

1. OVERVIEW

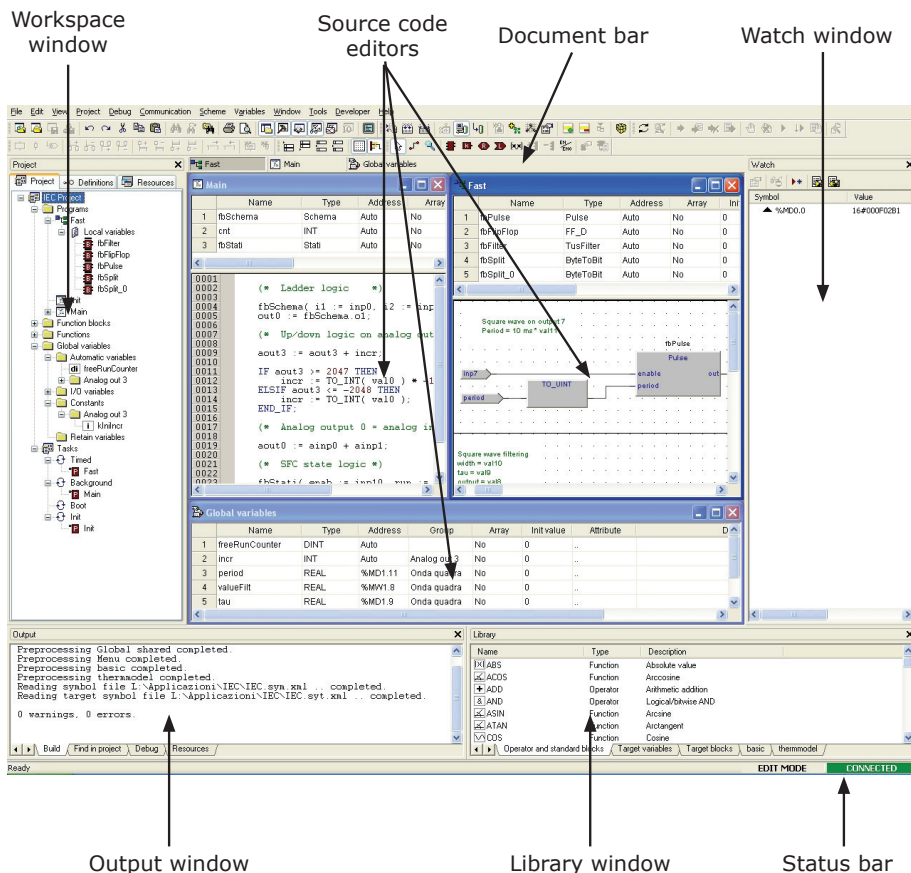
LogicLab is an IEC61131-3 Integrated Development Environment supporting the whole range of languages defined in the standard.

In order to support the user in all the activities involved in the development of an application, LogicLab includes:

- textual source code editors for the Instruction List (briefly, IL) and Structured Text (briefly, ST) programming languages (see Chapter 6.);
- graphical source code editors for the Ladder Diagram (briefly, LD), Function Block Diagram (briefly, FBD), and Sequential Function Chart (briefly, SFC) programming languages (see Chapter 6.);
- a compiler, which translates applications written according to the IEC standard directly into machine code, avoiding the need for a run-time interpreter, thus making the program execution as fast as possible (see Chapter 7.);
- a communication system which allows the download of the application to the target environment (see Chapter 8.);
- a rich set of debugging tools, ranging from an easy-to-use watch window to more powerful tools, which allows the sampling of fast changing data directly on the target environment, ensuring the information is accurate and reliable (see Chapter 9.).

1.1 THE WORKSPACE

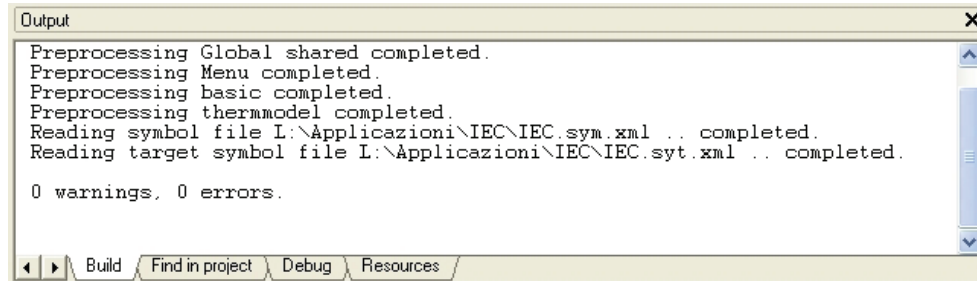
The figure below shows a view of LogicLab’s workspace, including many of its more commonly used components.



The following paragraphs give an overview of these elements.

1.1.1 THE OUTPUT WINDOW

The *Output* window is the place where LogicLab prints its output messages. This window contains four tabs: *Build*, *Find in project*, *Debug*, and *Resources*.



Build

The *Build* panel displays the output of the following activities:

- opening a project;
- compiling a project;
- downloading code to a target.

Find in project

This panel shows the result of the *Find in project* activity.

Debug

The *Debug* panel displays information about advanced debugging activities (for example, breakpoints).

Resources

The *Resources* panel displays messages related to the specific target device LogicLab is interfacing with.

1.1.2 THE STATUS BAR

The *Status* bar displays the state of the application at its left border, and an animated control reporting the state of communication at its right border.



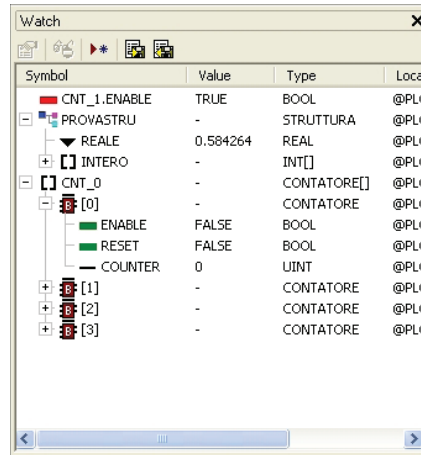
1.1.3 THE DOCUMENT BAR

The *Document* bar lists all the documents currently open for editing in LogicLab.



1.1.4 THE WATCH WINDOW

The *Watch* window is one of the many debugging tools supplied by LogicLab. Among the other debugging tools, it is worth mentioning the Oscilloscope (see Paragraph 9.2), triggers, and the live debug mode (see Paragraph 9.4).



1.1.5 THE LIBRARY WINDOW

The *Library* window contains a set of different panels, which fall into the categories explained in the following paragraphs.

You can choose the display mode by clicking the right button of your mouse. In the *View List* mode, each element is represented by its name and icon. Instead, a table appears in the *View details* mode, each row of which is associated with one of the embedded elements. The latter mode also displays the *Type* (Operator/Function) and the description of each element.

If you right-click one of the elements of this panel, and you click *Object properties* from the dialog box, then a window appears with further details on the element you selected (input and output supported types, name of input and output pins, etc.).

1.1.5.1 OPERATORS AND STANDARD BLOCKS

This panel lists basic language elements, such as operators and functions defined by the IEC 61131-3 standard.

Name	Type	Description
<input type="checkbox"/> ABS	Function	Absolute value
<input checked="" type="checkbox"/> ACOS	Function	Arccosine
<input checked="" type="checkbox"/> ADD	Operator	Arithmetic addition
<input checked="" type="checkbox"/> AND	Operator	Logical/bitwise AND
<input checked="" type="checkbox"/> ASIN	Function	Arcsine
<input checked="" type="checkbox"/> ATAN	Function	Arctangent

Operator and standard blocks | Target variables | basic | themmodel



1.1.5.2 TARGET VARIABLES

This panel lists all the system variables, also called target variables, which are the interface between firmware and PLC application code.

Name	Type	Address	Group	Description
i Ad_InPo	INT	%MWO.1	DEB - ANALOG-DIGITAL EN...	incremental position
i Ad_NuCi	INT	%MWO.12	DEB - ANALOG-DIGITAL EN...	DSP cycles without position increment
di Ad_PeSp	DINT	%MWO.10	DEB - ANALOG-DIGITAL EN...	calculated speed
i Ad_SeOf	INT	%MWO.9	DEB - ANALOG-DIGITAL EN...	sine channel offset
di Ad_ViPo	DINT	%MWO.2	DEB - ANALOG-DIGITAL EN...	virtual position
di Ad_ViPoIni	DINT	%MWO.218	DEB - ANALOG-DIGITAL EN...	

Operator and standard blocks | Target variables | basic | thermmodel

1.1.5.3 TARGET BLOCKS

This panel lists all the system functions and function blocks available on the specific target device.

Name	Type	Description
sysMsgInterpMono	Function	Checks messages from single axis int...
sysQuiesInterpMonoPlc	Function	Verifies the end of an interpolated sin...
sysResetInterpMonoPlc	Function	Resets an interpolated single axis mo...
sysSleep	Function	Puts the current task in the sleeping s...
sysStartInterpMono	Function	Starts an interpolated single axis mov...
sysStartInterpMonoPlc	Function	Starts an interpolated single axis mov...
sysTargetInterpMonoPlc	Function	Verifies if the target of an interpolated ...
sysWaitInterpMono	Function	Waits until the end of an interpolated ...

Operator and standard blocks | Target variables | Target blocks | basic

1.1.5.4 INCLUDED LIBRARY PANELS

The panels described in the preceding paragraphs are usually always available in the *Library* window. However, other panels may be added to this window, one for each library included in the current LogicLab project. For example, the picture above was taken from a LogicLab project having two included libraries, *basic.pll* and *thermmode1.pll* (see also Paragraph 4.7).

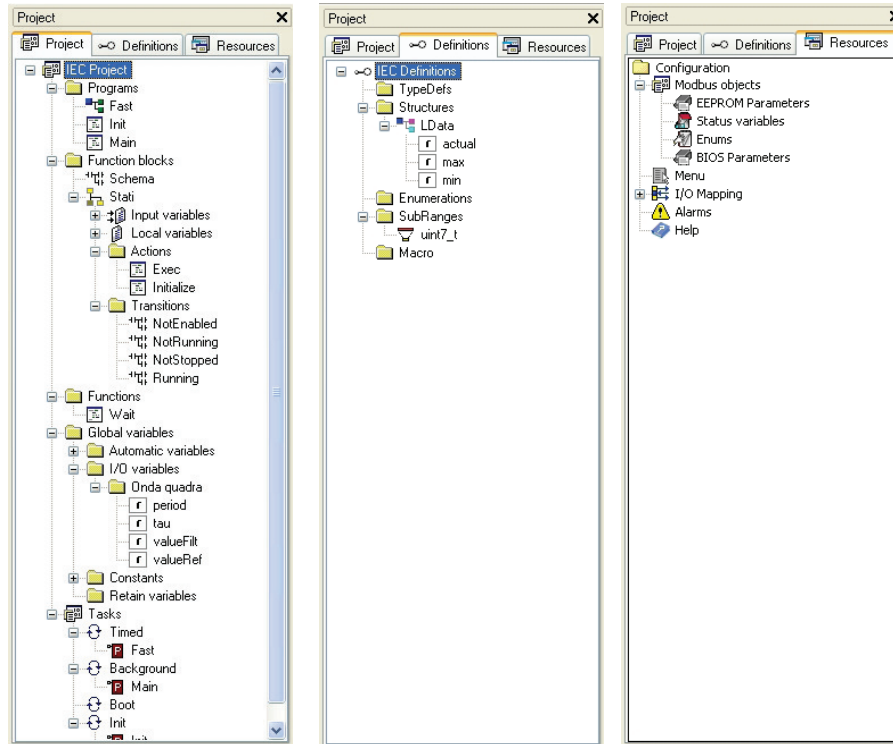
Name	Type	Description
BitToByte	Function	Compose a byte from 8 bits
BitToWord	Function	Compose a word from 16 bits
ByteToBit	Function block	Split a byte into bits
ByteToWord	Function	Compose a word from 2 bytes
F_TRIG	Function block	Falling edge detector
FF_D	Function block	D-type flip-flop

Operator and standard blocks | Target variables | basic | thermmodel



1.1.6 THE WORKSPACE WINDOW

The *Workspace* window consists of three distinct panels, as shown in the following picture.



1.1.6.1 PROJECT

The *Project* panel contains a set of folders:

- *Program, Function blocks, Functions*: each folder contains Program Organization Units (briefly, POUs - see Paragraph 5.1) of the type specified by the folder name.
- *Global variables*: it is further divided in *Variables, I/O Variables, Constants* and *Retain variables*. Each folder contains global variables of the type specified by the folder name (see Paragraph 5.2).
- *Tasks*: this item lists the system tasks and the programs assigned to each task (see Paragraph 5.3).

1.1.6.2 DEFINITIONS

The *Definitions* panel contains the definitions of all user-defined data types, such as structures or enumerated types.

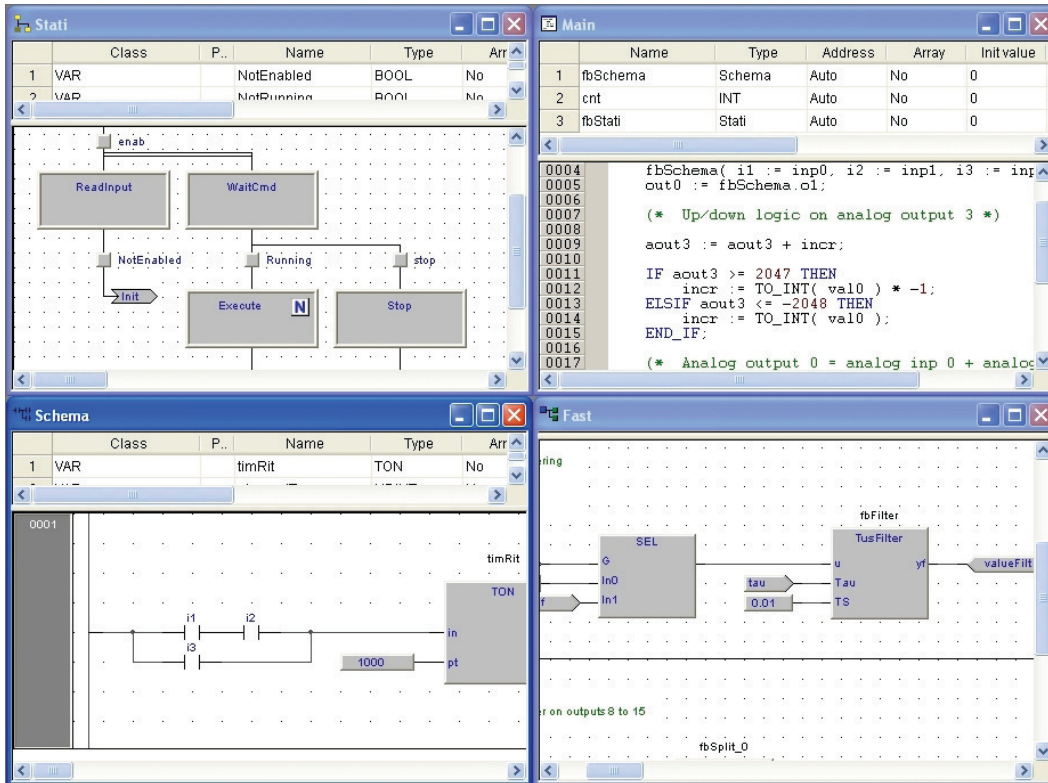
1.1.6.3 RESOURCES

The contents of the *Resources* panel depends on the target device LogicLab is interfacing with: it may include configuration elements, schemas, wizards, and so on.



1.1.7 THE SOURCE CODE EDITORS

The LogicLab programming environment includes a set of editors to manage, edit, and print source files written in any of the 5 programming languages defined by the IEC 61131-3 standard (see Chapter 6.).



The definition of both global and local variables is supported by specific spreadsheet-like editors.

	Name	Type	Address	Group	Array	Init value	Attribute	Description
1	freeRunCounter	DINT	Auto		No	0	..	
2	incr	INT	Auto	Analog out 3	No	0	..	
3	period	REAL	%MD1.11	Onda quadra	No	0	..	
4	valueFilt	REAL	%MW1.8	Onda quadra	No	0	..	
5	tau	REAL	%MD1.9	Onda quadra	No	0	..	
6	valueRef	REAL	%MD1.10	Onda quadra	No	0	..	
7	khIncr	INT	Auto	Analog out 3	No	50	CONSTANT	

2. QUICK START

This chapter is a step-by-step tutorial that introduces you to the use of the LogicLab development environment.

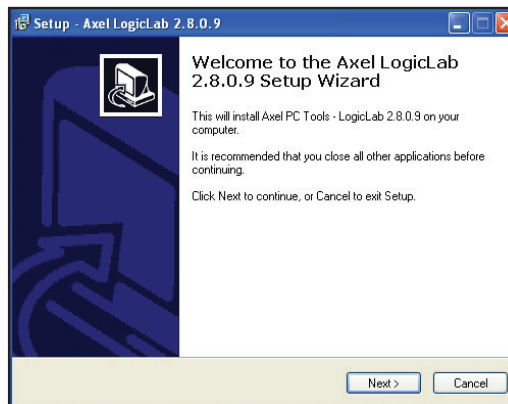
You will be guided through all the tasks you need to accomplish in order to run and monitor a simple PLC application.

The examples in this chapter refer to a virtual target, *VPLC1*, which is installed along with LogicLab during the setup procedure.

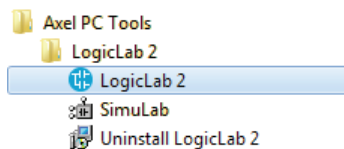
2.1 SETUP PROCEDURE

In order to install LogicLab, follow this simple procedure:

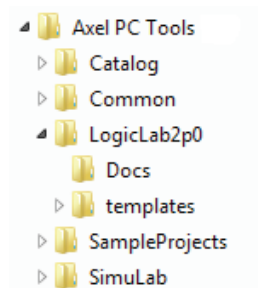
- 1) Close all the applications running on the system.
- 2) Run LogicLab's setup executable file and follow the instructions shown in the setup wizard.



The setup procedure adds the following *Start* menu.



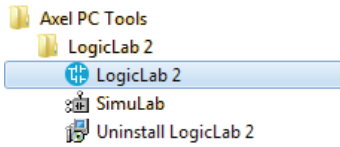
The LogicLab's folder in the file system has the following structure.



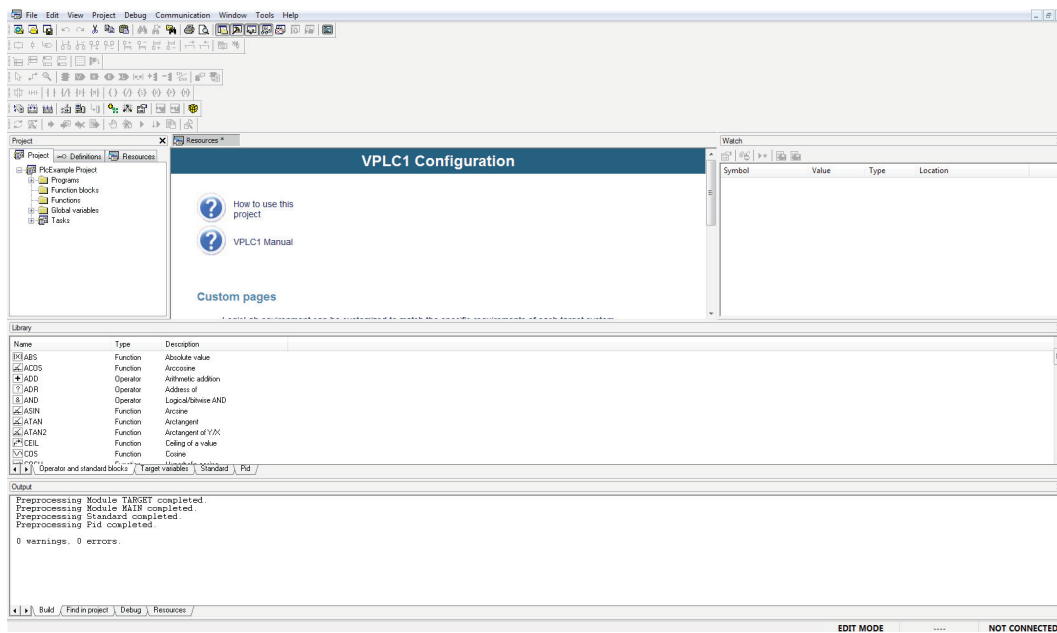
2.2 USING THE IDE

The following steps show you how to create, run and debug a simple LogicLab PLC program. The program will be written in the ST language and will be executed directly on your machine. A virtual target, *VPLC1*, has been installed by the setup procedure. *VPLC1* emulates a simple programmable controller provided with digital and analog I/O.

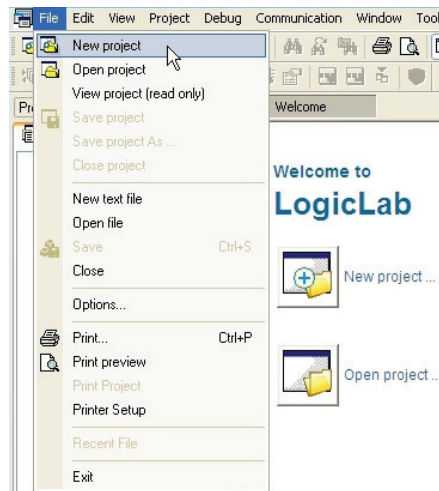
- 1) Launch LogicLab from the *Start* menu.



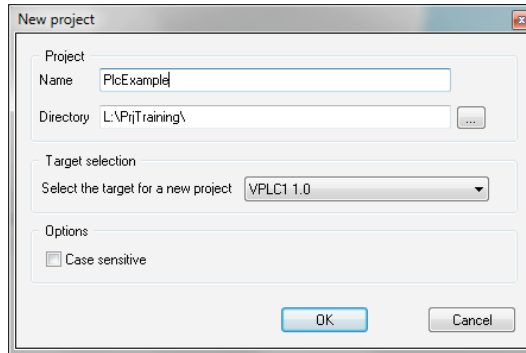
- 2) When LogicLab is started for the first time, not all windows and toolbars are shown. You can customize LogicLab's workspace through the *View* menu.



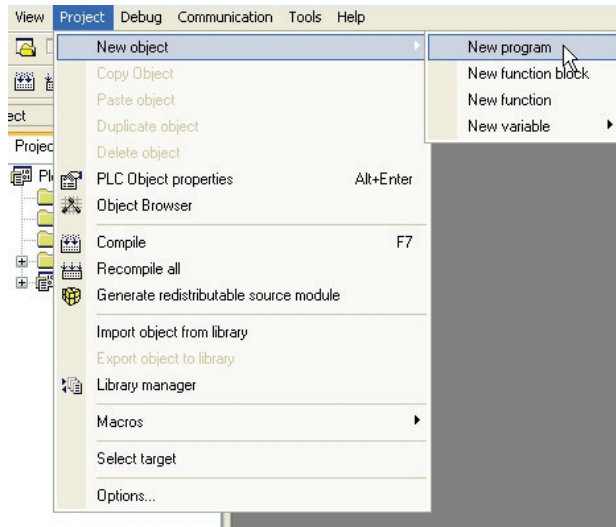
- 3) In order to create a new project, select the item *New project* from the menu *File*.



- 4) Insert the new project name (for example, *PlcExample*) and choose the destination folder, then press *OK*. *VPLC1* virtual target has been selected for this project. LogicLab's workspace looks now like in the picture shown at step 2 (that is, all operators and target variables are shown, preprocessing status log has been printed in the output window).

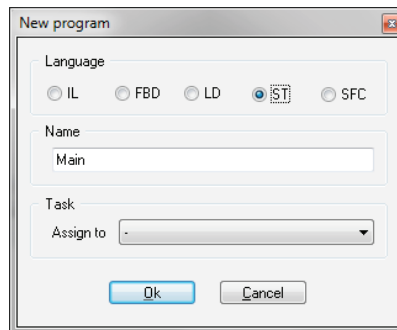


- 5) To add a new empty program to the project, select *New program* from the menu *Project > New object*.

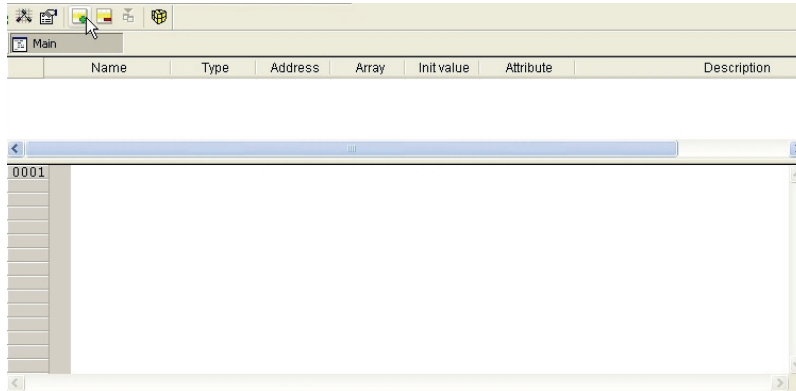


Choose the ST language, insert the new program name (for example, *Main*), and press *OK*.

It is also possible to assign here the program to one of the available tasks.



LogicLab shows now the editor window of the *Main program* object. On the top of the window there is the local variables editor. The variables editor allows to add, remove, copy and paste variable definitions.



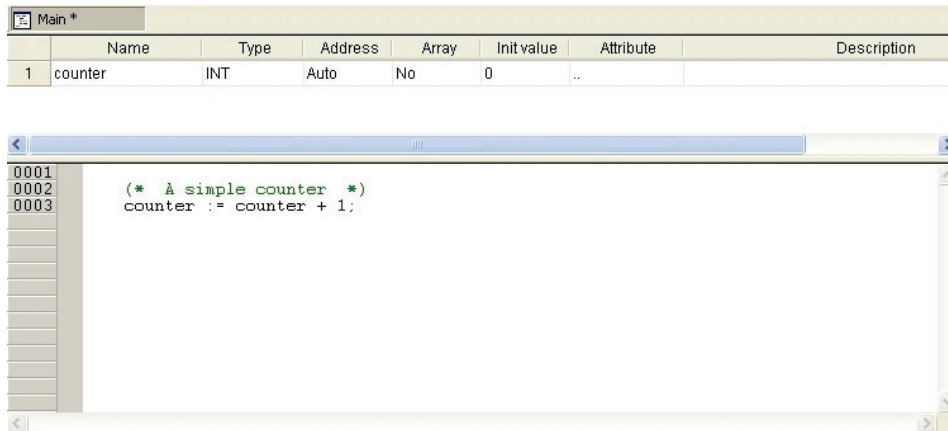
The textual source code editor is placed in the bottom part of the window.

- 6) Insert a new local variable by pressing the *Insert record* button in the *Project* toolbar. Alternatively, you can use the *Insert* option of the *Variables* menu.

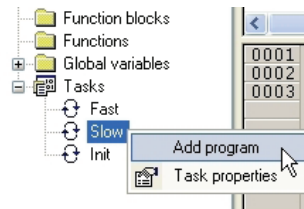


- 7) Edit the name (for example, *Counter*) and the type of the variable. A double-click in the type field opens the type selection dialog. Choose the INT data type.

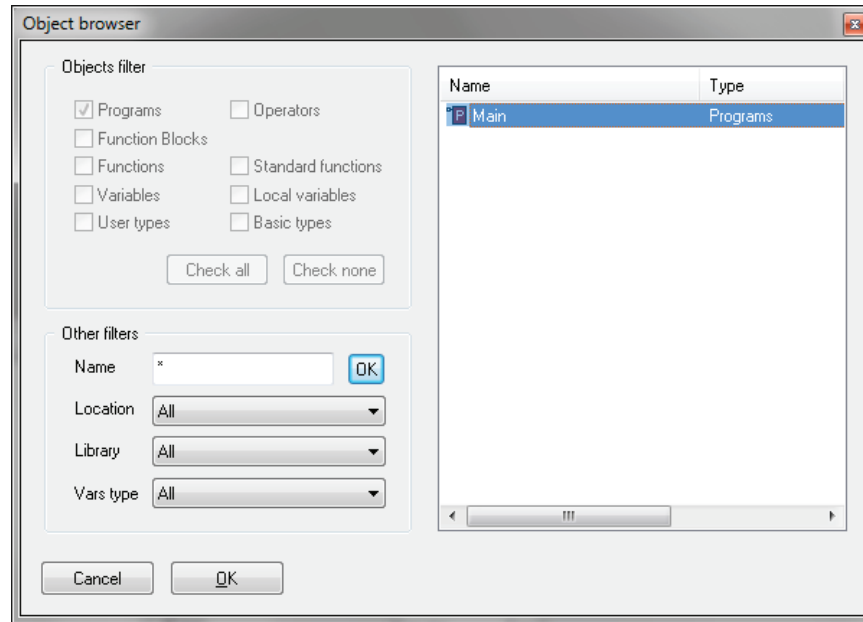
- 8) Implement a simple counter in ST language, as shown below.



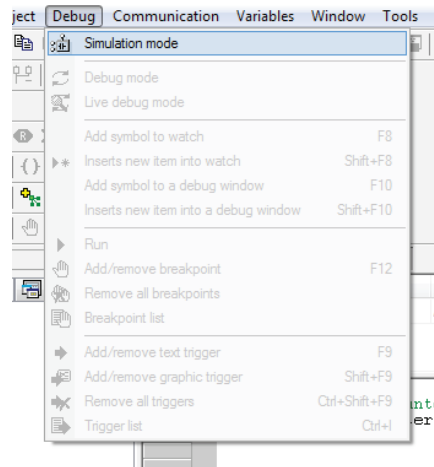
- 9) In order to be executed, the program *Main* has to be assigned to a task. In the *Workspace* window, right-click on the *Slow* task and select the *Add program* option. The *Object browser* window pops up.



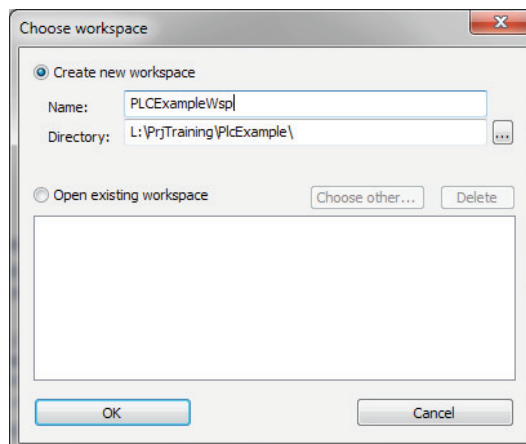
10) Select the *Main* program and click *OK*. The picture below shows the *Workspace* window after the successful completion of the operation.



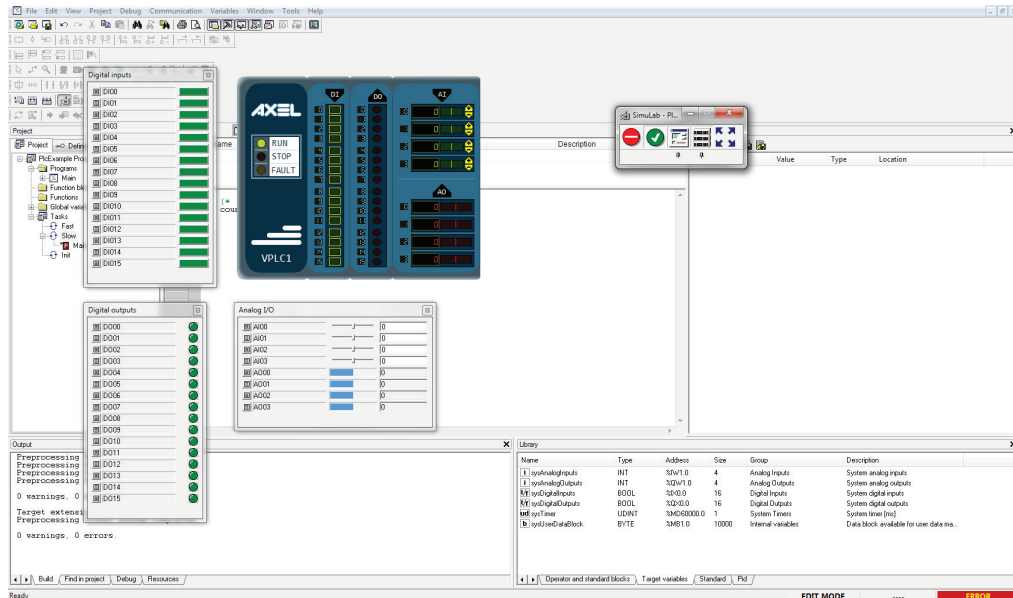
11) Select *Simulation mode* from *Debug* menu.



Name your *VPLC1* workspace as *PLCExampleWsp*, then press *OK*.



- 12) *VPLC1* is a software which emulates a simple programmer controller provided with digital and analog I/O. It is possible to interact with simulator as explained in SimuLab manual or briefly in *Resources* tab.



The status of the *VPLC1* I/O is available in LogicLab as the *Target variables* listed in the *Library* window.

- 13) With LogicLab connect to *VPLC1* through the apposite button on the toolbar.



- 14) Press *F7* (or select the menu option *Project - Compile*) to compile the application. Verify that the compilation ends with *0 warning - 0 errors*, otherwise correct the errors and compile again.

- 15) Download the code by means of the appropriate toolbar command.

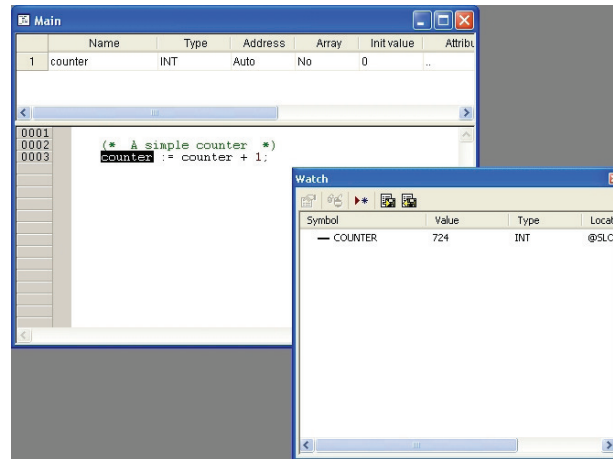


PLC is now running on *VPLC1*. You can see the LogicLab bottom bar indicating *Source ok - Connected*.



- 16) Open the *Watch* window through the *View* menu, then select the variable in the text editor with a double-click and drag and drop it into the *Watch* window.

The *Watch* window should display the value of the *Counter* variable which should increment continuously.



You are at the end of this very simple but complete (edit-compile-download-debug) working session with LogicLab.

Further steps you should take in order to acquire a deeper knowledge of the tool include the use of the other languages, function & function blocks, global variables, libraries, real time debugging and so on. Refer to the following sections of this manual to obtain the necessary information.

2.3 A MORE COMPLEX EXAMPLE

In the *SampleProjects\Sample* subdirectory of your LogicLab installation you can find a more complex *VPLC1* example, which offers an overview of IEC 61131-3 languages.

To open the project use the *File - Open project* menu option and open *VPLC1Sample.ppjs* or select it from *Example projects* list in the welcome page.





3. USING THE ENVIRONMENT

This chapter shows you how to deal with the many UI elements LogicLab is composed of, in order to let you set up the IDE in the way which best suits to your specific development process.

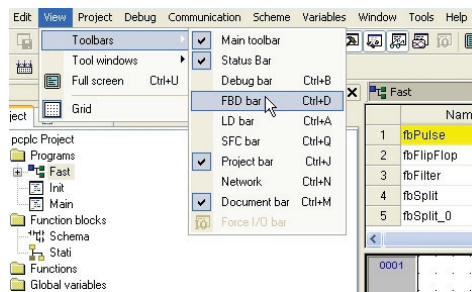
3.1 LAYOUT CUSTOMIZATION

The layout of LogicLab's workspace can be freely customized in order to suit your needs. LogicLab takes care to save the layout configuration on application exit, in order to persist your preferences between different working sessions.

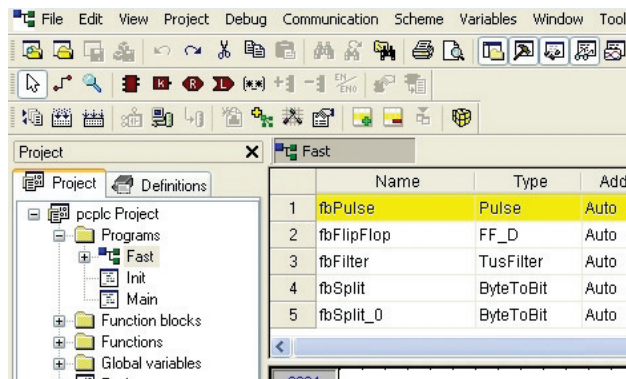
3.2 TOOLBARS

3.2.1 SHOWING/HIDING TOOLBARS

In details, in order to show (or hide) a toolbar, open the *View>Toolbars* menu and select the desired toolbar (for example, the *Function Block Diagram* bar).

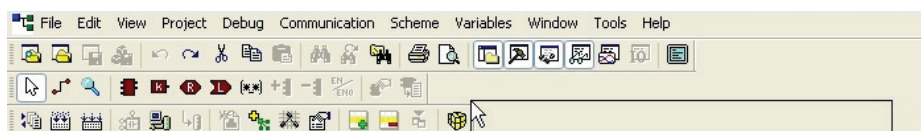


The toolbar is then shown (hidden).



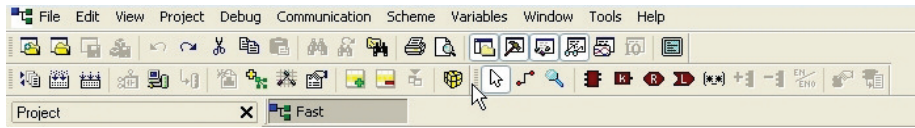
3.2.2 MOVING TOOLBARS

You can move a toolbar by clicking on its left border and then dragging and dropping it to the destination.

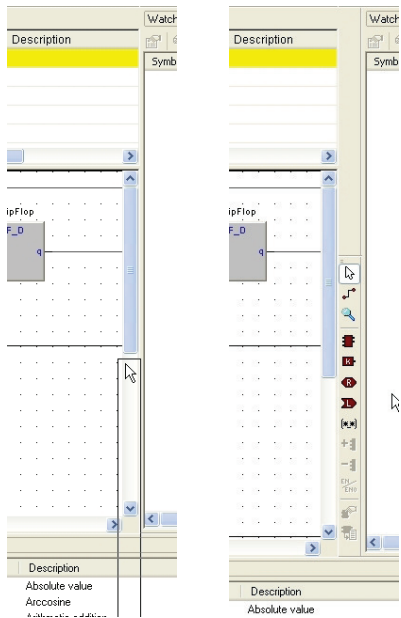


The toolbar shows up in the new position.

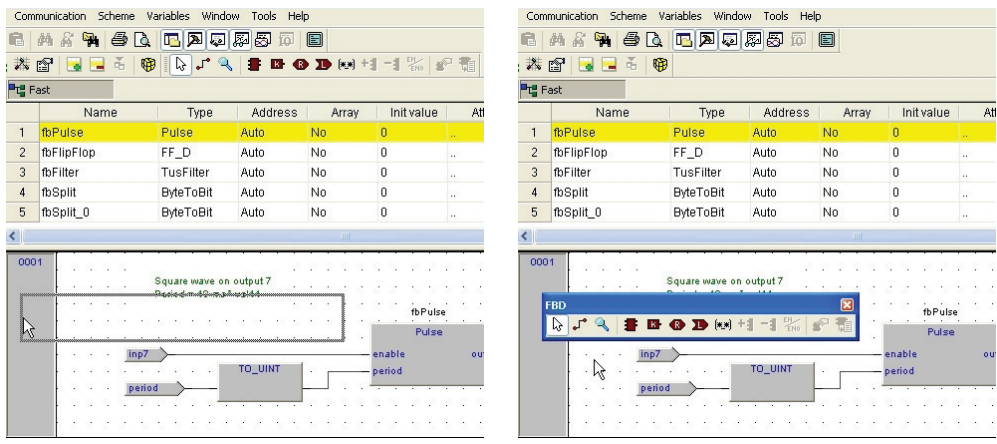




You can change the shape of the toolbar, from horizontal to vertical, either by pressing the *Shift* key or by moving the toolbar next to the vertical border of any window.



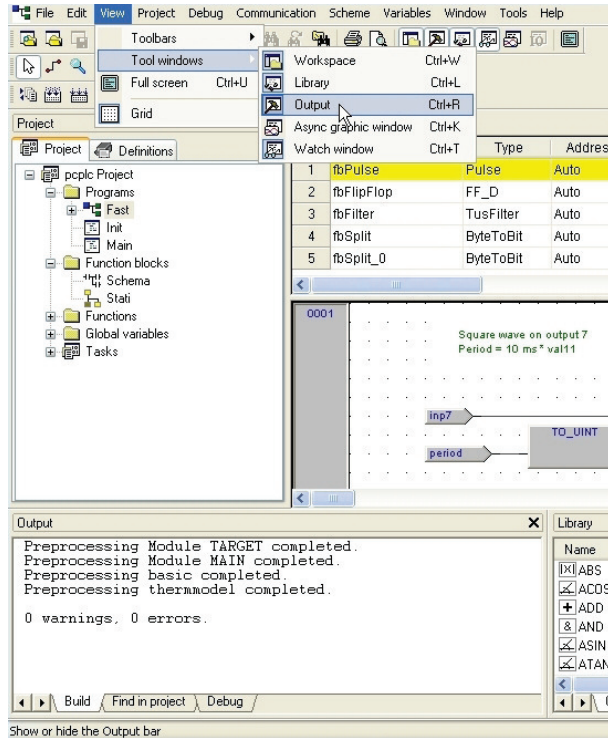
You can also make the toolbar float, either by pressing the *CTRL* key or by moving the toolbar away from any window border.



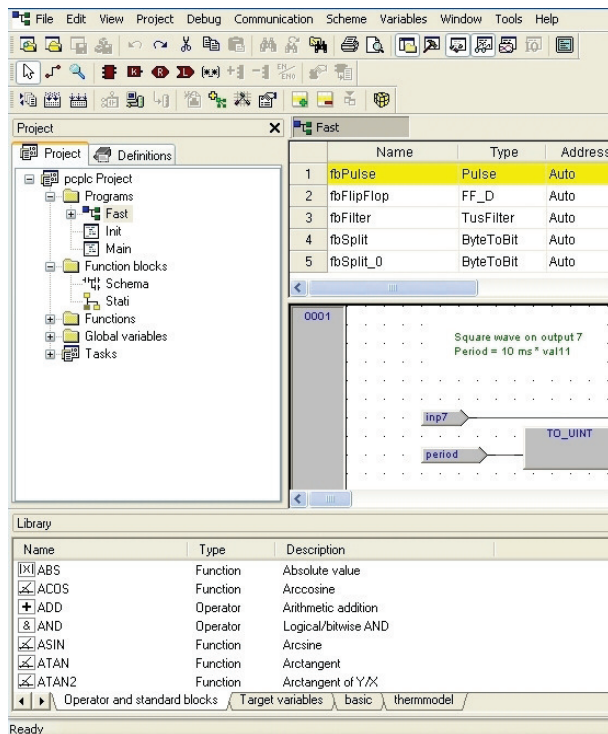
3.3 DOCKING WINDOWS

3.3.1 SHOWING/HIDING TOOL WINDOWS

The *View>Tool* windows menu allows you to show (or hide) a tool window (for example, the *Output* window).

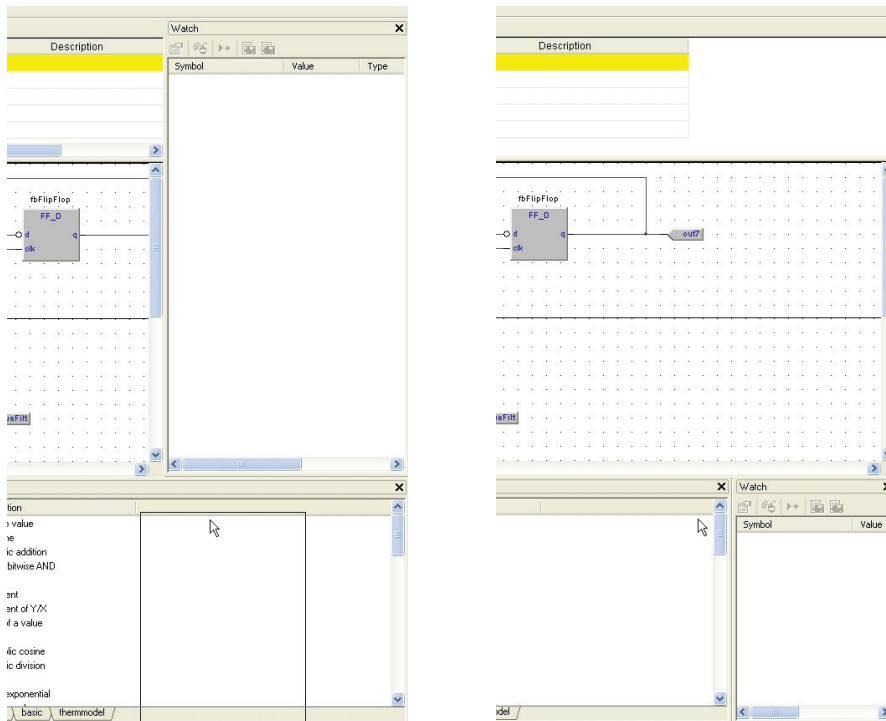


The tool window is then shown (hidden).

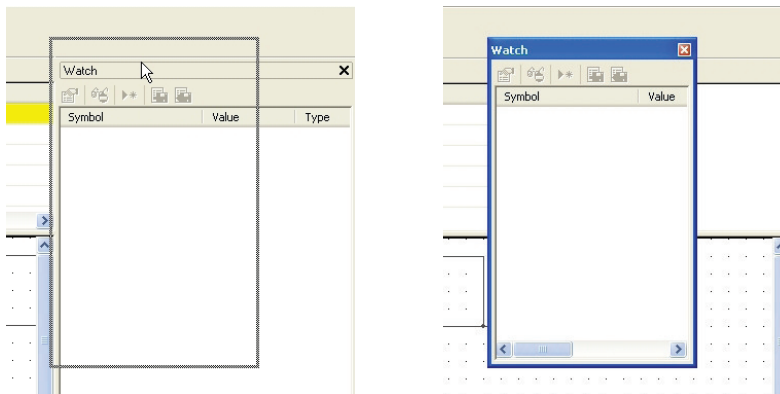


3.3.2 MOVING TOOL WINDOWS

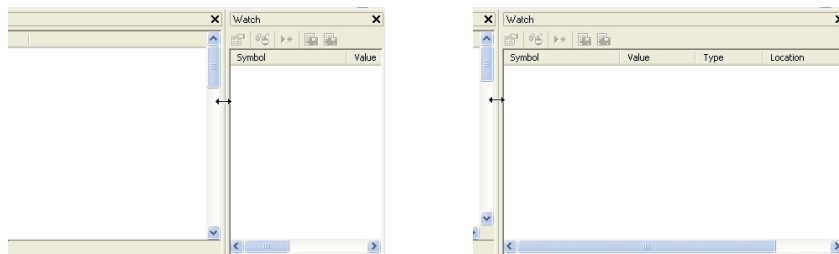
In order to move a tool window, click on its name (at the top of the window) and then drag and drop it to the destination.



You can make the tool window float, by double-clicking on its name, or by pressing the *CTRL* key, or by moving the tool window away from the main window borders.



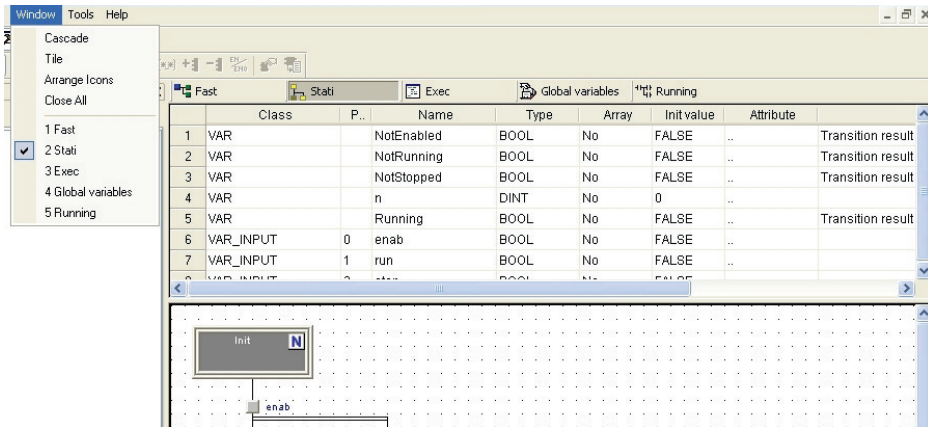
A tool window can be resized by clicking-and-dragging on its border until the desired size is reached.



3.4 WORKING WITH WINDOWS

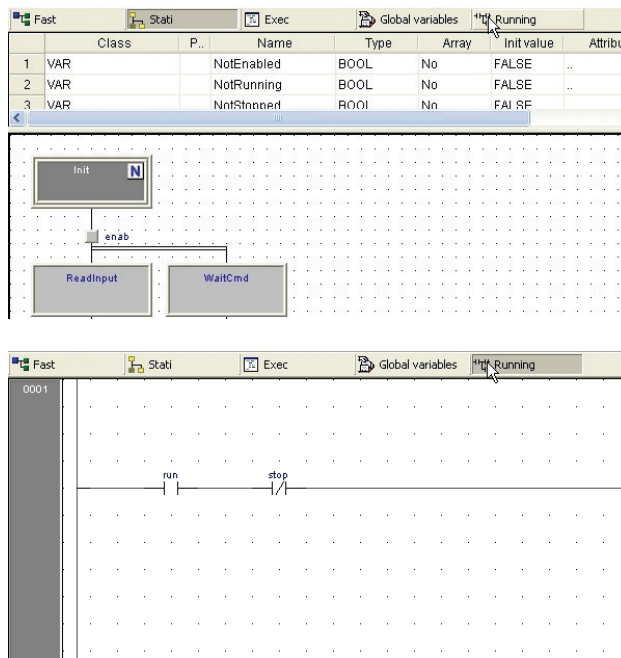
LogicLab allows to open many source code editors so that the workspace could get rather messy.

You can easily navigate between these windows through the *Document* bar and the *Window* menu.



3.4.1 THE DOCUMENT BAR

The *Document* bar allows to switch between all the currently open editors, simply by clicking on the corresponding name.

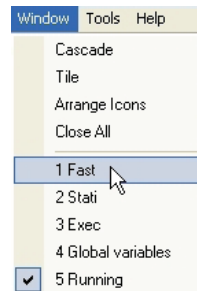


You can show or hide the *Document* bar with the menu option of the same name in the menu *View>Toolbars*.



3.4.2 THE WINDOW MENU

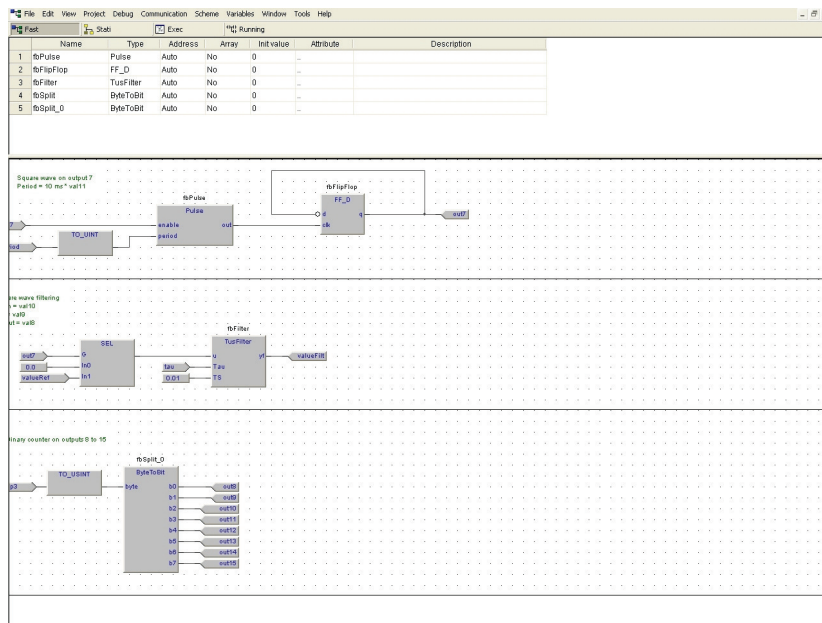
The *Window* menu is an alternative to the *Document* bar: it lists all the currently open editors and allows to switch between them.



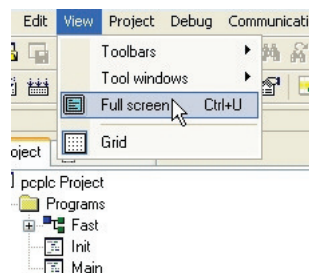
Moreover, this menu supplies a few commands to automate some basic tasks, such as closing all windows.

3.5 FULL SCREEN MODE

In order to ease the coding of your application, you may want to switch on the full screen mode. In full screen mode, the source code editor extends to the whole working area, making easier the job of editing the code, notably when graphical programming languages (that is, LD, FBD, and SFC) are involved.



You can switch on and off the full screen mode with the *Full screen* option of the menu *View* or with the corresponding command of the *Main* toolbar.



3.6 ENVIRONMENT OPTIONS

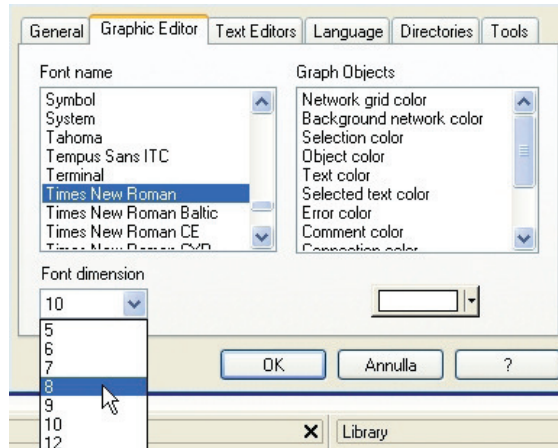
If you click *Options...* in the *File* menu, a multi-tab dialog box appears and lets you customize some options of LogicLab.

General

Autosave: if the *Enable Autosave* box is checked, LogicLab periodically saves the whole project. You can specify the period of execution of this task by entering the number of minutes between two automatic savings in the *Autosave interval* text box.

Graphic Editor

This panel lets you edit the properties of the LD, FBD, and SFC source code editors.

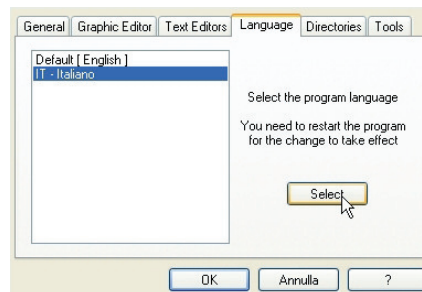


Text Editors

Language

You can change the language of the environment by selecting a new one from the list shown in this panel.

After selecting the new language, press the *Select* button and confirm by clicking *OK*. This change will be effective only the next time you start LogicLab.

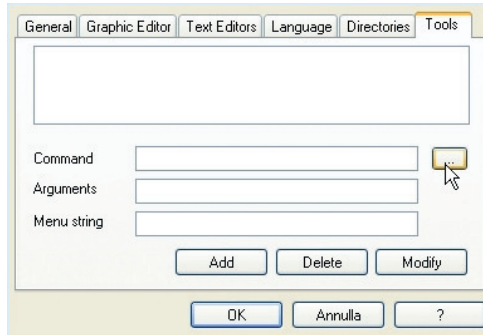


Tools

You can add up to 16 commands to the *Tools* menu. These commands can be associated with any program that will run on your operating system. You can also specify arguments for any command that you add to the *Tools* menu. The following procedure shows you how to add a tool to the *Tools* menu.



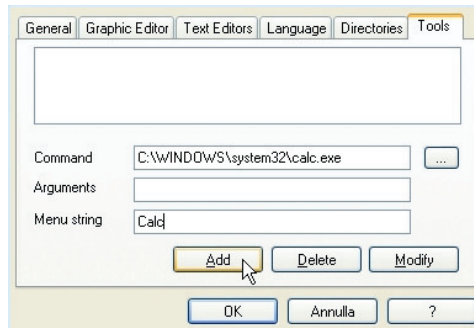
- 1) Type the full path of the executable file of the tool in the *Command* text box. Otherwise, you can specify the filename by selecting it from Windows Explorer, which you open by clicking the *Browse* button.



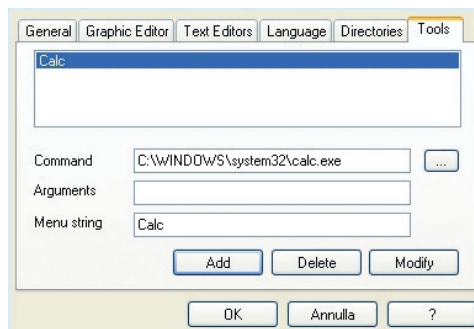
- 2) In the *Arguments* text box, type the arguments - if any - to be passed to the executable command mentioned at step 1. They must be separated by a space.
- 3) Enter in *Menu string* the name you want to give to the tool you are adding. This is the string that will be displayed in the *Tools* menu.
- 4) Press *Add* to effectively insert the new command into the suitable menu.
- 5) Press *OK* to confirm, or *Cancel* to quit.

For example, let us assume that you want to add *Windows calculator* to the *Tools* menu:

- Fill the fields of the dialog box as displayed.



- Press *Add*. The name you gave to the new tool is now displayed in the list box at the top of the panel.



And in the *Tools* menu as well.

