Graphics editors in CPDev environment
Agenda

- CPDev engineering environment and its applications
- Visual programming and motivation to create graphics editors in CPDev
- Editors for graphical languages
- Selected editors features:
  - automatic connection finding,
  - diagram translation to ST code,
  - conversion to XML format,
  - program execution.
CPDev engineering environment

- Integrated development environment (IDE) for programming PLCs, PACs, softPLCs and distributed control systems
- Project has been being developed in the Division of Informatics and Control (at Rzeszow University of Technology)
- Based on virtual machine
- Support for IEC 61131-3 languages:
  - textual: ST, IL,
  - graphical: FBD, LD,
  - mixed: SFC.

Selected applications of CPDev:

- by **Praxis Automation Technology** (Leiderdorp, NL) in ship control and monitoring system,
- by **Lumel S.A.** (Zielona Góra, PL),
- by **Nauka i Technika Sp. z o.o.** (Rzeszów, PL),
- in **softPLC** software,
- in multiprocessor **FPGA** controller.
Visual programming

- **Cooperation with industry** indicates a necessity of further CPDev development, including adding support for graphical languages.

- **Visual programming** has many advantages, e.g.:
  - diagrams can be more legible,
  - simplification of understanding and making changes, especially by people who do not have enough skills of programming in textual languages,
  - can require less work to create a program,
  - possibility of attaching printouts to the documentation.
Graphics editors

- **Assumptions** for graphics editors:
  - strict integration with CPDev environment and its modules (including CPSim),
  - fast and convenient way of programs creation, e.g. by automatic connections finding,
  - execution mode supporting many data sources, tracing values of variables, and breakpoints,
  - usage of existing ST compiler,
  - usage of XML format and *PLCopen* standard,
  - possibility of running programs on all platforms supported by CPDev
IEC 61131-3 language support

- FBD editor
  - DELAY_ON TON IN Q PT ET
  - DELAY_OFF TOF IN Q PT ET
  - PUMP
  - START
  - FBD: CLEANING (E)
  - ST: CALIBRATION (I)
  - T_START
  - SYSTEM_READY
  - PRODUCTION
  - IL: PRODUCTION_1 (I)
  - LD: PRODUCTION_2 (I)
  - FBD: PRODUCTION_3 (I)
  - T_PR
  - PARTS_COUNT > 1000
  - ASSEMBLING
  - FBD: ASSEMBLING_1 (E)
  - ST: ASSEMBLING_2 (L)
  - T_AS
  - ASSEMBLED_COUNT > 100
  - END

- LD editor
- SFC editor

Graphics editors – FBD support

- Function Block Diagram
- FBD programs consist of:
  - input and output variables,
  - constants,
  - instances of function blocks,
  - functions

- Elements are connected with lines
- Very convenient for electricians
Graphics editors – LD support

- Ladder Diagram
- LD programs consist of:
  - contacts,
  - coils,
  - constants,
  - power rails (left and right),
  - rungs,
  - optionally – function block instances and functions
- Very convenient for electricians and mechanics
Graphics editors – SFC support

- **Sequential Function Chart**
- **Not an independent language**, i.e. it requires parts prepared in different IEC 61131-3 languages

- Programs consists of:
  - **steps** (with one initial step),
  - **actions**,
  - **transitions**,
  - **sequences** (also simultaneous)
  - **jumps**
Graphics editors

- All editors are equipped with **many features**, e.g.:
  - basic edition options (i.e. adding, copying, pasting),
  - loading and saving diagrams,
  - automatic connections finding,
  - translation to ST code,
  - conversion to XML format based on *PLCopen*,
  - printing accordingly to the printout template,
  - adjusting way of displaying diagrams,
  - basic checking of diagram completeness,
  - execution mode.
Automatic connections finding

- **Finding paths** between graph nodes
- Mechanism assumptions:
  - passing round existing elements on the diagram,
  - limiting number of line intersections,
  - limiting number of direction changes.
- **A* algorithm** adjusted to the problem, e.g. by:
  - dividing grid to smaller areas,
  - setting suitable costs,
  - usage of Manhattan metric: \( h(n) = |x_k - x_n| + |y_k - y_n| \),
    \((x_k, y_k) - \text{end node coordinates}, (x_n, y_n) - \text{n node coordinates})\.

Automatic connections finding

Exemplary diagrams (in FBD and LD languages) with automatically created connections

Cost of the connection calculation
Program compilation

- Programs created in FBD, LD, and SFC languages are **translated** into **ST** code, that is then **compiled** into virtual machine code.

![Diagram of program compilation process.]
Translation to ST code

PROGRAM START_STOP
VAR_EXTERNAL
START: BOOL;
STOP: BOOL;
ALARM: BOOL;
ENGINE: BOOL;
PUMP: BOOL;
END_VAR

VAR
DELAY_ON: TON;
DELAY_OFF: TOF;
out_or1: BOOL;
out_and1: BOOL;
out_DELAY_ON_Q: BOOL;
out_DELAY_OFF_Q: BOOL;
END_VAR

out_or1 := OR(out_and1, START);
out_and1 := AND(out_or1, NOT STOP, NOT ALARM);
DELAY_ON(IN:=out_and1, PT:=T#5s, Q=>out_DELAY_ON_Q);
DELAY_OFF(IN:=out_DELAY_ON_Q, PT:=T#10s, Q=>out_DELAY_OFF_Q);
ENGINE := out_and1;
PUMP := out_DELAY_OFF_Q;
END_PROGRAM
Conversion to XML format

- Structure of XML file based on PLCopen standard (for FBD, LD, and SFC languages)

```xml
<block height="100" instanceName="DELAY_ON" localId="8" typeName="TON" width="80">
    <position x="330" y="220" />
    <inputVariables>
        <variable formalParameter="IN">
            <connectionPointIn>
                <relPosition x="-10" y="50" />
                <connection formalParameter="Return" refLocalId="7">
                    <position x="320" y="270" />
                    <position x="290" y="270" />
                    <position x="290" y="180" />
                    <position x="270" y="180" />
                </connection>
            </connectionPointIn>
        </variable>
        (...)<inputVariables>
    </inputVariables>
    <outputVariables>
        <variable formalParameter="Q">
            <connectionPointOut>
                <relPosition x="90" y="50" />
            </connectionPointOut>
        </variable>
        (...)<outputVariables>
    </outputVariables>
</block>
```
Editors are equipped with an execution mode

Simulation and commissioning

Many data sources are supported, including:
- simulator (local CPDev virtual machine),
- using Modbus protocol,
- supporting multiprocessor FPGA controller.
Debugging features

- Support for **tracing variable values and breakpoints** (both conditional and unconditional) on FBD, LD, and SFC diagrams.
Summary

- CPDev software is equipped with a set of editors that make possible to create program organization units (POUs) in all languages defined in IEC 61131-3 norm (including graphical).

- Graphics editors have a set of features which aim is to simplify and speed up POU creation (including automatic connections finding with the usage of A* algorithm).

- Many problems solved during development (e.g. translation to ST code and conversion to/from XML format based on PLCopen).
Thank you for attention

**Chosen bibliography:**