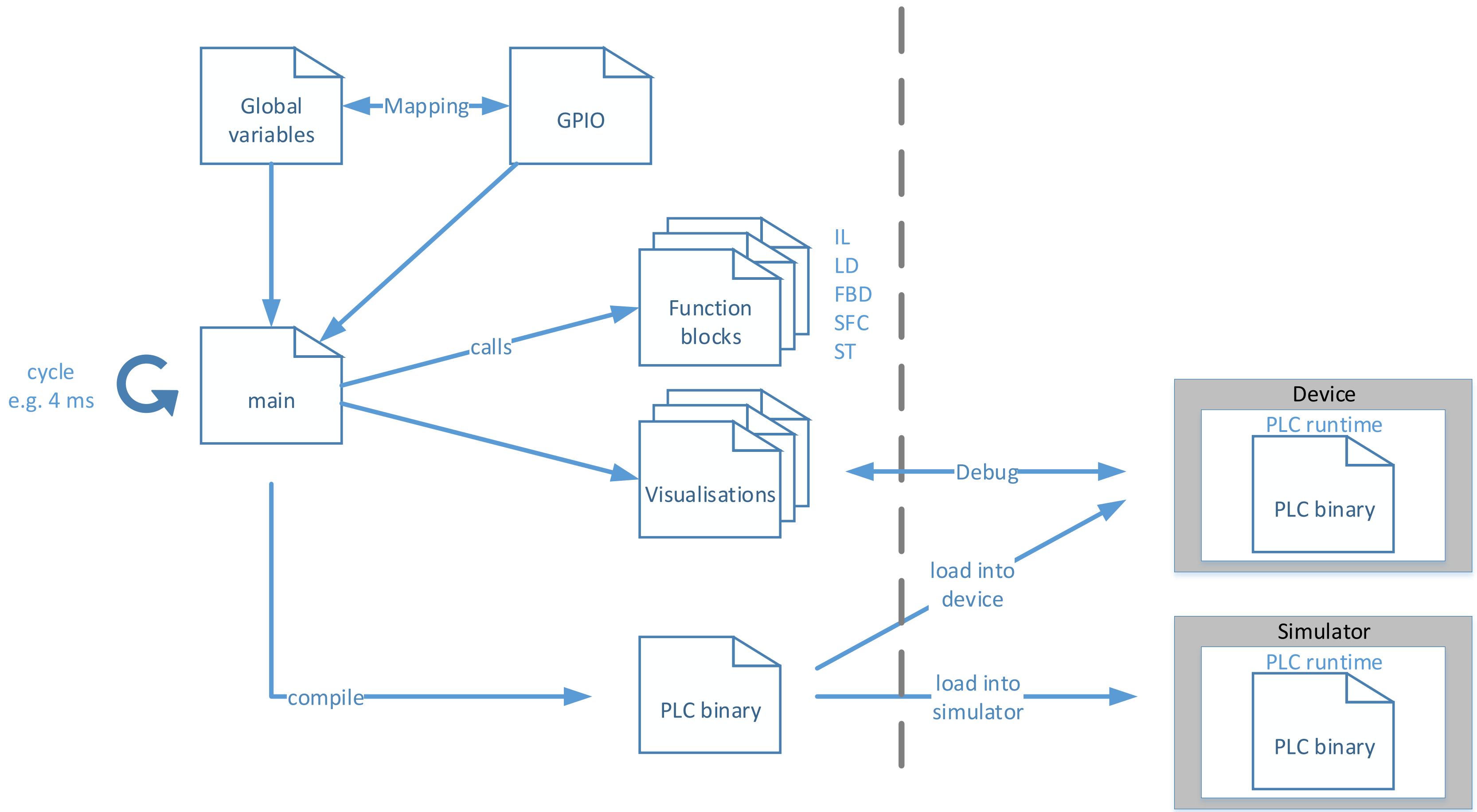
Books:

- SPS-Programmierung mit IEC 61131-3
- Programmable Logic Controllers: A Practical Approach TO IEC 61131-3 Using CoDeSys

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC overview



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC function blocks and data types

Function block definitions have inputs, outputs, and local variables.

Example:*

```
FUNCTION_BLOCK PLC_FBD
VAR_INPUT
    E_5_1: BOOL;
    E_5_2: BOOL;
    E_5_3: BOOL;
    E_5_4: BOOL;
END_VAR
VAR_OUTPUT
    A_1_0: BOOL;
END_VAR
VAR
    M_1017: BOOL;
END_VAR
```

Data types:*

- BOOL (1 bit) (TRUE/FALSE)
- BYTE (8 bits)
- WORD (16 bits)
- DWORD (32 bits)
- INT (16 bits)
- DINT (32 bits)
- UNT (16 bits)/UDINT (32 bits)
- ...
- STRING, ARRAY, TIME, REAL, Structure
- ...

*Based on CODESYS (different PLCs can have different types)

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC: IL – Instruction list

Example:

LD	E_5_1
AND	E_5_2
ORN	E_5_3
S	M_1017
LD	E_5_4
R	M_1017
LD	M_1017
ST	A_1_0

Symbols:

A	=	Accumulator
N	=	negated
operand	=	input/output/memory/...
(	=	grouping
C	=	conditional execution if (A == 1)

Operators:

`LD, LDN`	Load operand into A
`AND, ANDN, AND(, ANDN(`	AND with given operand and A
`OR, OR, OR(, OR(`	OR with given operand and A
`XOR, XOR, XOR(, XOR(`	XOR with given operand and A
`NOT`	Bitwise boolean negation
`ST, STN`	Store A into operand (output/memory)
`S`	Set operand to true if A == 1
`R`	Set operand to false if A == 0
`)`	Closes group

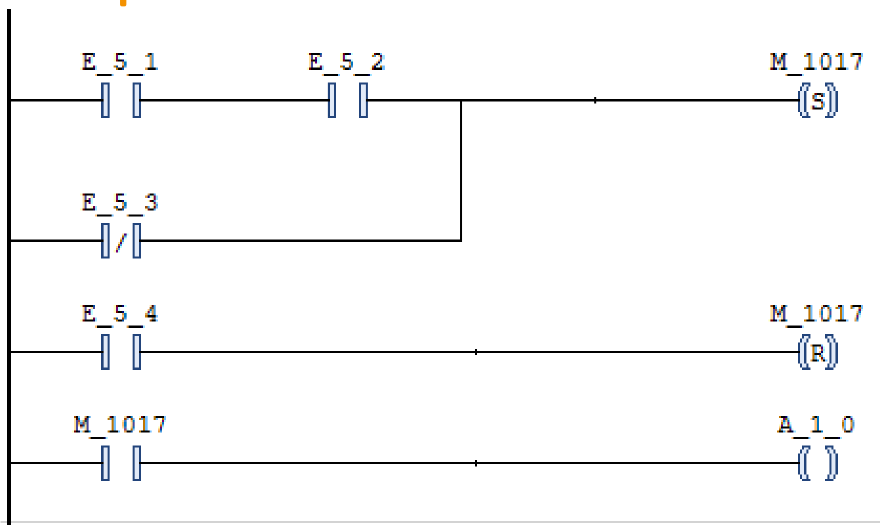
`ADD, SUB, MUL, DIV, MOD, GT, GE, EQ, NE,` and many more ...

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC: LD – Ladder diagram

Example:



Symbols:

Designation		Symbol	Function
Test element	Normally open contact	—| |—	The contact closes when the associated Boolean object becomes TRUE
	Normally closed contact	—|/|—	The contact closes when the associated Boolean object becomes FALSE
	Flank-detecting contacts	—|P|—	Rising edge: The contact is closed only in the scan (see Section 1.3.3) during which the associated Boolean object changes state from 0 to 1
		—|N|—	Falling edge: The contact is closed in the scan where the associated Boolean object changes state from 1 to 0
Connections	Horizontal	—	To connect elements in series
	Vertical	|	To connect elements in parallel
Action element	Direct coil	—( )—	The associated Boolean object is set to the same state as the state of the left side of the coil
	Inverse coil	—(/)—	The associated Boolean object gets the inverse of the state of the left side of the coil
	On-coil	—(S)—	The associated Boolean object is set to TRUE when the state of the left side of the coil is TRUE
	Off-coil	—(R)—	The associated Boolean object is set to FALSE when the state of the left side of the coil is TRUE
	Conditional jump to another rung	→ Label	Enables jump to another named rung in the program (the POU) When a jump is activated: 1. The active rung is interrupted 2. The named rung is activated
	Return to call	**< RETURN >**	If a function or function block is programmed in LD, this is used to return to the POU that called up the function or block[a]

[a] RETURN is implicit at the end of the function or function block.

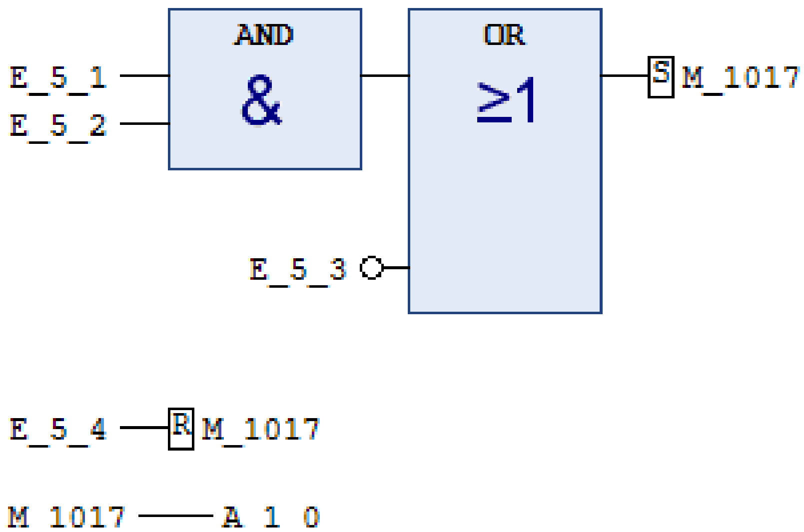
source: [3, Hanssen, p. 228]

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC: FBD – Function block diagram

Example:



Symbols:

Function	Symbol	Description
AND	&	
OR	≥ 1	
XOR		
ADD	+	
MUL	*	
GT	>	
RS or SR		Set-reset

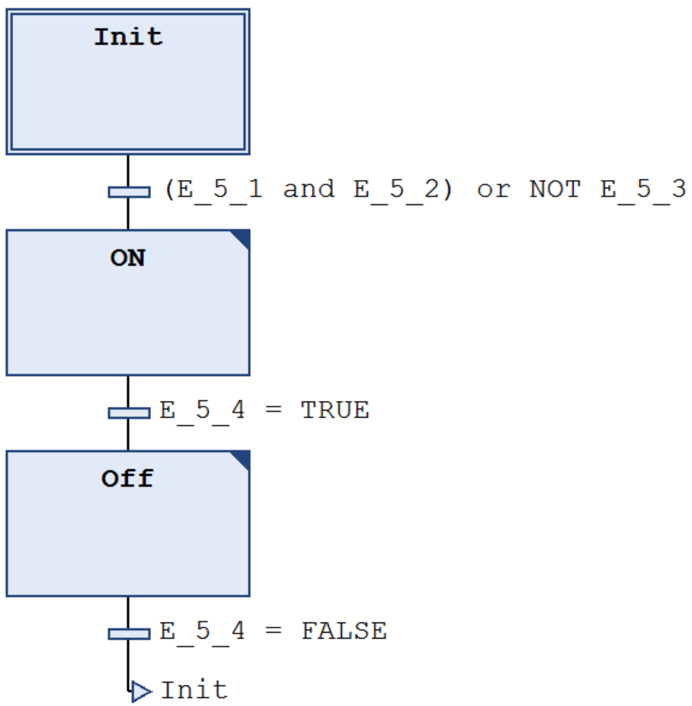
...and many more...

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

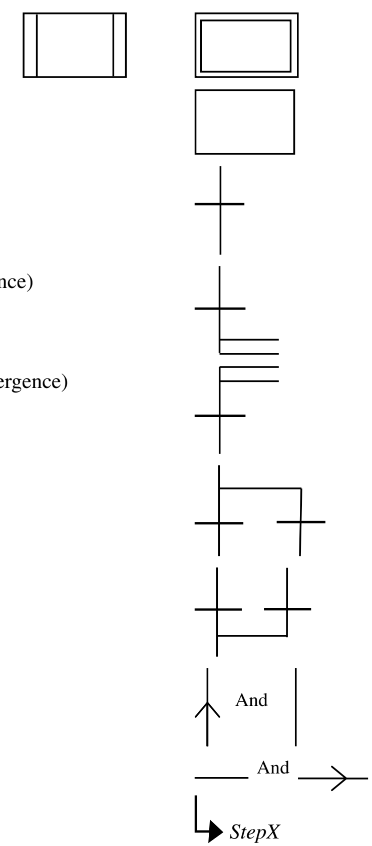
PLC: SFC – Sequential function chart

Example:



Symbols:

Steps	Initiation step (one of these)	
	Ordinary step	
Transitions	Between two steps	
Parallel sequences	Parallel branching (AND divergence)	
	Parallel convergence (AND convergence)	
Alternative sequences	OR divergence	
	OR convergence	
Connecting lines	Up and down	
	Left and right	
Jumps	Jump to StepX	



source: [3, Hanssen, p. 310]

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

PLC: ST – Structured text

Example:

```
IF E_5_1 AND E_5_2 OR NOT E_5_3 THEN
    M_1017 S= TRUE;
END_IF

IF E_5_4 THEN
    M_1017 R= TRUE;
END_IF

A_1_0 := M_1017;
```

Key words:

Schlüssel-wort	Beschreibung	Beispiel	Erklärung
:=	Zuweisung	d := 10;	Zuweisung des rechts berechneten Wertes an den links stehenden Bezeichner
	Aufruf und Gebrauch von FBs [a]	**FBName**(Par1:=10, Par2:=20, Par3=>Erg);	Aufruf einer anderen POE vom Typ FB, sowie deren Parameterversorgung. „:=“ für Eingangs-, „=>“ für Ausgangsparameter
RETURN	Rücksprung	**RETURN**;	Rückkehr in die aufrufende POE
IF	Verzweigung	**IF** d < e **THEN** f:=1; **ELSIF** d=e **THEN** f:=2; **ELSE** f:= 3; **END_IF**;	Auswahl von Alternativen durch boolesche Ausdrücke
CASE	Multiauswahl	**CASE** f **OF** 1: g:=11; 2: g:=12; **ELSE** g:=FunName(); **END_CASE**;	Auswahl eines Anweisungs-blocks, abhängig vom Wert des Ausdrucks f

FOR	Wiederholungs-anweisung (1)	**FOR** h:=1 **TO** 10 **BY** 2 **DO**
		f[h/2] := h;
		END_FOR;
			Mehrfachdurchlauf eines Anweisungsblocks mit Start- und Endebedingung

WHILE	Wiederholungs-anweisung (2)	**WHILE** m > 1 **DO**
		n := n / 2;
		END_WHILE;
			Mehrfachdurchlauf eines Anweisungsblocks mit Endebedingung

REPEAT	Wiederholungs-anweisung (3)	**REPEAT**
		i := i*j;
		UNTIL i > 10000
		END_REPEAT;
			Mehrfachdurchlauf eines Anweisungsblocks mit nachgeschalteter Ende-bedingung

EXIT	Schleifen-Abbruch	**EXIT**;	Abbruch einer Wieder-holungsanweisung
;	Leeranweisung	; ;	

source: [4, John, p. 120]

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Assessment of different approaches

Principle

Solve simple problems with simple solutions.

Depending on:

- Real-time requirements
- Number of inputs/outputs
- Connections: Software protocols/hardware wires
- Required logic and mathematical calculations
- Also considered: Size, budget, criticality, resistance, modularity
- In safety critical applications, synchronous programming is also preferred

But: Do not try to solve complex problems with simple solutions:

- Often confusing solutions
- Inflexible
- Hard to maintain

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Assement of different approaches

Example	Solution with ... electromechanics	PLC (SPS)	simple μC-based SW	complex software system	automation devices
Rearview of a car					
Electric shaver					
Moving staircase (escalator)					
Elevator					
Traffic light					
Aeroplane control					
Vehicle control system					
Computerized numerical control					

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Summary and outlook

Summary

- Overview of realisation approaches
- Programmable logic control (PLC)
- Assessment of different approaches

Outlook

- Real-time basic mechanism and timing aspects