

# CPDev – FBD programming

---

## CONTENTS

• <b>Introduction</b> .....	<b>1</b>
• <b>START-STOP diagram</b> .....	<b>1</b>
Diagram elements .....	1
• <b>New program</b> .....	<b>2</b>
Project .....	2
Program name, POU type and language .....	2
• <b>FBD graphic editor</b> .....	<b>3</b>
Tree .....	3
Find element .....	4
Editor options .....	5
• <b>Development steps</b> .....	<b>5</b>
Global variables .....	5
Diagram .....	5
Task. Compilation. Simulation .....	6
• <b>Elements of START-STOP</b> .....	<b>6</b>
Placing elements on the diagram .....	6
Input, output and global variables .....	7
Functions .....	7
Functions blocks .....	8
Constants .....	8
Input/output negation .....	8
Change order of elements .....	8
Comments .....	8
• <b>Creating connections</b> .....	<b>9</b>
Connection.....	9
Branch point.....	9
Editing .....	10
Tips on automatic connections .....	10
Verification .....	11
• <b>Task and compilation</b> .....	<b>11</b>
Task .....	11
Compilation – <i>Build</i> .....	11
• <b>Online mode</b> .....	<b>12</b>
Go online .....	12
FBD diagram online .....	12
Value list .....	14
Data sources .....	14
• <b>CPSim simulator</b> .....	<b>15</b>
Global variables .....	15
Control panel .....	15
Individual view .....	16

Variable list .....	16
Sessions .....	16
• <b>User library</b> .....	<b>16</b>
Source project.....	17
Export as a library .....	17
Using the library .....	18
• <b>Other issues</b> .....	<b>19</b>
Changing FBD program name .....	19
Library warnings .....	19
Automatic save .....	19
• <b>Examples</b> .....	<b>20</b>
Start stop - RS .....	20
Protection against frequent restarts .....	20
Start-up horn .....	20
Blinking light and counter .....	21
Single pushbutton .....	21
Burglar alarm .....	21
Low / High alarms .....	22
Time measurement .....	23
Signal filter .....	23
• <b>Printing</b> .....	<b>24</b>
Size adjustment .....	24
Page setup.....	24
Print.....	25

15 February, 2018

Editor version: FBD 0.4.2.1

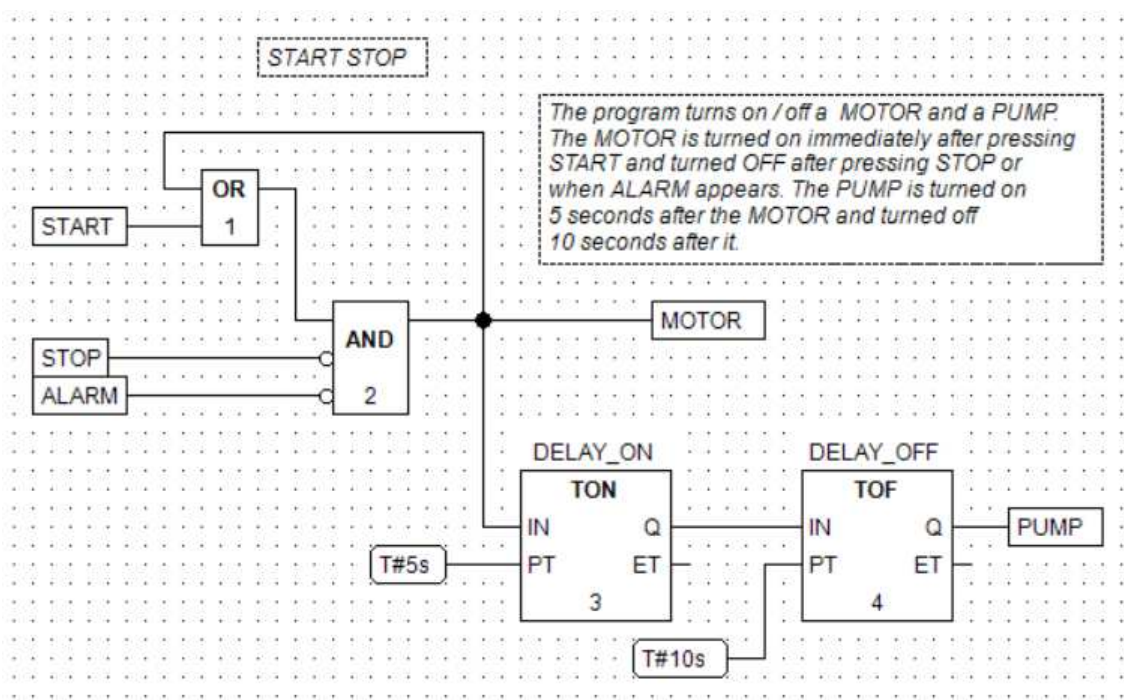
## Introduction

CPDev environment supports programming in graphic FBD language (*Function Block Diagram*) by means of *FBD editor* module (external component). Users can create all POU types (programs, functions and function blocks) and trace values of variables during simulation. In this instruction creating a typical program in FBD language involving input/output variables, functions and function blocks is described in details. Creating user-defined function and function block is also presented. Other examples are also provided, such as common switching circuits, low/high alarms, time measurement and signal filter. It is assumed that the user is familiar with basics of using CPDev environment.

## START-STOP diagram

### Diagram elements

- Final program in FBD language is shown below. It contains:
  - input variables START, STOP, ALARM
  - output variables MOTOR, PUMP
  - functions OR, AND
  - instances DELAY\_ON, DELAY\_OFF of function blocks TON, TOF
  - constants T#5s, T#10s
  - comments.

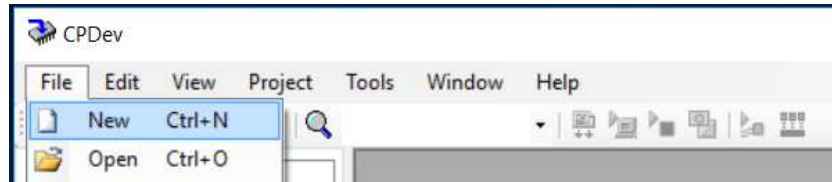


- Execution order of functions and function blocks is specified by numbers 1, 2, 3, 4 written inside rectangles.
- Main part of this instruction describes how to create and simulate such program. Other examples are presented later.

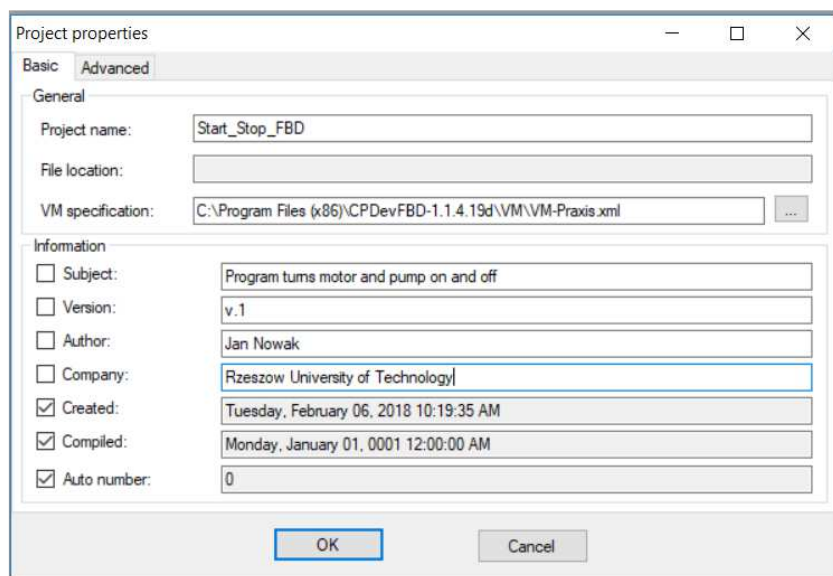
# New program

## Project

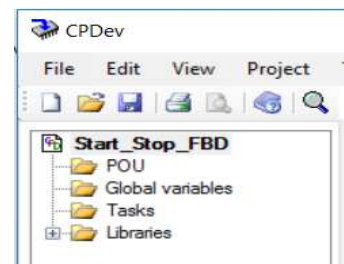
- New project is defined by selecting *File* → *New* in CPDev main menu or clicking *New* icon in the toolbar.



- Enter *Project name*, here *Start\_Stop\_FBD*, and eventually other data in *Project properties* window.



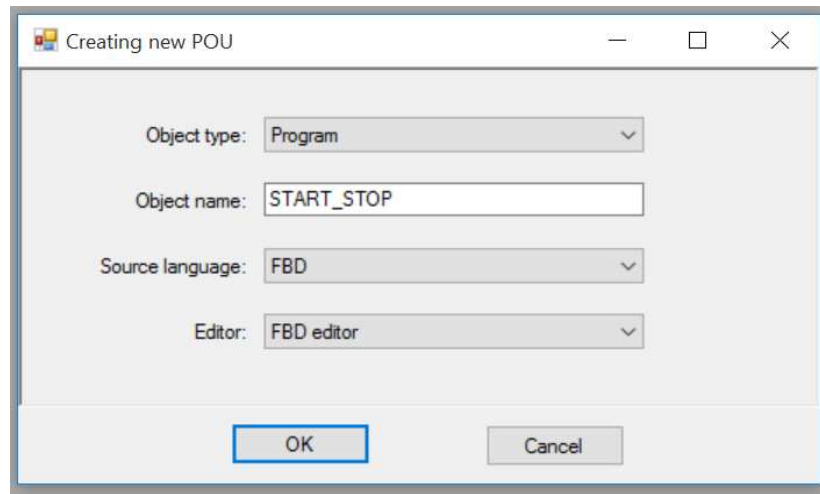
- File location, compilation date and auto number will be filled in automatically while saving the project.  
VM specification indicates path to virtual machine description file.
- Left part of CPDev window presents initial tree of the *Start\_Stop\_FBD* project.



## Program name, POU type and language

- Program can be added to the project in two ways:
  - from project tree:
    - select project name, i.e. *Start\_Stop\_FBD*
    - from context menu, select *Add Item* → *Program* or select *POU* and from its context menu choose *Add* → *Program*
  - from CPDev main menu:
    - select *Project*
    - from drop-down menu, select *Item* → *Add* and select *Program* in *Adding new element* window.

- Enter *Object name*, here START\_STOP, and choose FBD language (default *FBD editor* appears).



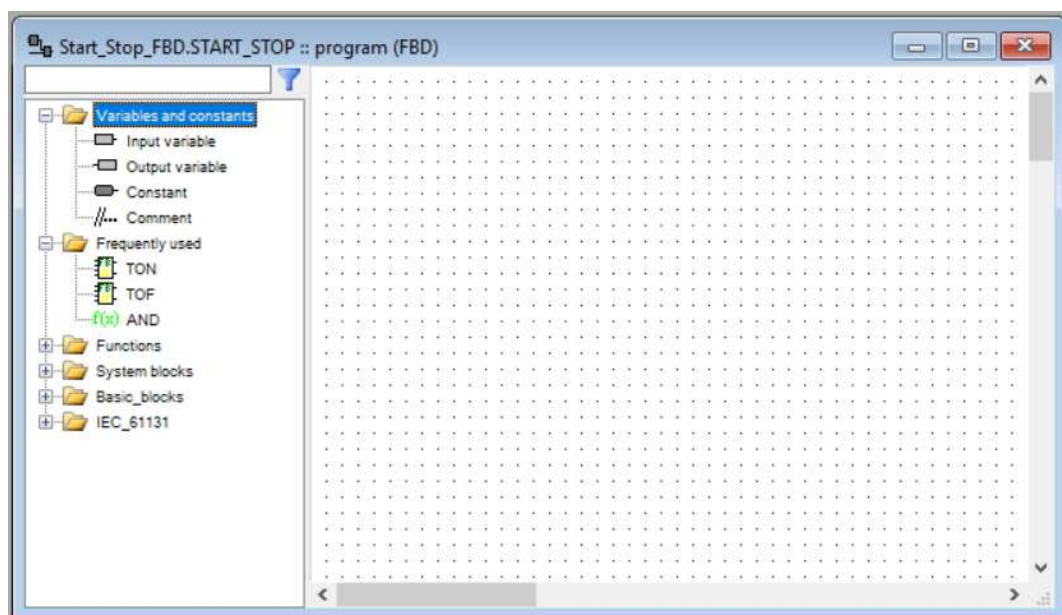
- In this window the user can also select another object type, namely function or function block.
- Project tree involves now START\_STOP program.



## FBD graphic editor

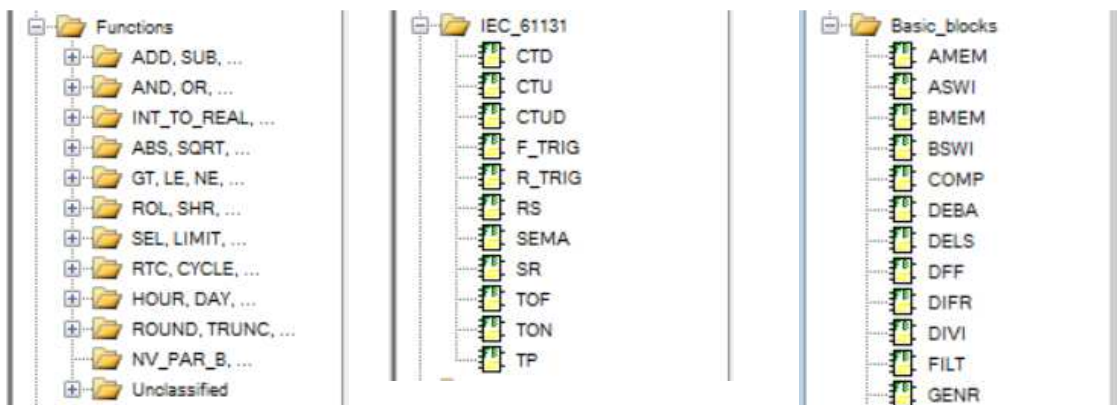
### Tree

- Left part of FBD editor window contains tree whose elements can be dragged or selected and dropped on the drawing board on the right.



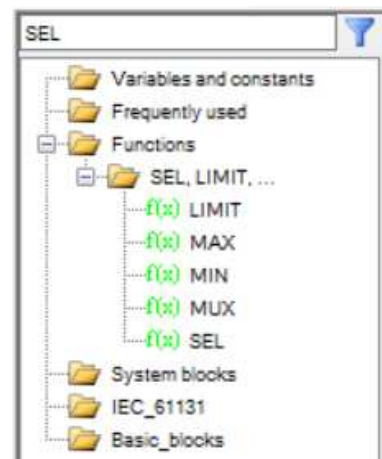
- Grey line farther to the right (move scrollbar or maximize) is the limit of first printed page. Grey lines partition total work area into a few pages (see *Printing*).
- The tree involves:
  - input/output variables, constant and comment
  - a few frequently used blocks and functions
  - *Functions* branch with subbranches, such as arithmetic ADD, SUB, ..., logic AND, OR, ..., conversions INT\_TO\_REAL, ..., etc.
  - hardware dependent *System\_blocks* (see below)
  - blocks of standard *IEC\_61131* library
  - blocks of *Basic\_blocks* library with special arithmetic blocks, switches, memory blocks, comparator, flip-flops, pulsers, filters, and some nonlinearities (see basic CPDev instruction).

*Remark.* Hardware dependent *System\_blocks* such as NV\_PAR\_... store parameters in nonvolatile memory (BOOL, INT, REAL, TIME). aPON and aSTR raise alarms, i.e. alarm power-on (“warm” restart) and alarm start (“cold”).

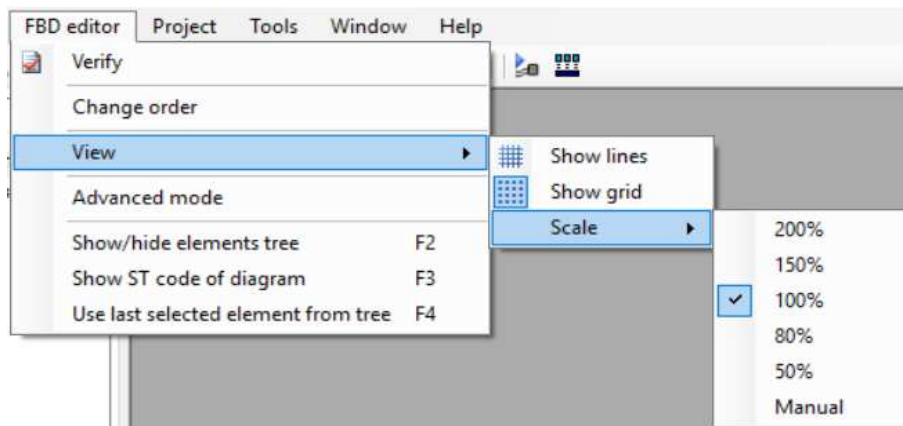


## Find element

- Expanding the tree to find rarely used function or function block could be cumbersome. To make it easier, user can simply write a part of name in the text box above the tree to trigger automatic filtering of elements.
- Function subbranches and/or libraries with elements including the written text are presented. For example, after typing SEL corresponding function appears in the tree. Another text may result in a few elements.



## Editor options

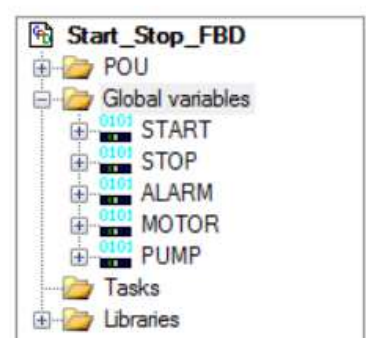


- The following options are available in *FBD editor* menu:
  - *Verify* – checks diagram completeness (elements, connections, etc.)
  - *Change order* – changes execution order of all successive elements (or preceding)
  - *View* – contains options of diagram presentation:
    - *Show lines* – switches on/off supplementary lines on the diagram
    - *Show grid* – switches on/off grid and grey lines on the diagram
    - *Scale* – chooses scale ratio
  - *Advanced mode* – switches on/off advanced mode, where some extra functions and blocks are available (and visible in the tree)
  - *Show/hide elements tree* **F2**
  - *Show ST code of diagram* **F3** (after *Build*)
  - *Use last selected element from tree* **F4**.

## Development steps

### Global variables

- Global variables are usually declared at the beginning of creating the project (see basic CPDev instruction).
- Declared global variables shown in the project tree are included into hint list.
- The list appears in *Variable properties* window (see below).



### Diagram

- Place all elements on the drawing board at locations corresponding to processing order. Temporary names are given automatically. Leave **sufficient space** between the elements for connections.
- Make eventual adjustments of original locations. Take printing into account. Careful initial planning limits later corrections considerably.
- Change temporary names of:
  - input/output variables into names of global variables and target names of local variables
  - function block instances into target names.
- Write in values and types of constants.

- Create connections between inputs and outputs, preferably in the following steps:
  - short connections between constants, parameters and inputs of corresponding blocks
  - connections between adjacent blocks
  - connections between remaining blocks
  - branch points and relevant connections
  - connections to external inputs and outputs.
- Check execution order of functions and function blocks.
- *Verify* FBD diagram (editor option).

### **Task. Compilation. Simulation**

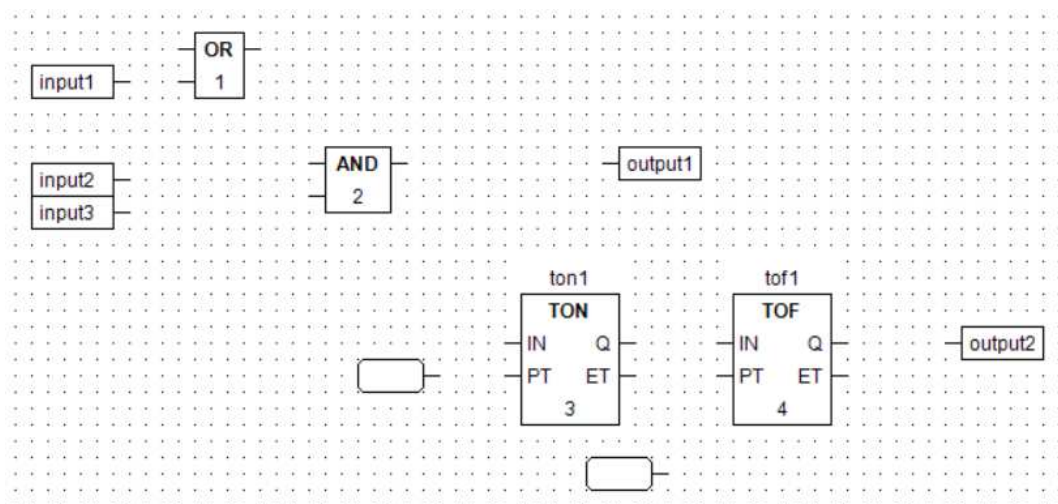
- Declare task, select cycle and assign program (programs) to the task.
- During compilation the open diagram is verified again and the whole project translated into ST language. If the ST translation does not have errors, it is compiled into VMASM code of virtual machine.
- Online operating mode and CPSim simulator provide tracing of variable values during execution.

## **Elements of START-STOP**

### **Placing elements on the diagram**

- To **place element** on the diagram:
  - select the element in the tree on the left
  - drag it to the target location (or select and drop).
 Execution numbers are automatically selected while placing elements on the diagram.
- To **move element**:
  - select it
  - press and hold left mouse button and move the cursor to new location
  - release the button.

*Remark.* Element cannot be moved into new place if there is not enough room. It is shown in red and returns to previous place after releasing the button.

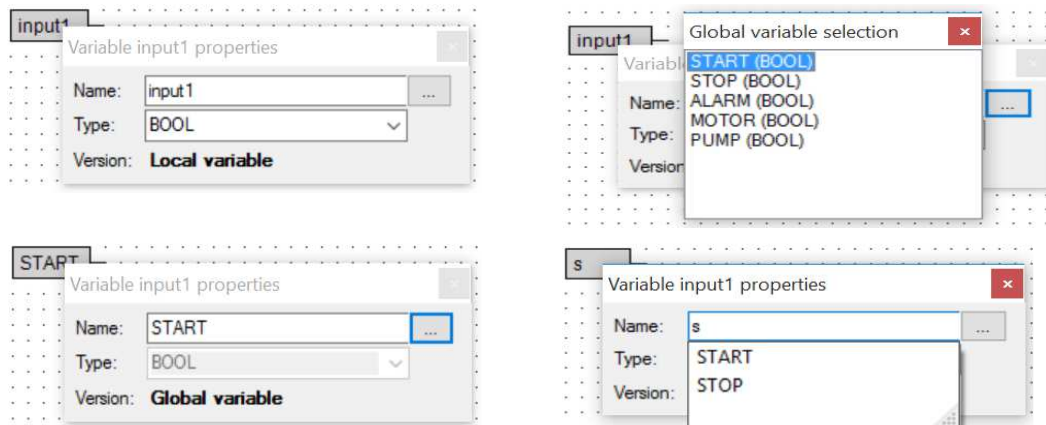


- To **remove element**:
  - select it (becomes gray)

- press *Delete* key or right-click, select *Delete* in the window and left-click.

## Input, output and global variables

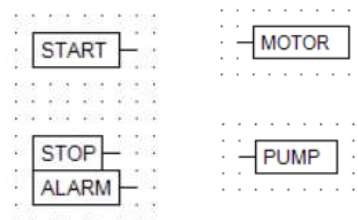
- *Variable properties* window allows to change **name** of variable from temporary name such as **input1** or **output1** given automatically. To open the window, **double-click** the variable or select it and choose *Properties* option from context menu.
- If the project has been rebuilt after declaration of global variables, pressing the **ellipsis** button (...) or **F2** opens the new *Global variable selection* window where the user can choose the variable without typing. The choice is confirmed by pressing *Enter*, *Escape* or closing the window.



- *Variable properties* window also provides **name prompting** functionality as an alternative to the ellipsis button or **F2**. This may be convenient for long global variable list.
- Variables with names other than in the global variable list automatically become local.

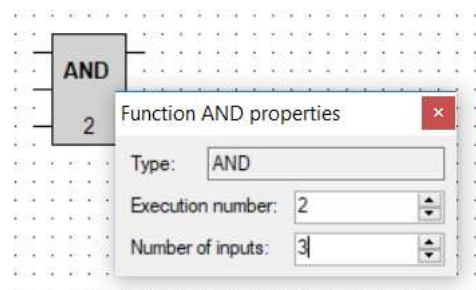
*Remark.* CPDev compiler does not distinguish lower-case and upper-case characters. So for instance *name1* and *NAME1* denote the same variable.

- To close *Variable properties* window, press *Enter*, *Escape* or closing button.

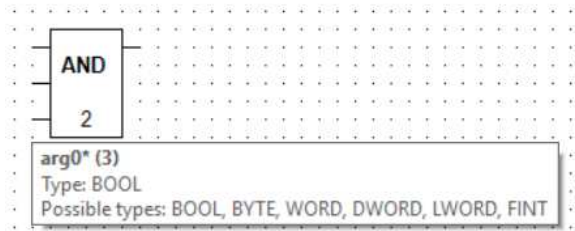


## Functions

- **Number of inputs** of some functions can be selected as follows:
  - open *Function properties* window (double-click the function)
  - choose suitable number of inputs ( $2 \div 15$ ).
- Number of inputs can also be incremented or decremented by selecting the function on the diagram and pressing **]** or **[** bracket keys (+1 or -1).
- To change execution order:
  - choose suitable execution number in *Functions properties* window.

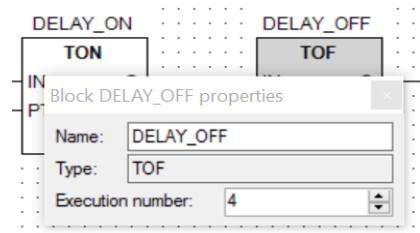


- Type of function inputs matches type of output while making first connection. Available types are shown in tooltip.



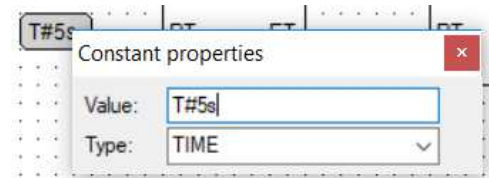
### Function blocks

- While placing function block on the diagram a temporary instance name appears above the block.
- Double-click the block to change temporary name in *Block properties* window.
- Block width is updated automatically every time when length of the name is changed.
- *Block properties* window also allows to change execution order.



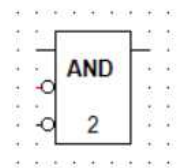
### Constants

- Double-click the empty rectangle (rounded) to enter value and type in *Constant properties* window. Type may be prompted automatically according to value.



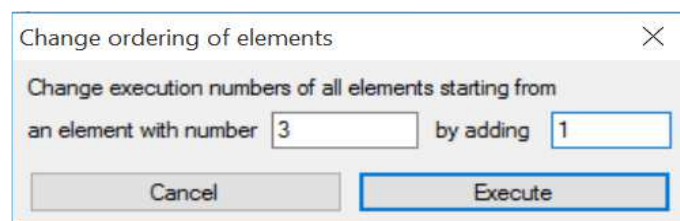
### Input/output negation

- **Right-click** particular input or output of function or function block to negate it (boolean signals only).



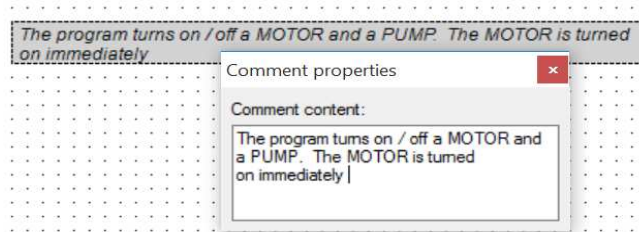
### Change order of elements

- Making room for new elements inside sequence of execution numbers of already existing elements is common application of *Change order* option of FBD editor. For example, if new element should be executed as 3<sup>rd</sup>, then execution numbers of all existing elements starting from 3 must be increased by 1.



### Comments

- After adding a comment to the diagram, double-click the empty rectangle. The window, where content can be defined, is presented.

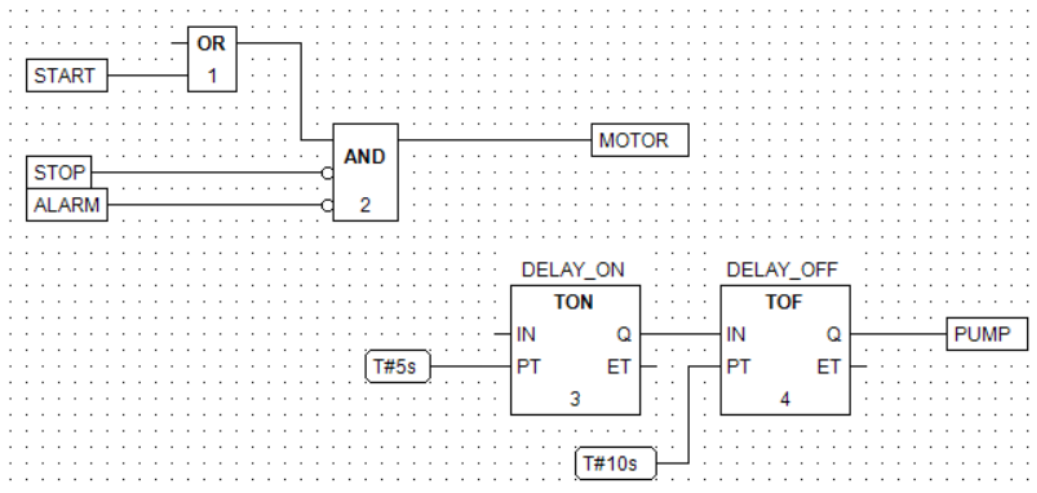


*Remark.* Comments should be added at the final stage, otherwise their locations would be taken into account by automatic connections algorithm.

## Creating connections

### Connection

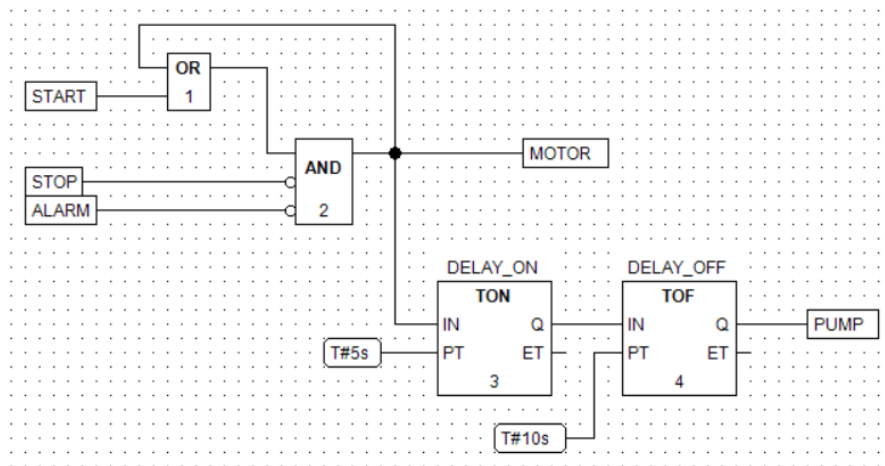
- Connection can be drawn from input to output or from output to input.
- To **create connection**:
  - select or point at the input or output, which will be the beginning of the connection (color changes to red)
  - press left mouse button and drag cursor to the output or input which will be the end of the connection (while moving the cursor, dashed line from the beginning is drawn)
  - release left mouse button, what creates the connection with **path calculated and drawn automatically**.



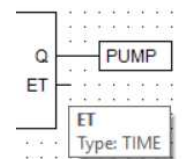
### Branch point

- **Right-click** any place on the connection line to add branch point with two tips.
- Connection between the branch point and input of an element is done in the same way as between input and output.





- For information, type of each input/output of any element is shown in tooltip after indicating the element's tip.



## Editing

- Selection of more than one element:
  - select elements with Ctrl key pressed
  - alternatively, press left mouse button at any empty place on the diagram and surround the elements.
- Moving:
  - with element/elements selected, press and hold left mouse button while moving the cursor to new location
  - release the button.

*Remark.* Moving an element (or elements) updates connections automatically **calculating relevant paths anew**. They may be different than before.

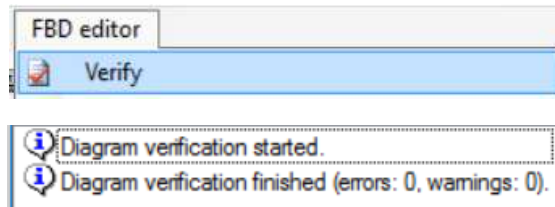
- Removing:
  - press Delete after selecting element/elements.
- Copy and paste:
  - select element/elements
  - copy (Ctrl+C)
  - move cursor to new location
  - paste (Ctrl+V).

## Tips on automatic connections

- Do not make the diagram **too tight**, since some of the connections may be drawn around.
- Undesirably looking connection may be corrected by deleting it temporarily, moving relevant elements apart and connecting them again.
- While moving an element, two different connections originally close to each other may overlap. Then remove one of them and proceed as above.
- Complicated diagram may be split into subdiagrams “connected” by local variables (outputs in one, inputs in another).

## Verification

- Choose *Verify* option from the editor menu. It checks completeness of the diagram (connections) and consistency of types of inputs and outputs. Results are reported at the bottom of CPDev window.

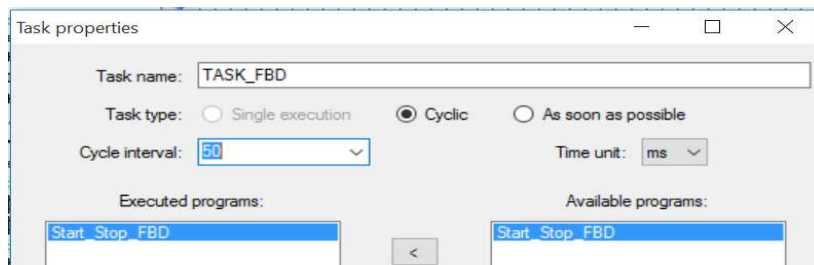


- If FBD-to-ST translation of the verified diagram is of interest, press **F3** (see below). Open diagram is automatically saved every 10 minutes into temporary directory followed by verification (see *Automatic save*).

## Task and compilation

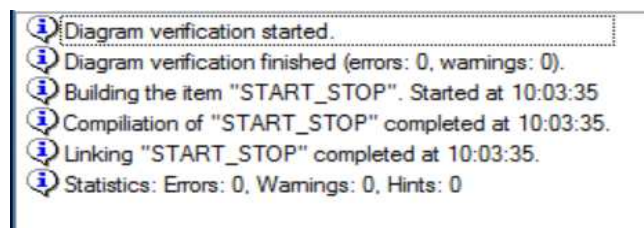
### Task

- Task can be added to the project in two ways:
  - from project tree by *Tasks* → *Add task* or by *Start\_Stop\_FBD* (project name) → *Add item* → *Task*
  - from CPDev main menu by *Project* → *Item* → *Add* → *Task* (in *Adding new element* window).
- Enter *Task name*, here TASK\_FBD, choose *Cycle interval* and *Executed programs* from *Available* ones.



### Compilation – Build

- By clicking *Build* icon in the toolbar or selecting *Project* → *Build* in the menu the whole project is first translated into ST language and then compiled into VMASM code for virtual machine. If errors occur, compilation is stopped. Warnings are reported, however the program can still be compiled. The project may be eventually saved before compilation (*Environment Options* → *Editing*).
- Verification runs automatically before compilation provided that editor window is open.



*Remark.* Program may be compiled even without task definition, however a warning will appear.

- ST code of the translated diagram may be shown by pressing **F3** (click the diagram first).

```

Code in ST language
001 PROGRAM Start_Stop_FBD
002
003 VAR_EXTERNAL
004   START: BOOL;
005   STOP: BOOL;
006   ALARM: BOOL;
007   MOTOR: BOOL;
008   PUMP: BOOL;
009 END_VAR
010
011 VAR
012   out_orl (*$HIDDENONLINE*) : BOOL;
013   out_andl (*$HIDDENONLINE*) : BOOL;
014   DELAY_ON: TON;
015   out_DELAY_ON_Q (*$HIDDENONLINE*) : BOOL;
016   DELAY_OFF: TOF;
017   out_DELAY_OFF_Q (*$HIDDENONLINE*) : BOOL;
018 END_VAR
019
020 out_orl := OR(out_andl, START);
021 out_andl := AND(out_orl, NOT STOP, NOT ALARM);
022 MOTOR := out_andl;
023 DELAY_ON(IN:=out_andl, PT:=T#5s, Q=>out_DELAY_ON_Q);
024 DELAY_OFF(IN:=out_DELAY_ON_Q, PT:=T#10s, Q=>out_DELAY_OFF_Q);
025 PUMP := out_DELAY_OFF_Q;
026
027 END_PROGRAM

```

- Local variables with (\*\$HIDDENONLINE\*) directives created automatically do not appear on variable value list in online mode.

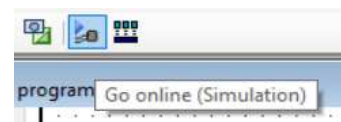
## Online mode

Tracing execution of a project is provided by CPDev online mode. The project may consist of a number of programs written in the same or different languages. Tracing is particularly convenient for graphic languages, such as FBD.

### Go online

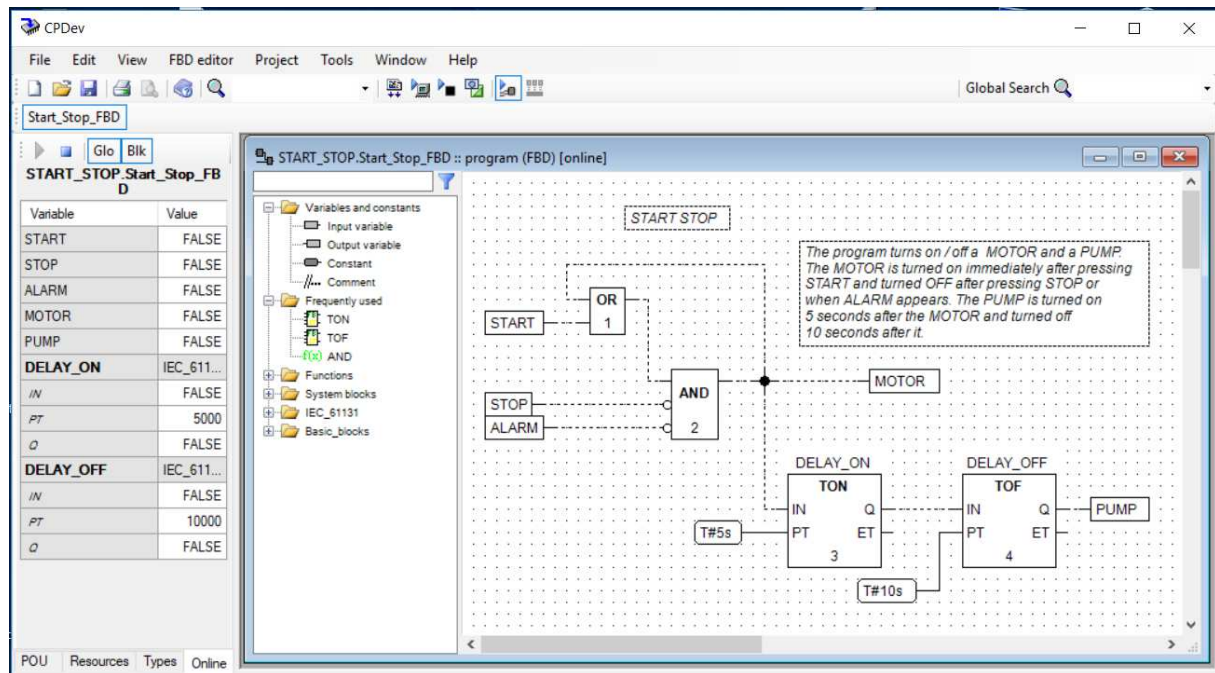
Before going into online mode the project must be built (rebuilt after changes).

- Click *Go online* icon in the toolbar (becomes framed) or select *Project → Go online* in the menu.
- The icon indicates that *Data sources* are set to *Simulation*.



### FBD diagram online

- CPDev window in online mode involves “live” FBD diagram and list of variable values on the left. BOOL values are distinguished by continuous (TRUE) or dotted (FALSE) connections. Values of other types are written above connections, except connections to constants.

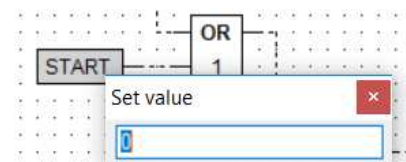


- Value may also be shown in tooltip after indicating a connection or tip of element. Type appears as well if tip is indicated (TIME in milliseconds).



- Change value:**

- select input variable
- double-click
- enter new value, accept or close the window.  
Change is reflected on the value list.  
TRUE, FALSE may be entered as 1,0.  
Constants cannot be changed in online mode.



- If a few diagrams are open, the one on the top is "live". It is indicated by [online] after name of the program.

## Value list

- List with values involves variables associated with “live” diagram, so global and local variables, and function blocks with inputs/outputs. If no program is open, global variables are shown only.
- Change value:**
  - click current value (becomes blue)
  - enter new value, accept.
- Icons above the list provide:
  - start simulation again (if stopped before)
  - stop simulation temporarily (still in online mode)
  - show/hide global variables
  - show/hide inputs/outputs of function blocks.
 Stop freezes current values and FBD diagram. Start resumes online operation.
- The open list is associated with appearing *Online* tab at the bottom. Clicking *POU* tab on the left opens project tree, for instance to choose another program for tracing. Then click *Online* again.

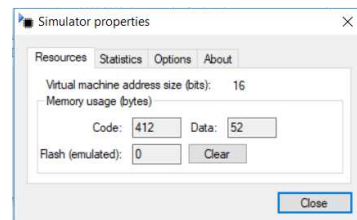
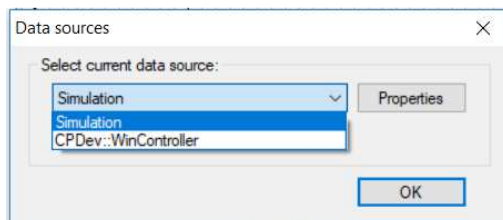
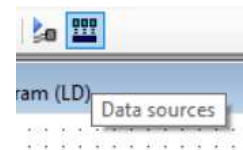
Variable	Value
START	TRUE



Variable	Value
START	FALSE
STOP	FALSE
ALARM	FALSE
MOTOR	FALSE
PUMP	FALSE
DELAY_ON	IEC_611...
IN	FALSE
PT	5000
Q	FALSE
DELAY_OFF	IEC_611...
IN	FALSE
PT	10000
Q	FALSE

## Data sources

- Clicking *Data sources* icon shows data sources for online mode, beginning with *Simulation*. The second, WinController, is an advanced CPDev runtime for Windows. Other sources, if shown, represent specific controllers for which online mode means *commissioning*.

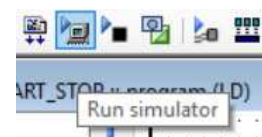


- Some data of simulated project are shown in *Simulator properties* window.

## CPSim simulator

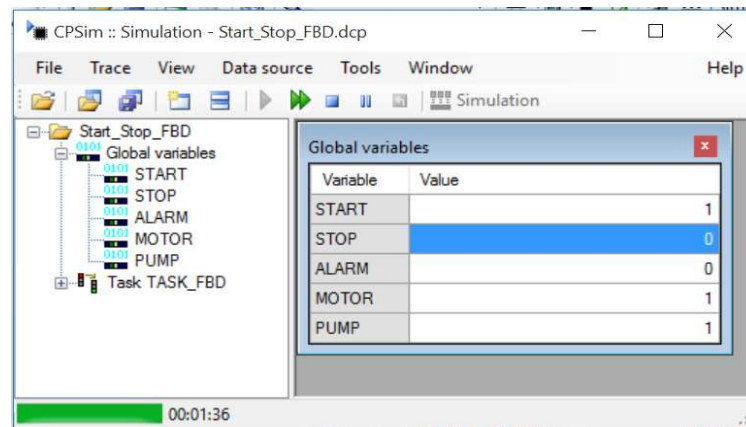
Another way of tracing variable values during execution is provided by CPSim simulator. It is more flexible than online mode allowing to assign most important variables to panels with buttons and “LEDs”, as well as to individual views or custom lists.

- CPSim is run by clicking *Run simulator* icon in the toolbar or by selecting *Project → Run simulator* (also *Tools → Simulator*).



## Global variables

- *CPSim* window presents initially the list of global variables. Execution begins by clicking *Start trace* icon or by choosing *Trace* → *Start*.

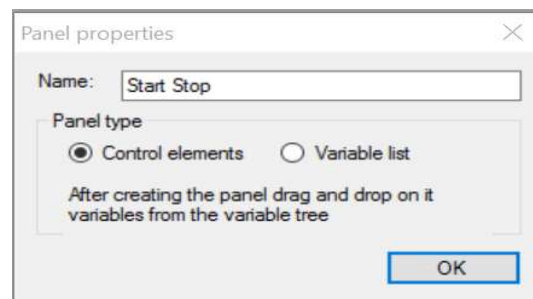


- Values of those global variables which are inputs to the program can be changed. More on CPSim can be found in *Help* → *Programming instruction*.

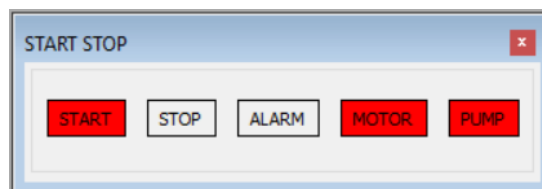
*Remark.* Values of boolean variables are displayed as 0 (FALSE) and 1 (TRUE).

## Control panel

- More convenient way of testing is provided by selecting *Group panel* icon or by relevant option in *View*.



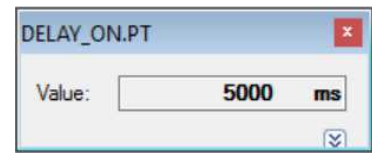
- Type name in *Panel properties* window, accept and drag required variables from simulation tree on the left into the rectangle panel (which grows accordingly).



- Test the program by pressing input buttons (BOOL) or by typing values in input cells (other types).

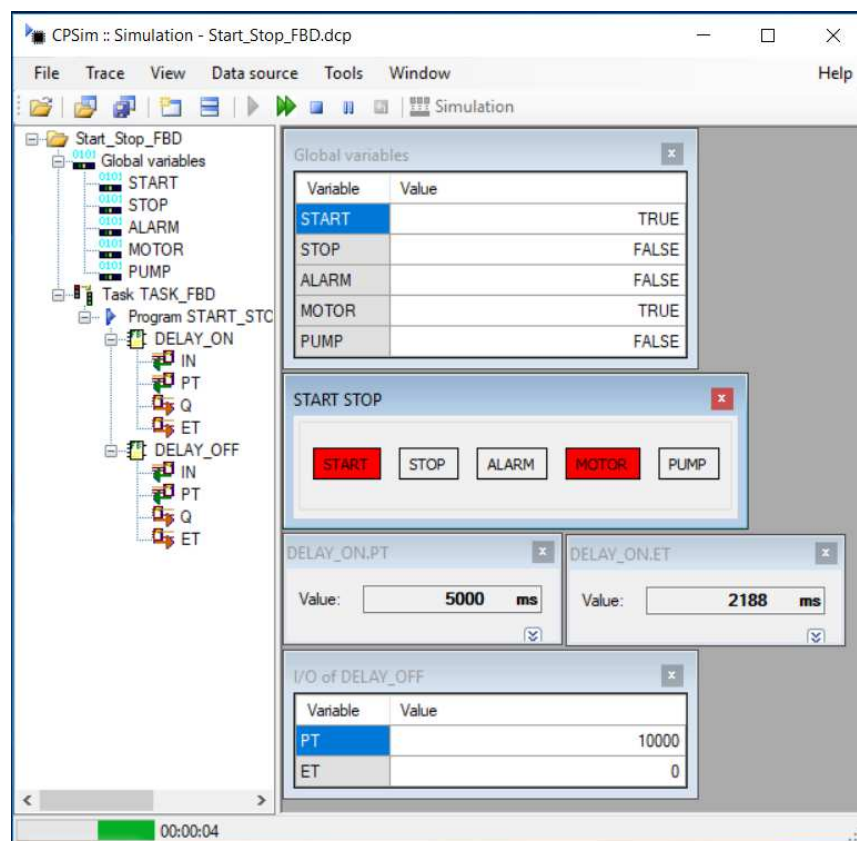
## Individual view

- Single variable can be traced by dragging it from the tree into view area. Individual views are used for important local variables and inputs/outputs of function blocks (PT here).



## Variable list

- By choosing *Variable list* in *Panel properties* window a list of variables dragged from the tree is created. Inputs/outputs of function blocks or local variables used in the diagram (if any) are usually assigned to such list (which looks the same as the list of global variables).



## Sessions

- Lists, panels and views created for simulations can be saved for future use by selecting *Trace* → *Save session* or by yes answer to the question *Do you want to save the trace session?* asked on CPSim exit. Session data are saved in the project folder as SCP file.
- Complex project may require a few simulation sessions with different views. Data of each session should be saved in separate file and run by *Trace > Open session...*

## User library

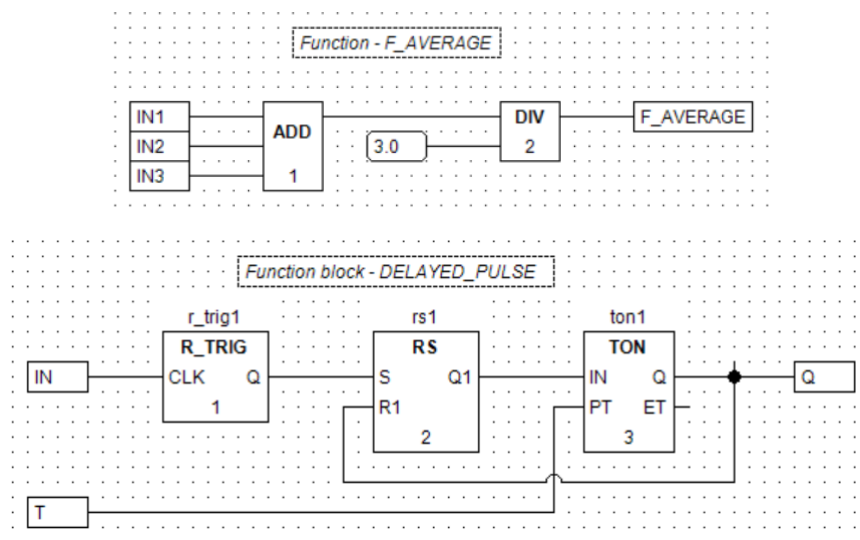
User can create his own functions or function blocks and store them in a library (programs can be stored as well). Two examples are shown below:

- function F\_AVERAGE for calculation of average from three REAL inputs

- function block DELAYED\_PULSE for generation of one-cycle impulse after time T since leading edge appears at the input.

### Source project

- Create a new project with suitable name and add two new POU's with appropriate object types, i.e. function F\_AVERAGE and function block DELAYED\_PULSE.
- Creating the function and function block in FBD language proceeds in the same way as described earlier.
- In case of function block, input variables placed on the diagram are translated into block inputs and output variables into outputs. Names of the inputs and outputs will be shown while using the block in this or another project.
- In case of function, input variables are translated into function inputs and output variable with the same name as function is translated into value returned by the function. As indicated in IEC 61131-3 standard, names of the inputs will not be shown while using the function (however, they can be found in the project tree; see below).



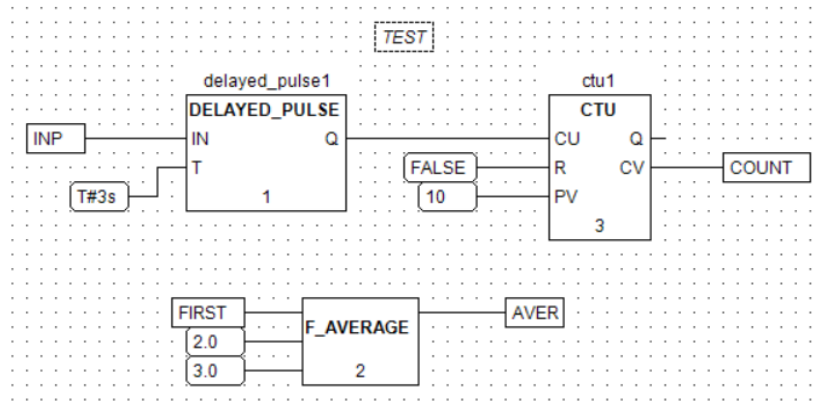
- **Save** source XML project of the library for future extensions and for recompilation in case when new version of the compiler is installed (see *Library warnings*).

### Export as a library

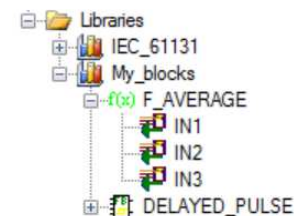
- To export project as a library:
  - choose options *Export* → *Library* from *Project* menu
  - in *Export project as library* window:
    - write library name (for instance *My\_blocks* below)
    - choose library file location (press ellipsis button (...))
    - select POU's exported into library
    - press *OK*.
- Library is saved as LCP file.

## Using the library

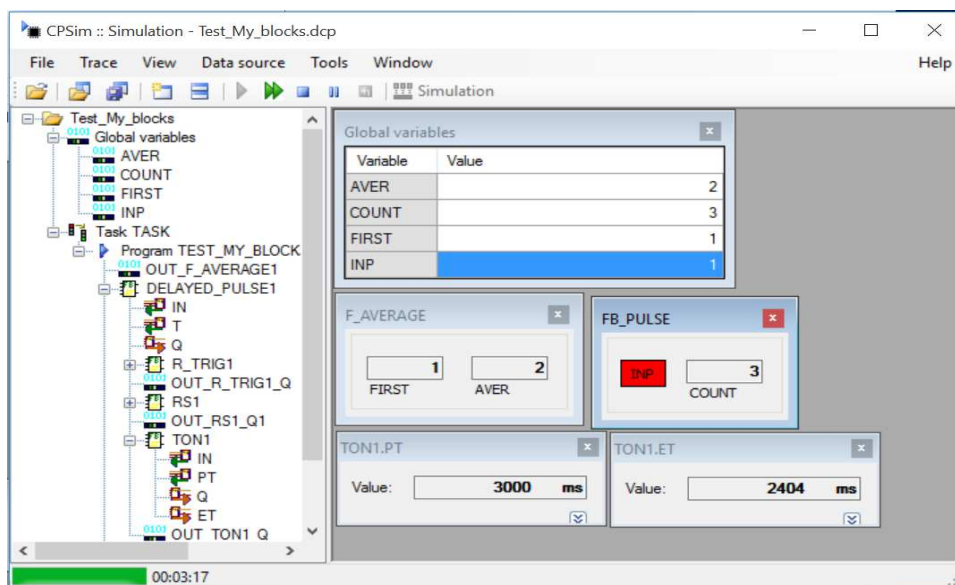
- Library imported to a project is available in the tree of FBD editor and also shown at the left part of the main CPDev window. Function blocks and functions stored in it can be used in FBD diagram.
- FBD program shown below tests DELAYED\_PULSE and F\_AVERAGE. CTU block counts pulses.



- To remind names of F\_AVERAGE function inputs one can expand F\_AVERAGE in the *Libraries* section of the main CPDev window.



- Views of sample simulation session for *Test\_My\_blocks* project are shown below. Note that the last two individual views present input and output of the timer which is a component of user-created function block (third level of hierarchy).



## Other issues

### Changing FBD program name

Program name given initially needs to be changed sometimes in the final version.

- To change program name:
  - open the diagram (if closed)
  - select the program in the project tree
  - choose *Project* → *Items* → *Rename*
  - enter new name in the tree
  - accept (*Enter*)
  - build and save the project.

### Library warnings

When library is linked to a project, CPDev automatically stores library version, timestamp and number of compiler version which created the library (LCP file). When the project is open again and built, those markers are compared with markers of actual library and compiler. Inconsistencies trigger warnings.


- The following warning indicates outdated library:

 Declared library timestamp *date* is not equal to timestamp found in library *name* from file "C:" *location*

If library content and compiler version have not been changed, ignore the warning and proceed with the project. Warning will not appear on next opening.

Otherwise:

- remove the library from the project tree by selecting the library and choosing *Remove* from context menu
  - import actual library into the project by *Project* → *Import* → *Library* from the file indicated by location.
- New version of CPDev compiler includes recompiled system libraries, i.e. *IEC\_61131.lcp* and *Basic\_blocks.lcp*. However user library, like *My\_blocks.lcp* above, has not been recompiled yet. Therefore the following warning will appear:

 Library *name* file "C:" *location* was designed for other compiler than currently used

- To remove the warning:
  - compile source code of the user library with the new compiler
  - export it as LCP file into given location
  - remove the library (with old timestamp) from the project tree
  - import the recompiled library into the project.

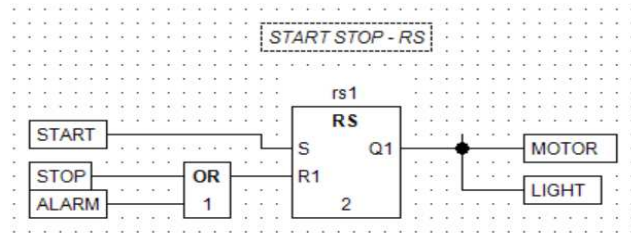
### Automatic save

- Every 10 minutes (default) CPDev creates a file in **temporary directory** (see operating system settings) called *CPDev\_ProjectNameOrFileName\_EightHexNumbers.xml* with source code of the project. This is a back-up in case of CPDev crash.
- If FBD diagram has been changed within that period, it is automatically verified (if open).
- The automatic save period may be changed in *Environment options* → *Editing*

## Examples

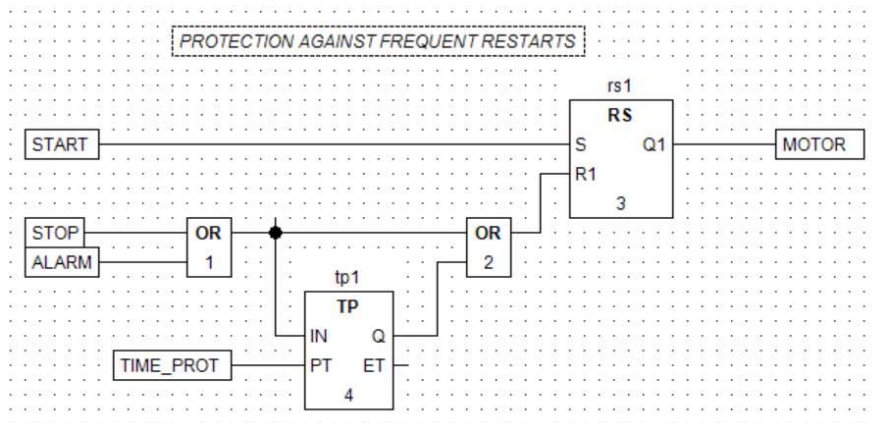
### Start stop – RS

- Implementation of START STOP circuit with RS flip-flop.



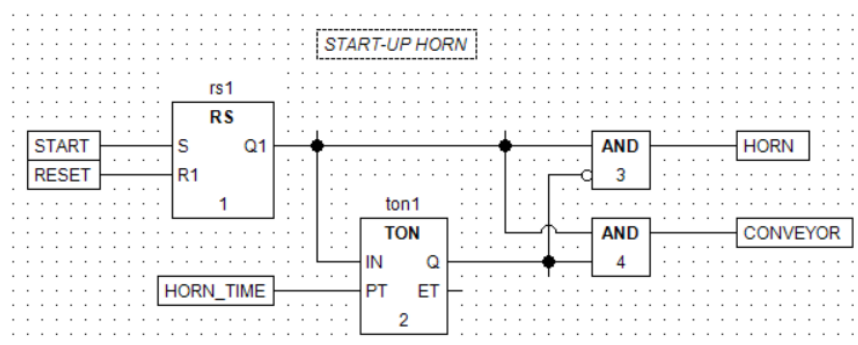
### Protection against frequent restarts

- After STOP or ALARM (thermic sensor) another restart possible only after time set by TIME\_PROT variable (initially 10s, i.e. 10000ms).
- CPSim: control panels with I/O and time data.



### Start-up horn

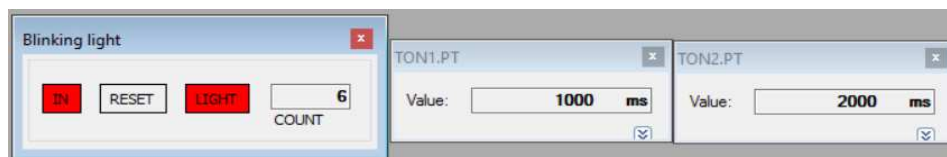
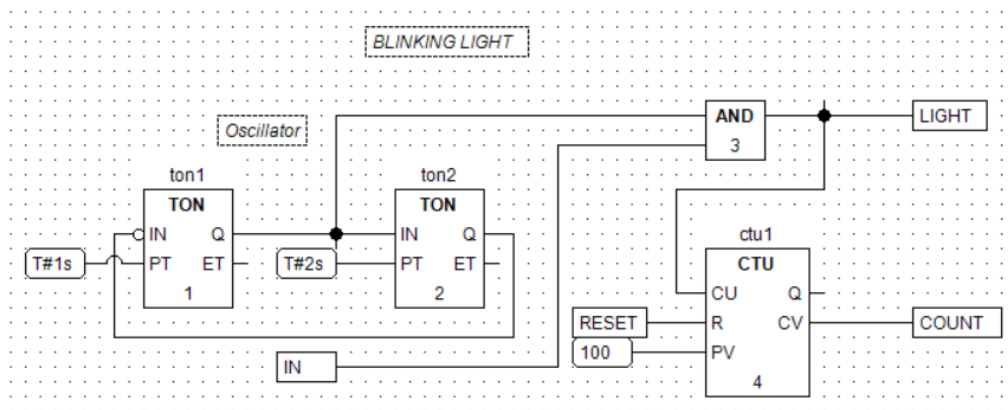
- START turns HORN on for time set by TIME\_HORN (10s). Then CONVEYOR begins to move.
- CPSim: control panel while HORN is on.





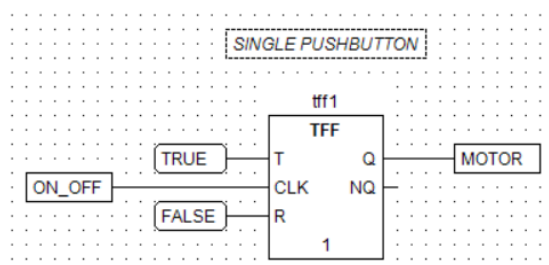
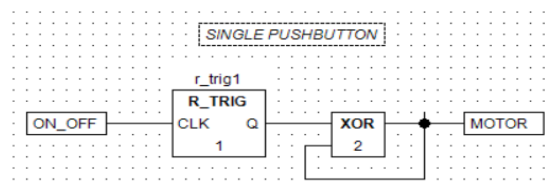
### Blinking light and counter

- LIGHT blinks when IN is on. The two timers implement a pulse generator (oscillator). CTU counts pulses.
- CPSim: control panel with I/O and individual views.



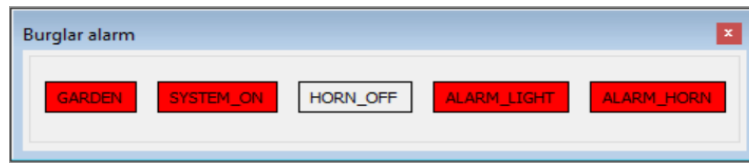
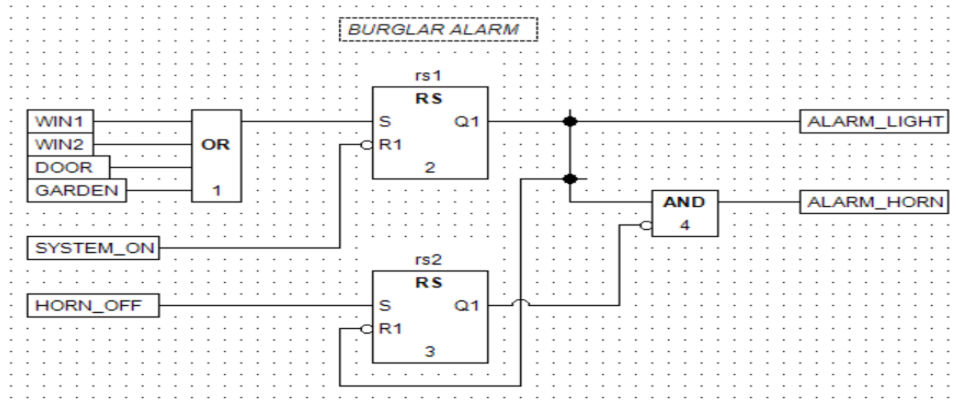
### Single pushbutton

- First pressing ON\_OFF turns MOTOR on and another turns it off.
- Implementations with blocks from *IEC\_61131* and *Basic\_blocks* libraries.

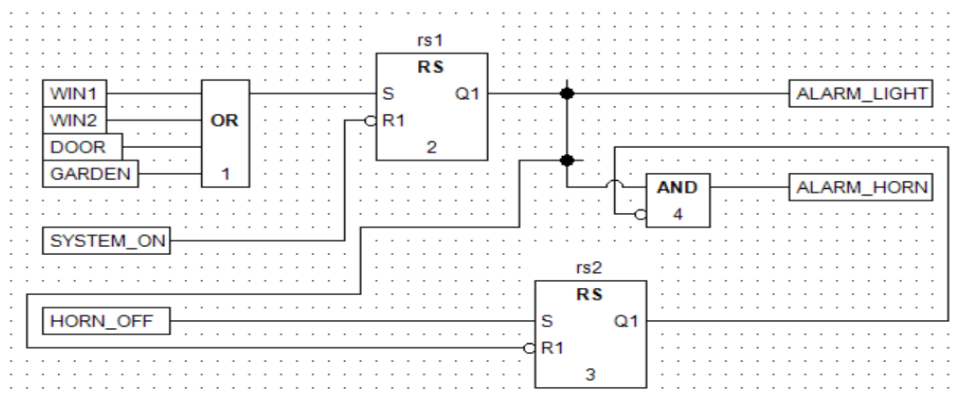


### Burglar alarm

- Sensors WIN1 (window), WIN2, DOOR and GARDEN activate ALARM\_LIGHT and ALARM\_HORN, provided that SYSTEM\_ON is set. HORN\_OFF switches off ALARM\_HORN till SYSTEM\_ON is reset.

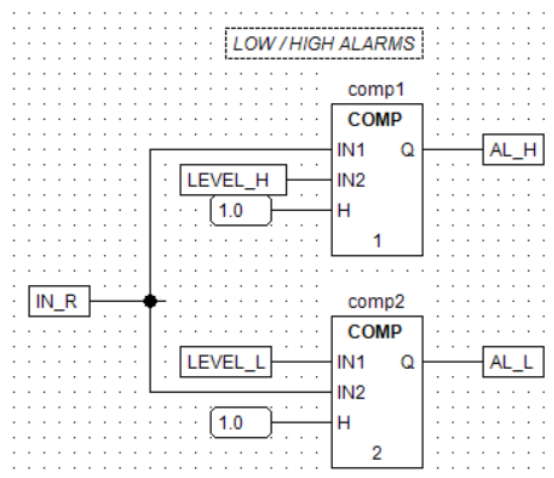
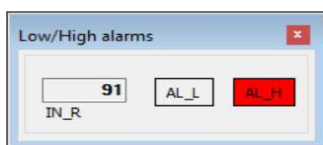
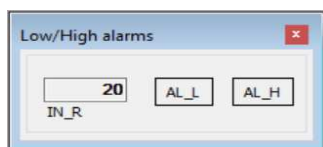


- Note that if connections are not created in recommended steps (see *Tips on automatic connections*) the following diagram may result due to automatic calculation of paths.



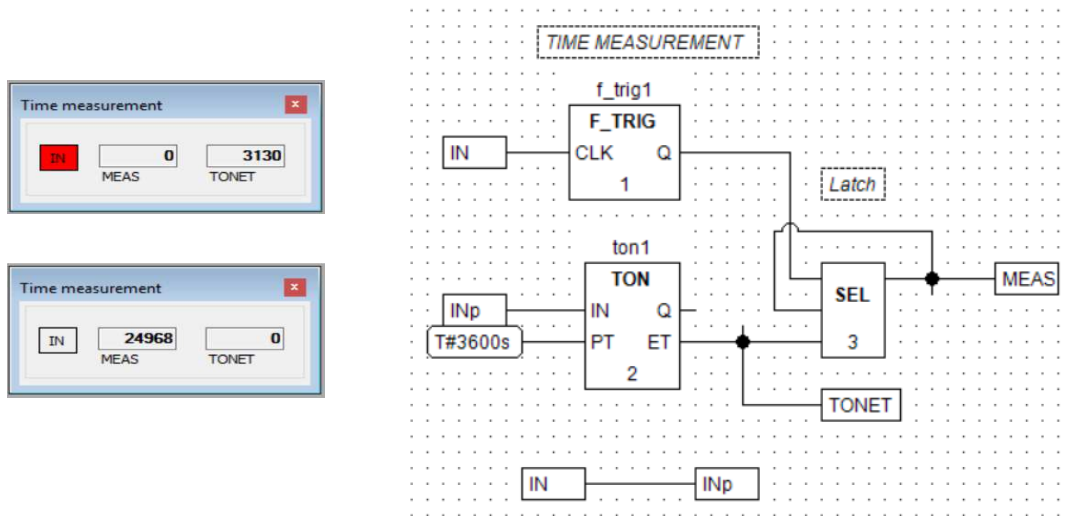
## Low/High alarms

- AL\_H or AL\_L arise when IN\_R (REAL) violates LEVEL\_H (90%) or LEVEL\_L (10%) alarm level. COMPArators are set with 1% hysteresis (*Basic\_blocks*).



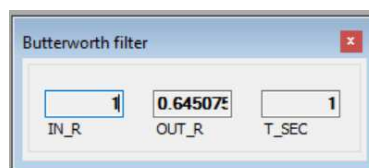
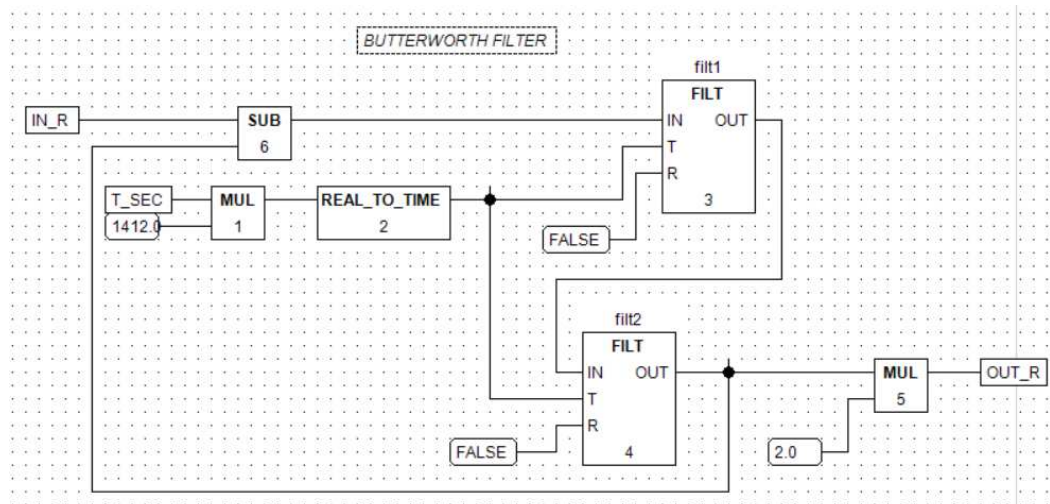
## Time measurement

- Time period  $T_x$  is measured when IN is on and indicated as MEAS. TONET shows elapsed time. SEL latches measured value at the falling edge of IN. INp denotes IN delayed by one cycle.



## Signal filter

- Two FILTERs (*Basic\_blocks*) with feedback connection implement a low-pass Butterworth filter (damping  $1/\sqrt{2}$ ) with time constant  $T\_SEC$  (REAL, in seconds). Recall that T inputs of the filters must be  $\sqrt{2}$  larger, therefore multiplication by 1412.0 and REAL\_TO\_TIME conversion is needed.



# Printing

## Size adjustment

- Grey lines partition total work area of the editor into printed pages. Final size of the diagram may be decided before printing to fill in the pages conveniently.

*Remark.* Diagrams presented above are fairly tight (for consistency). They may be widened taking into account the grey lines, as the START\_STOP printout shown farther.

## Page setup

- Choose *Tools* → *Environment options* → *Configuration* → *Page setup*.

The screenshot shows the 'Configuration' dialog box with the 'Page setup' tab selected. The 'Border margins' section includes input fields for Left (0.5 cm), Right (0.5 cm), Top (0.5 cm), Bottom (0.5 cm), and Padding (0.3 cm), along with a checked 'Draw border' option. The 'Paper/Orientation' section shows 'Paper size' set to A4 and 'Portrait' orientation selected. The 'Printing scale' section shows 'User scale' set to 100%. The 'Title block fields' section has a checked 'Draw title block' option and a table of fields: Subject, Company, Project name, Project version, POU name, Author, Filename, Date, and Page number, all of which are checked. An 'Update settings upon printing session' checkbox is also checked. 'OK' and 'Cancel' buttons are at the bottom.

- Upper part of the options involves *Border margins*, *Paper size*, *Orientation* and *Printing scale* (independent of the editor scale).
- Number of pages being printed is determined automatically according to the size of the diagram and printing scale (user selected). Small diagrams need one page, large ones a few.
- Lower part specifies what fields appear in the title block (table) printed at the end of each page. Table changes somewhat if some fields are not selected.

Subject:			Company:	
Project: START_STOP	Version:	POU: Start_Stop_FBD		Author:
File: C:\Users\Leszek Trybus\Documents\CPDev - programy, instrukcje\Projekty...			Date: 08.02.2018 11:41	Page: 1 / 1

- Data for *Subject*, *Company*, *Project*, *Version*, *Author* and *File* are copied from *Project properties* window. Active window determines *POU* name.

*Remark.* Options of *Page setup* are kept in CPDev database (*Environment options*), so they apply to each printing till changed.

## Print

- Choose *File* → *Print* in the main menu or press **Ctrl+P** keys or icon in the toolbar (direct print).
- Select printer and preferences in printer selection window. Change of preferences and printout adjusts grey lines and *Page setup* options accordingly.
- Press *Print*.

*Remark.* Printing to virtual printer such as PDF Creator or Microsoft Print to PDF is convenient way to verify expected printout. Horizontal orientation (landscape) suits FBD usually, while vertical (portrait) LD and SFC.

