PLCs: Programmable Logic Controllers

Hipólito Guzmán Miranda
Departamento de Ingeniería Electrónica
Universidad de Sevilla
hguzman@us.es

What are PLCs

- PLC is the acronym for “Programmable Logic Controller”
- So, like an FPGA? -> not so fast!

- The IEC 61131-3 standard defines the basic software architecture and programming languages for PLC’s

Logic, but not FPGA

- Think of them as ‘ruggedized computers’.
- But not as ‘computers for which I can program whatever I want and make them hang, crash, go into infinite loops…’.
- How to program them is key!
- Some restrictions when programming them (you can’t do everything), but in return, more reliable software.

Origin

- Industrial control and automation systems of the 1960’s had a plethora of problems
- They were implemented with relays
- Troubleshooting when a component failed was hugely problematic and slow
- The idea of the PLC is to allow for more reliable computation for industrial control/automation

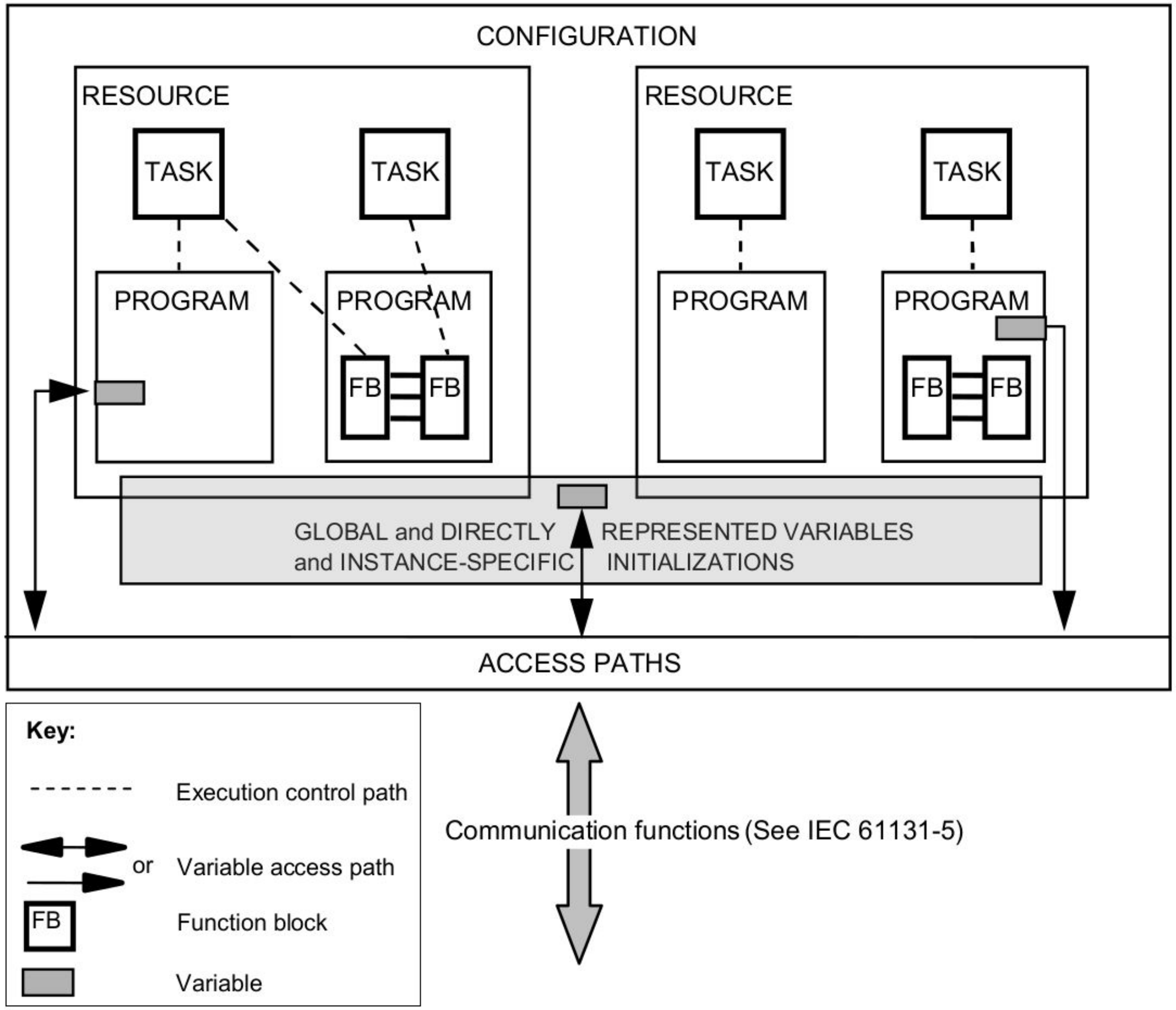
Software architecture

- Program organization unit
  - *“function, function block, class, or program”*
- Resource
  - *“language element corresponding to a “signal processing function” and its “man-machine interface” and “sensor and actuator interface functions”, if any”*
- Task
  - *“execution control element providing for periodic or triggered execution of a group of associated program organization units”*

Software architecture

- Function
    - *“language element which, when executed, typically yields one data element result and possibly additional output variables”*
- Function block type
    - *“language element consisting of the definition of a data structure partitioned into input, output, and internal variables; and a set of operations or a set of methods to be performed upon the elements of the data structure when an instance of the function block type is called”*
- Function block instance
    - *“instance of a function block type”*
- Function block diagram
    - *“network in which the nodes are function block instances, graphically represented functions or method calls, variables, literals, and labels”*

Software architecture



Source:

IEC 61131-3

Standard

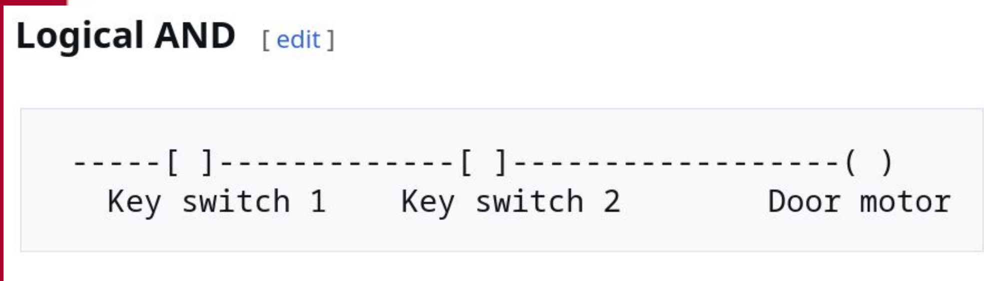
How to program them?

- Ladder diagram (LD)
- Function Block Diagram (FBD)
- Structured text (ST)
- Sequential function chart (SFC)

Defined in the IEC 61131-3 standard

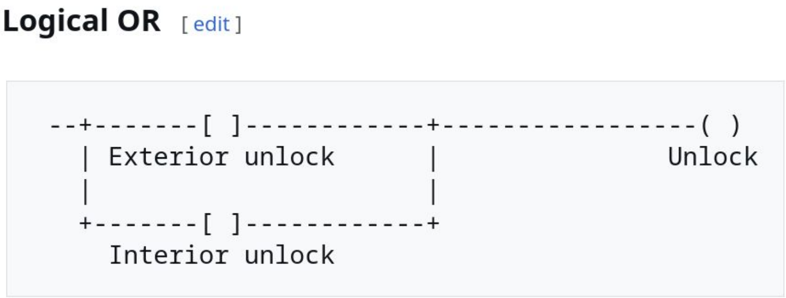
Ladder Diagram (LD)

- Graphical
- Use for simple but critical circuits, or to reimplement relay-based logic



Logical AND [edit]

```
-----[ ]-------------[ ]------------------( )
  Key switch 1    Key switch 2        Door motor
```



Logical OR [edit]

```
--+-------[ ]------------+-----------------( )
  | Exterior unlock      |                 Unlock
  |                      |
  +-------[ ]------------+
    Interior unlock
```

Image source: Wikipedia

Function Block Diagram (FBD)

- Graphical
- Parallel by nature
- Use for more complex logic than LD

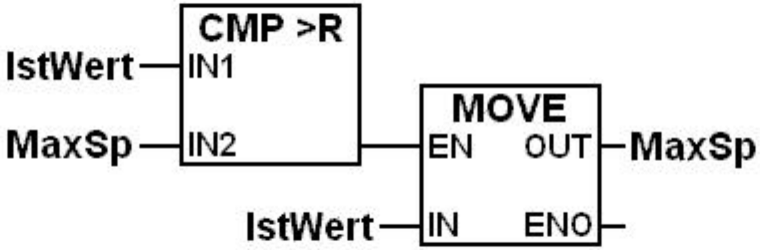
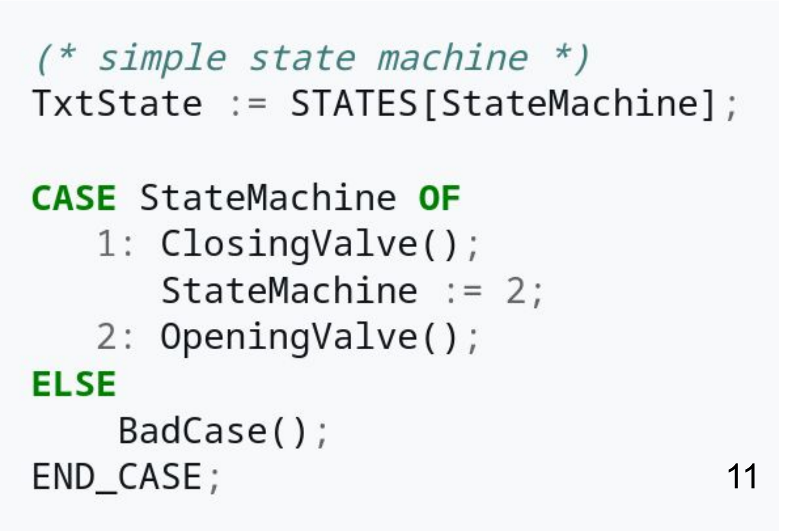


Image source: Wikipedia

Structured Text (ST)

- Text-based
- Code is sequential, but continuously run in parallel with everything else
- Use when FDB would yield very complex circuits (such as calculations with a lot of mathematical operations)



```
(* simple state machine *)
TxtState := STATES[StateMachine];

CASE StateMachine OF
   1: ClosingValve();
      StateMachine := 2;
   2: OpeningValve();
ELSE
    BadCase();
END_CASE;
```

Code snippet source: Wikipedia

Sequential Function Chart (SFC)

- Graphical
- Lets you organize a program with both sequential and parallel processing
- Can insert LD / FDB / ST/ SFC blocks inside
- Use for high-level organization of your program

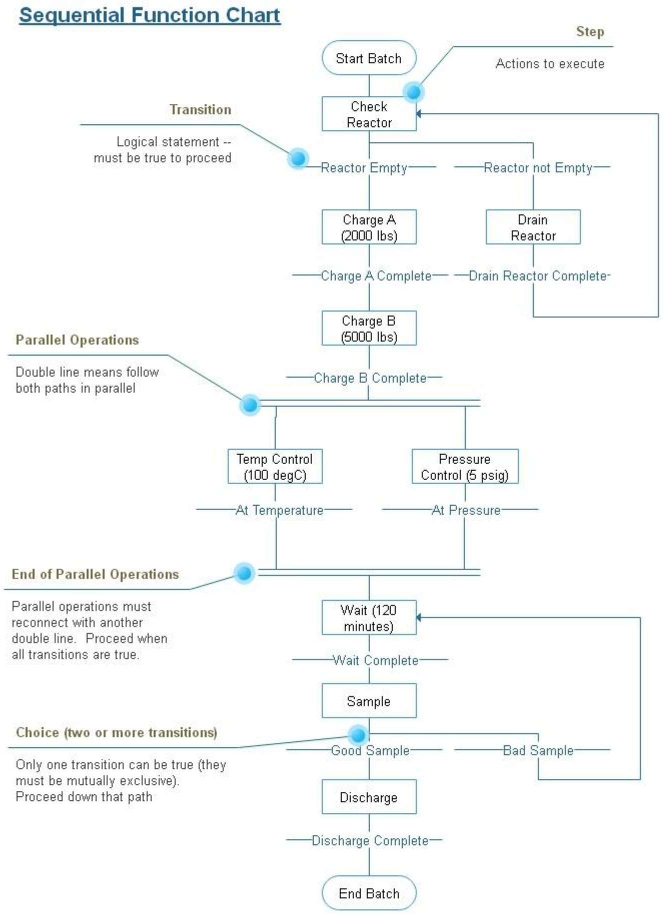

Sequential Function Chart
Step
Actions to execute
Start Batch
Check Reactor
Transition
Logical statement -- must be true to proceed
Reactor Empty
Reactor not Empty
Charge A (2000 lbs)
Drain Reactor
Charge A Complete
Drain Reactor Complete
Charge B (5000 lbs)
Parallel Operations
Double line means follow both paths in parallel
Charge B Complete
Temp Control (100 degC)
Pressure Control (5 psig)
At Temperature
At Pressure
End of Parallel Operations
Parallel operations must reconnect with another double line. Proceed when all transitions are true.
Wait (120 minutes)
Wait Complete
Sample
Choice (two or more transitions)
Good Sample
Bad Sample
Only one transition can be true (they must be mutually exclusive). Proceed down that path
Discharge
Discharge Complete
End Batch

Image source: Wikipedia

Conclusions

- PLCs are ruggedized computers with very specific programming languages
  - Avoid shooting yourself on the foot
  - Of course you will find limitations
  - The tradeoff is reliability
- When to use PLCs?
  - When reliability is key
  - When you don’t need really huge performance
  - In industries where its use is already established
- In principle, many things that they do can already be done using FPGAs, with actual parallelism
  - But of course there may be verification and reliability concerns

Bibliography

- A. Theorin, V. Romero Segovia, *“A Short History About PLCs and DCS”*
- International Electrotechnical Commision (IEC), *“Programmable controllers: EN 61131-3 : European standar”d*
- J. Karl-Heinz, *“IEC 61131-3 : programming industrial automation systems”*
- Wikipedia, *“Ladder Logic”*
- Wikipedia, “Function Block Diagram”
- Wikipedia, *“Structured Text”*
- Wikipedia, “Sequential Function Chart”