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 SIMATIC 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.


IEC 1131 programming languages classification

Languages of Programmable Logical Controllers (IEC 1131-1/3)

- Instruction List (IL)
  - U E 5.3
  - U E 2.6
  - Q A 2.1
  - ON M 23.1
  - A 2.4
- Structured Text (ST)
  - IF A=0 THEN ...
  - ELSE ...
- Function Block Diagram (FBD)
  - E3
  - E2
  - A2
  - N3 = C
- Ladder Diagram (LD)
  - M2 = 1
  - A2 = 1
  - N3 = C
- Sequential Function Chart (SFC)
  - S1
  - S2
  - S3
  - S4

IAS Universität Stuttgart, 2003
Programming language level

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

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 where used to control the digital machine events in the past
- PLC’s where invented to increase the control power
LD, a simple program based on the former relay example

Sensor

di_switch

Actor
do_bell

End

A coil symbol

A contact symbol

http://www.plcs.net/chapters/replace8.htm

LD, logic basics

AND

PB1 PB2 R1

PB3 PB4

R1 = PB1.AND.PB2

R2 = PB2.AND.¬PB4

make contact (Schliesserkontakt) break contact (Öffnerkontakt)

OR

PB1 PB2 R1

R1 = PB1.OR.PB2
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, 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

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)

- 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
- E0.0 AND E0.1 are set
- OR E0.2 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 (Strukturiertes 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, ..)
The outputs (Ausgang) $A_1$ and $A_2$ are only set, if either the input (Eingang) $E_3$ is set or $E_1$ and $E_2$ are set simultaneously.
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

- Equalities/Differences?
- Actors/Sensors?
Summarizing all discussed languages

<table>
<thead>
<tr>
<th>Low level</th>
<th>High level</th>
<th>Textual</th>
<th>Graphical</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>IU</td>
<td>FBD</td>
<td>ST</td>
</tr>
<tr>
<td>IL</td>
<td></td>
<td>SFC</td>
<td>STD</td>
</tr>
<tr>
<td>Low level</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>High level</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Textual</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Graphical</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

SIEMENS: LAD/KOP AWL FUP SCL Graph HiGraph

German: KOP AWL FBS ST AS Zustandsdiagram

level of language

high

low

Soft PLC versus PLC
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.

Machine controlled by a PLC

Control

Actuators

Basic system

Sensors
Machine controlled by a Soft-PLC

SIEMENS Soft-PLC: PLCSim

Programming Environment:
- defining the I/O’s
- programming
- compiling

Soft-PLC:
Control; processing the signals
Trends in automation

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

![Diagram showing the hierarchy of control types, including Electrical Controls, Logic control, Sequential control, Cyclic signal processing, and Event oriented signal processing.](image-url)
### Object oriented aspects in IEC61449 function blocks

<table>
<thead>
<tr>
<th>Feature</th>
<th>Objects</th>
<th>Function Blocks</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kapselung (Kapselung)</td>
<td>✔️</td>
<td>✔️</td>
<td>Objects may contain data that is also instances of other objects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Function blocks may contain instances of other function blocks.</td>
</tr>
<tr>
<td>External Interface</td>
<td>✔️</td>
<td>✔️</td>
<td>In IEC 61499 function block, there is no distinction between private and public interfaces.</td>
</tr>
<tr>
<td>Invocation</td>
<td>Objects have methods with arguments and returned values</td>
<td>Function blocks use input and output variables and events.</td>
<td>With function blocks, data can be synchronized with an event.</td>
</tr>
<tr>
<td>(Vererbung)</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polymorphismus</td>
<td>✔️</td>
<td>✔️</td>
<td>IEC 61499 introduces a new 'inherited' concept that allows function blocks to share common interfaces.</td>
</tr>
<tr>
<td>(Polymorphismus &quot;Vielgestaltigkeit&quot;)</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inheritance from a class</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

### IEC 61499: concept

![IEC 61499 concept diagram](image-url)
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)