



Beremiz User Manual, by LOLITECH

*Pre-Alpha preview
not for production use*

Copyright (c) 2008 LOLITECH.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being « About The Authors », with the Front-Cover Texts being « Beremiz User Manual, by Lolitech », and with no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

Table des matières

A - About The Authors.....	3
1° LOLITECH.....	3
2° University of Porto.....	3
3° Motivations.....	3
B - Overview of features.....	4
1° PLC builder GUI.....	5
2° PLCopen Editor.....	6
3° IEC-61131-3 compiler.....	8
4° CANopen Stack.....	9
5° SVG HMI toolkit.....	9
C - Start with Beremiz.....	10
1° Launch Beremiz.....	10
2° Create a new project.....	11
3° Open an existing project.....	12
4° Project's tool bar.....	12
5° Configure your project.....	13
a)Windows projects.....	13
b)Linux projects.....	13
6° PLCopen Editor.....	14
a)PlcOpen Editor Interface:.....	14
b)Add / Delete a POU :.....	14
c)Toolbars description :.....	15
7° Plugins.....	16
a)Adding a plugin to your project	17
b)CANopen plugin.....	18
c)Svgui plugin.....	20
8° The DEMO.....	21
a)Start the demo.....	21
b)Build the project	21
c)Run the PLC	21
d)Stop the PLC	22
e)Clean The PLC	22
D - Project Status.....	23
E - FAQ.....	24

A - About The Authors

Beremiz is the result of a long development effort, taking roots at LOLITECH in Saint-Dié-des-Vosges, France, and in University of Porto, Portugal.

1° LOLITECH

Focussed on Free & Open Source Software for automation, this company have been funded by the authors of the CanFestival project in 2005.

Today, LOLITECH not only develop, maintain and support CanFestival CANopen stack, but also extend his offer to Beremiz Open Source automation framework.

2° University of Porto

Mario de Sousa, working for the “Faculdade de Engenharia da Universidade do Porto“ developed the original IEC-61131-3 compiler, initially part of the MatPLC project. Thanks to him, Beremiz project embeds a IEC-61131-3 compiler that produce platform-independent C code.

3° Motivations

World of Automation is still exempt of Open Source software. As consequences :

- Despite of open standards such as IEC 61131-3, PLCOpen and CanOpen control engineers cannot easily transfer programs between vendor solutions.
- PLC application sustainability directly depend on the PLC programming workbench software provider company will.
- Teaching IEC-61131-3 involves acquiring expensive workbench licenses for PLC programming. Generally, students cannot use the software on their laptop for homework.
- Operating safety may hardly be proven, as source code of PLC runtime and compilers are closed.

We intend to solve those problems with our open source approach.

B - Overview of features

Beremiz is an Open Source framework for automation.

With Beremiz, you can :

- Automate everything.
- Take any processor into a PLC.
- Program once, run anywhere.
- Create PLC controled customizable HMIs.
- Conform to standards.
- Avoid vendor lock.

Beremiz relies on these sub-projects:

1. [PLCOpen Editor](#) : Multi-platform automation IDE
2. [MatPLC's IEC compiler](#) : IEC 61131-3 compiler
3. [CanFestival](#) : CANOpen interface to physical I/O
4. [SVGUI](#) : automated HMI tool, based on SVG

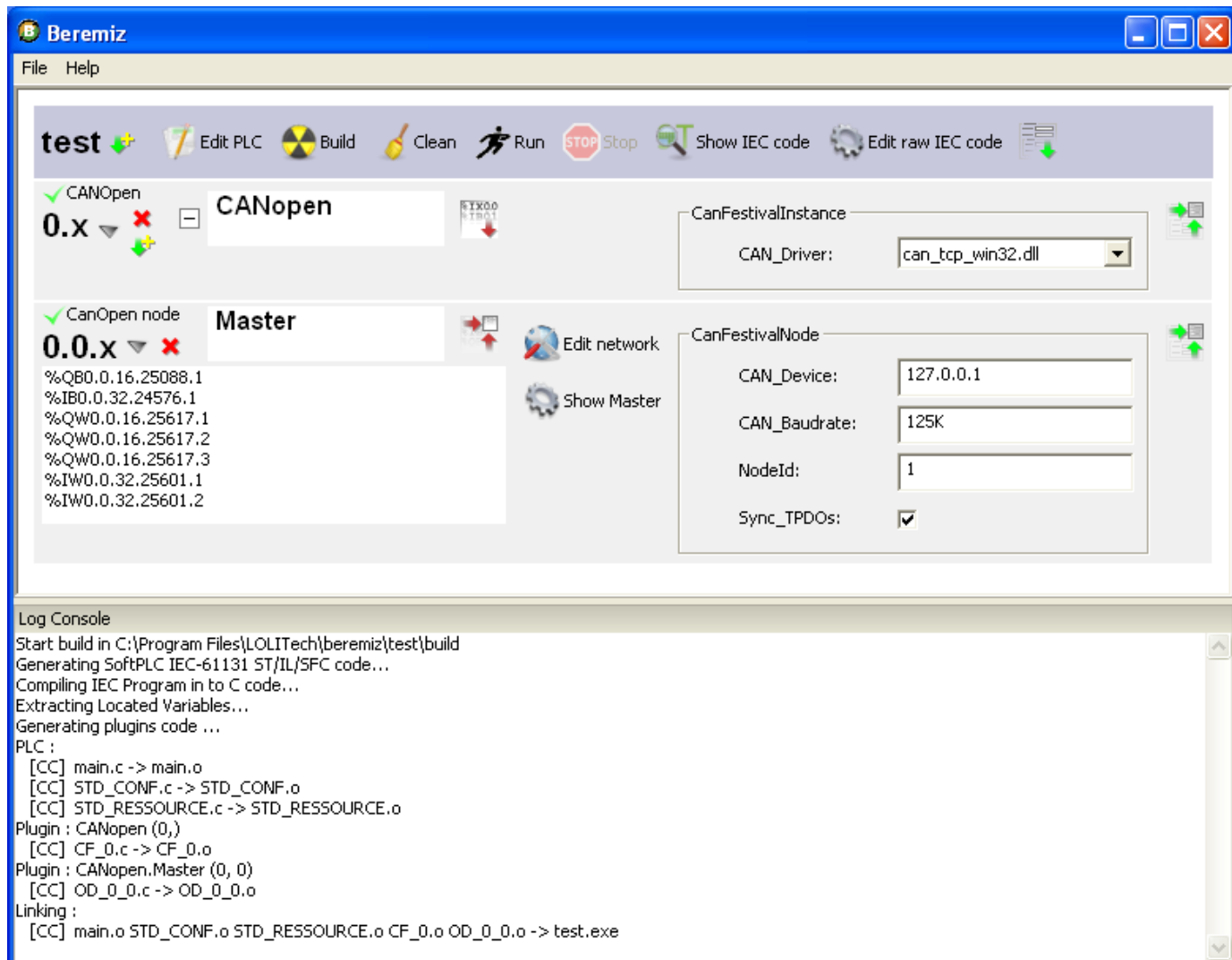


Beremiz user edits programs with the PLCOpen editor, compiles them into C with the IEC-61131-3 to C compiler, and can execute this code along CanFestival CANOpen stack to produce a CANOpen PLC.



1° PLC builder GUI

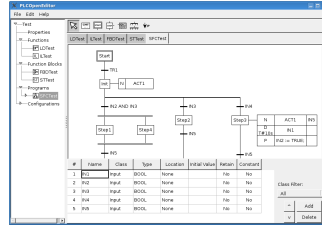
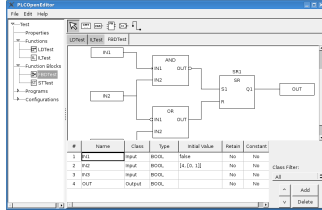
This is the GUI that allows PLC programmers to create new projects, define and map physical I/O to Directly represented IEC-61131 variables (for example : %IX0.1.2). More details in [6° Plugins](#)



2° PLCopen Editor

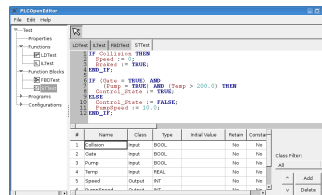
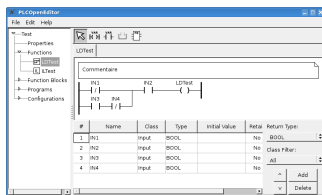
The PLCopen Editor saves and loads PLC projects accordingly to [PLCopen TC6-XML Schema](#).

Function Block Diagram - FBD Sequential Function Chart - SFC



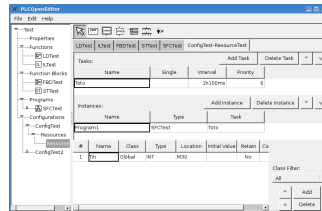
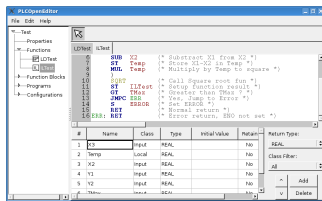
Ladder Diagram - LD

Structured Text - ST

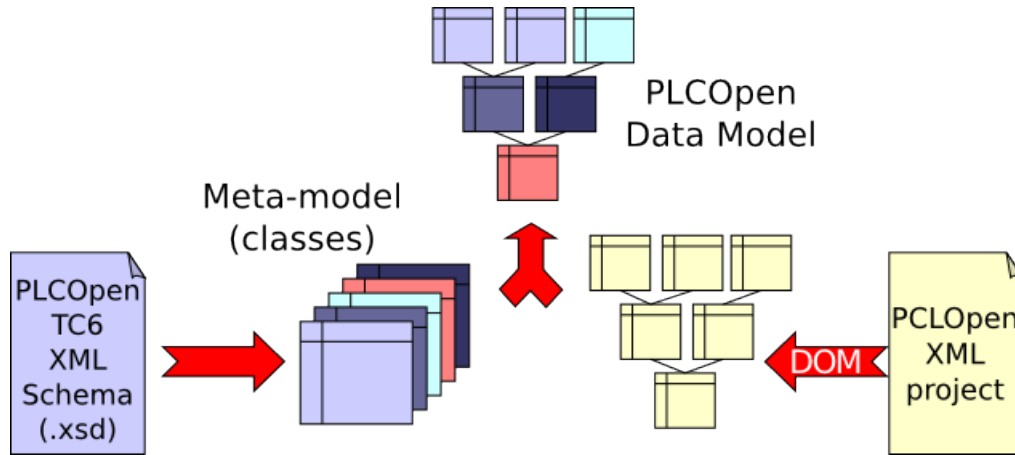


Instruction List - IL

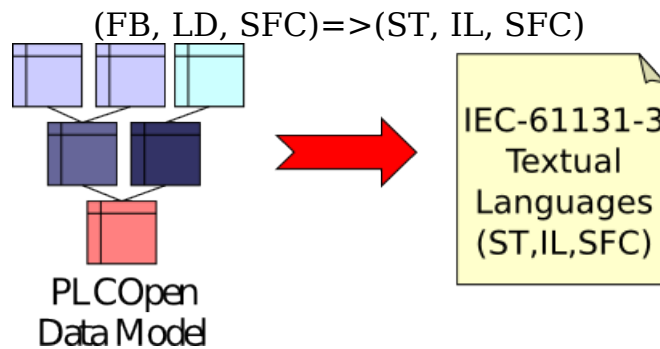
Configurations, Resources, Tasks



Data-model is based on the official TC6-XML XML Schema. The official .xsd file is used at startup to create a kind of meta model, that define relations between objects inside the PLCOpen model. Thanks to this feature, PLCOpen Editor can also be used as a PLCOpen TC6-XML validator.



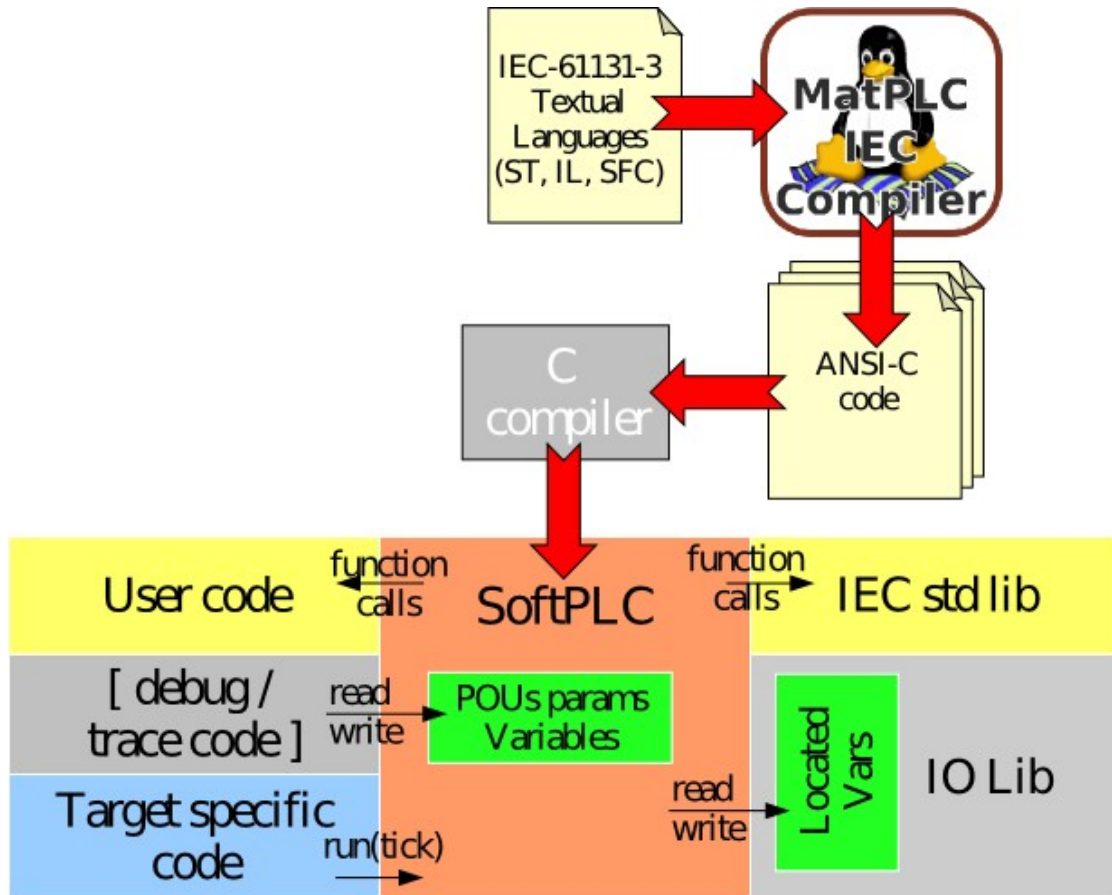
PLCOpen editor has built-in export filter that convert graphical languages to their equivalent textual form :



3° IEC-61131-3 compiler

EC-61131-3 compiler compiles ST/IL/SFC code into ANSI-C code.

- Project started in 2002 by Mario de Sousa (U-Porto).
- All POU parameters and variables are accessible through nested C structs
- Located variables are declared as extern C variables



4° CANopen Stack

[CanFestival](#) is an OpenSource CANOpen framework, licensed with GPLv2 and LGPLv2.

With CanFestival you CAN :

- Turn any μ C or PC into a CANOpen node
- Edit Object Dictionary and EDS files
- Use any CAN interface type
- Link with proprietary code
- Teach or learn CANOpen

[CanFestival](#) focuses on providing an ANSI-C platform independent CANOpen stack that can be implemented as master or slave nodes on [PCs](#), [Real-time IPCs](#), and [Microcontrollers](#).

More details on <http://www.canfestival.org>

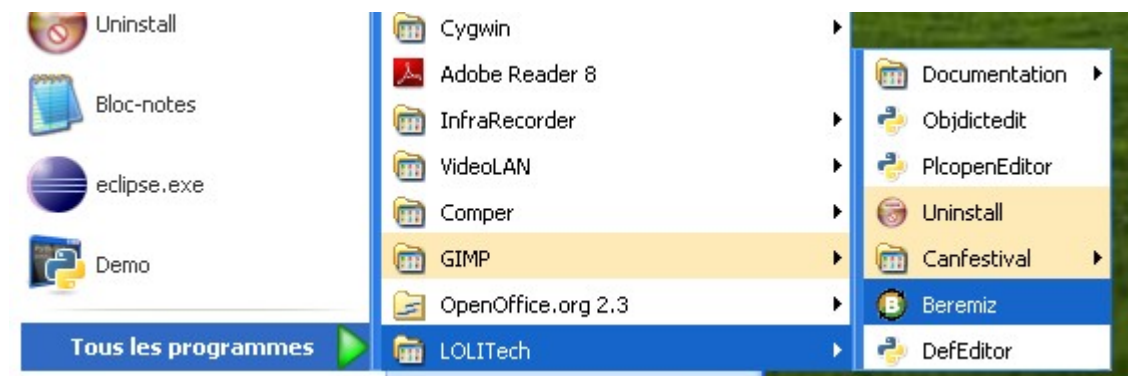
5° SVG HMI toolkit

PRE-ALPHA : Broken, documentation to be written.

C - Start with Beremiz

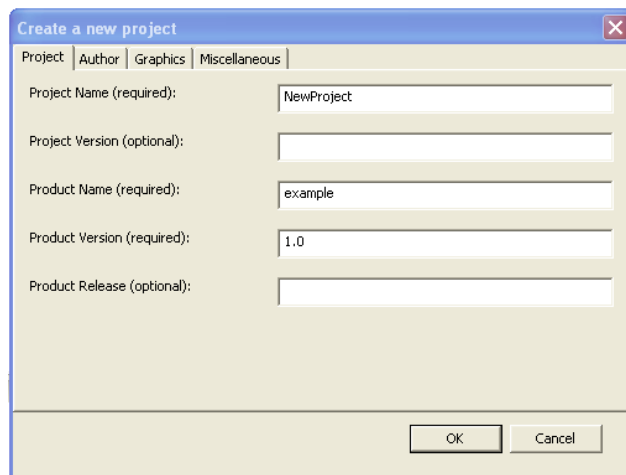
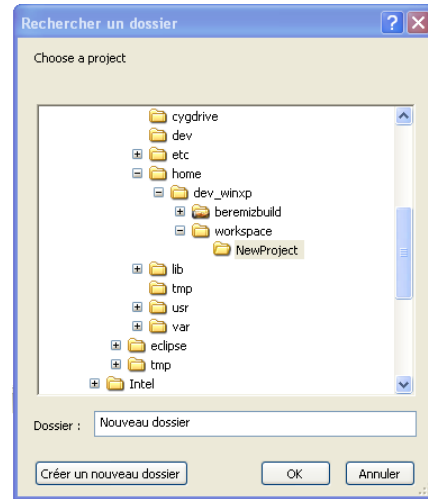
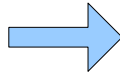
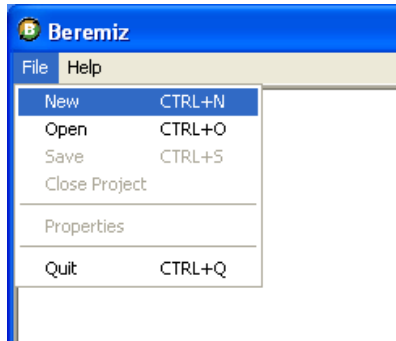
1° Launch Beremiz

Beremiz installer provides many shortcuts in the Windows' "Start Menu". So browse your start menu as the following and launch Beremiz :



2° Create a new project

Beremiz project needs an empty directory to be created. This directory will contain all the necessary files to develop and build your PLC.

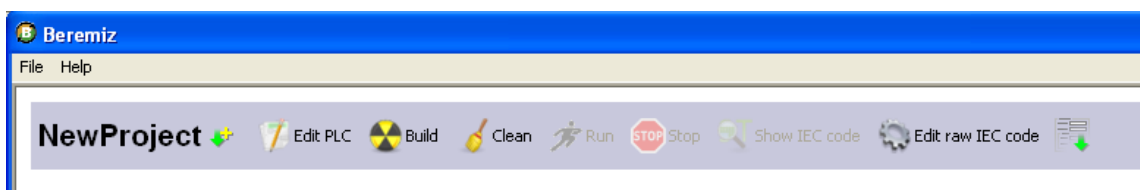


Choose where to create project, and create your project directory with project name

Define project's properties. It will be created only if all required fields are filled. Others are optional.



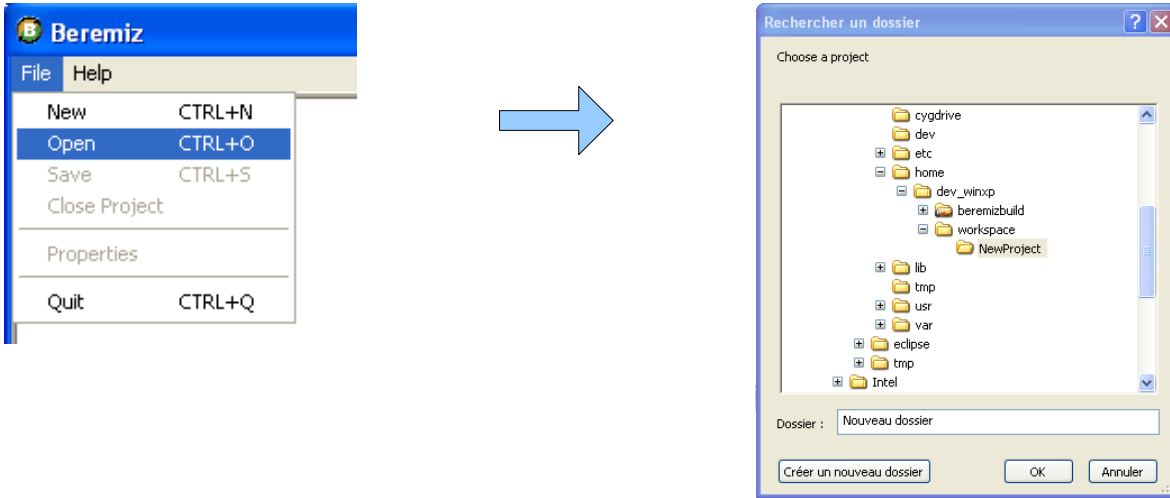
Once the project is created, the project's toolbar appeared



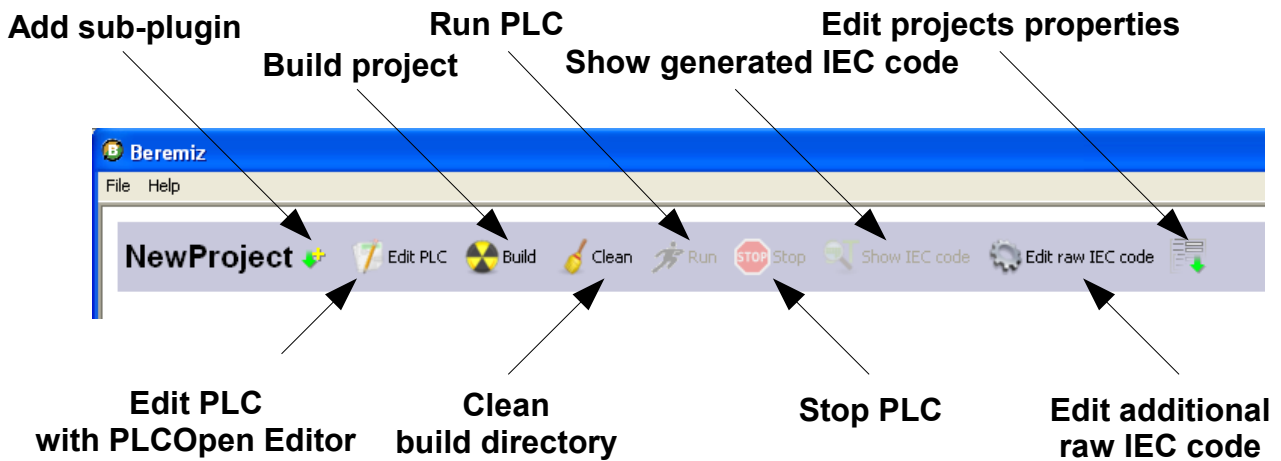
Your project directory now contains two files:

- plc.xml: PLC project file
- beremiz.xml: Beremiz default properties

3° Open an existing project



4° Project's tool bar

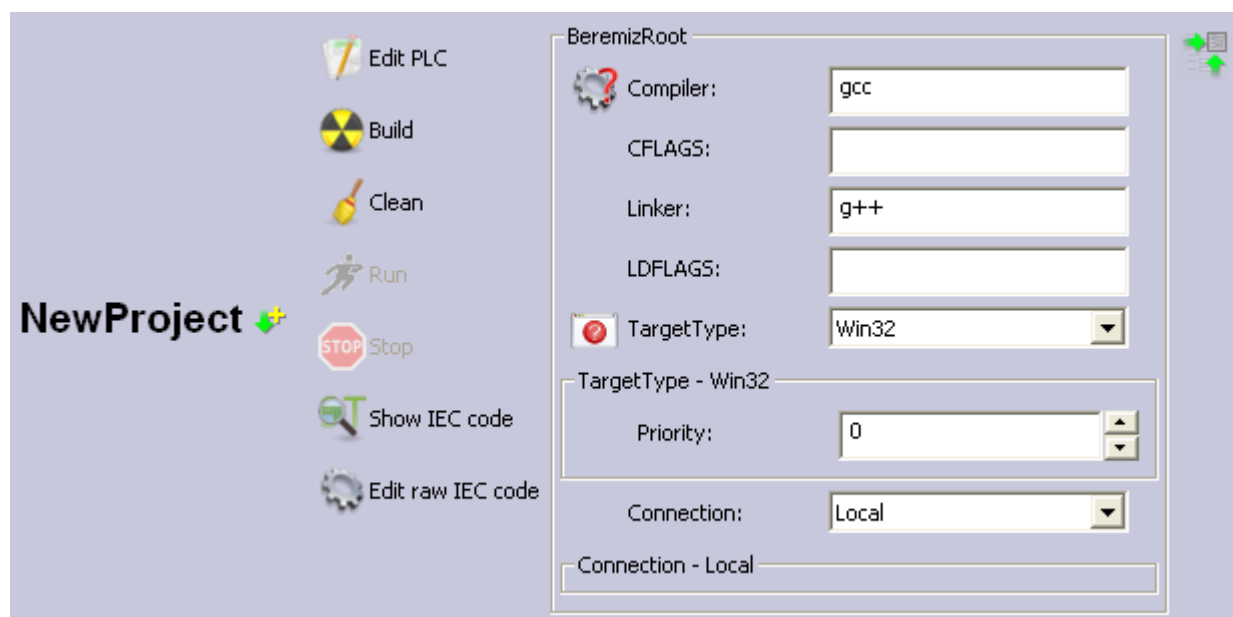


5° Configure your project

a) Windows projects

Windows Beremiz installer provides embedded MinGW GCC compiler, but you can specify your own GCC compiler version in the Beremiz properties.

To access to this parameters, click on  .



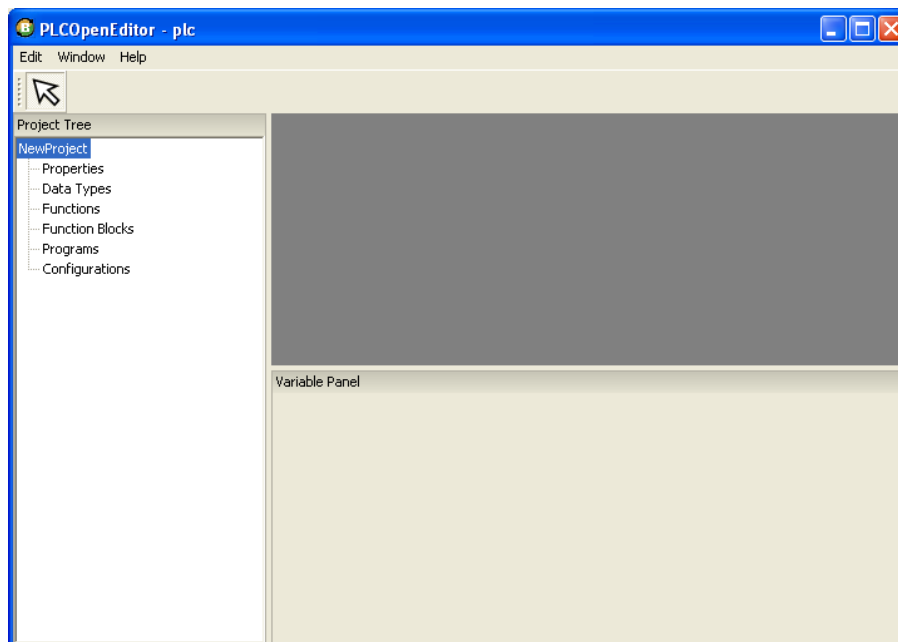
You can specify target type, compiler executable, optional compilation flags and linker flags. PRE-ALPHA : Priority has no effect.

b) Linux projects

PRE-ALPHA : To be written

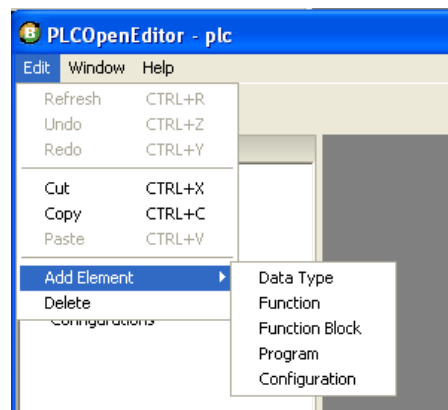
6° PLCOpen Editor

a) PlcOpen Editor Interface:

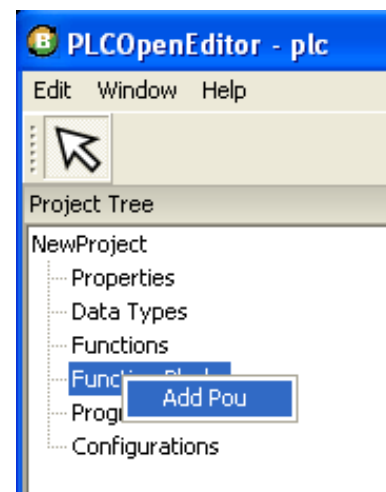
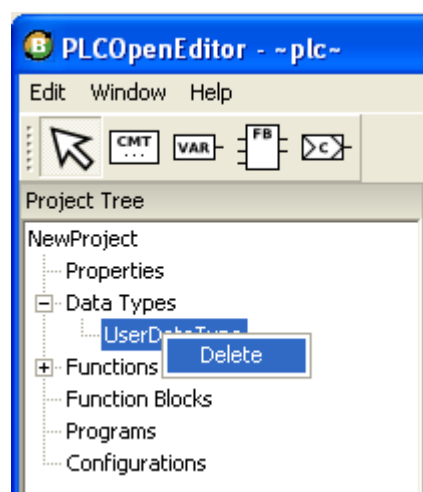


b) Add / Delete a POU :

Edit menu :

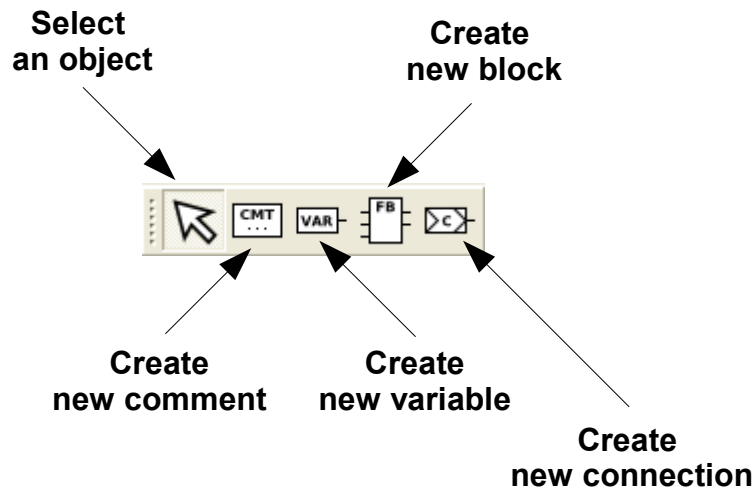


Right click:

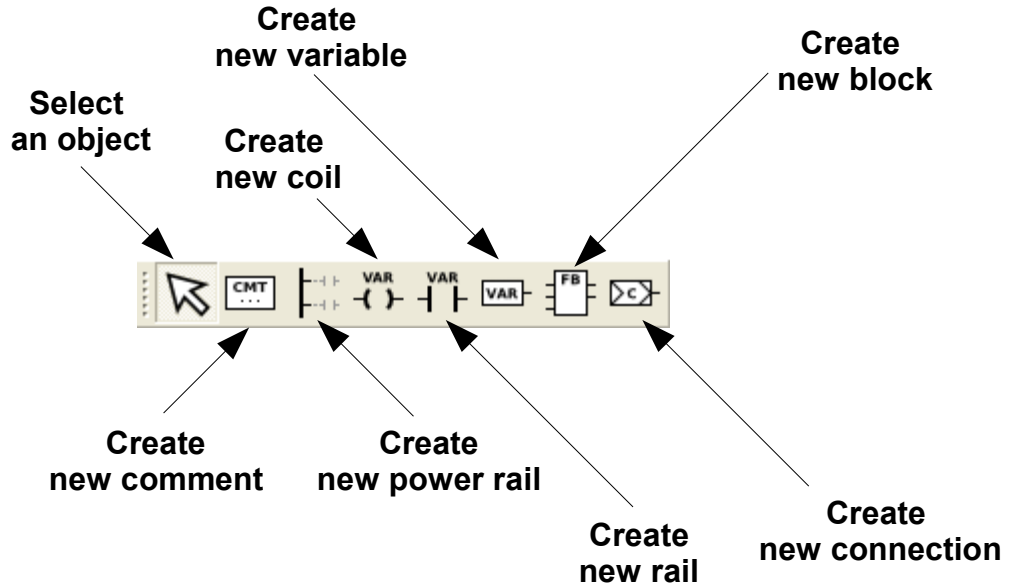


c) Toolbars description :

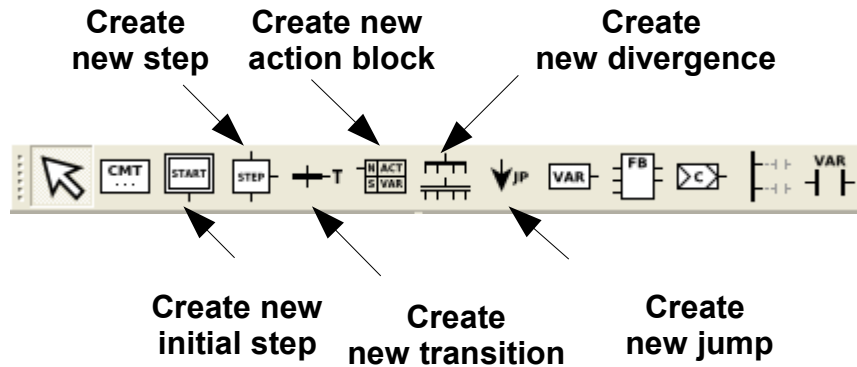
Function Diagram :



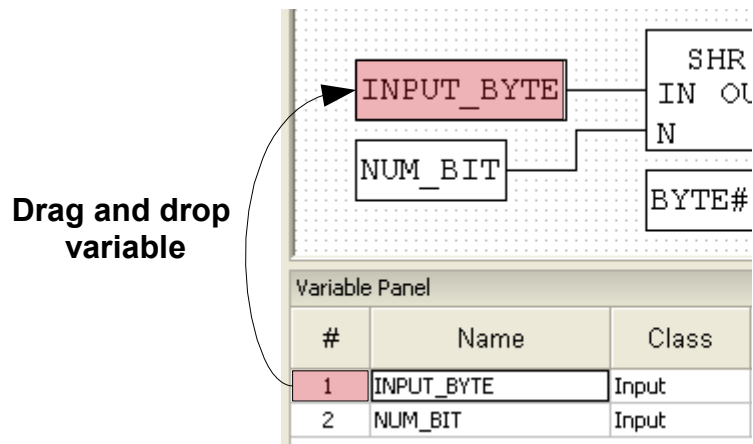
Ladder Diagram :



SFC :



You can drag and drop variables from the variable panel directly on the grid.



7° Plugins

Physical input and outputs variables are hierarchically organized in a plugin tree.

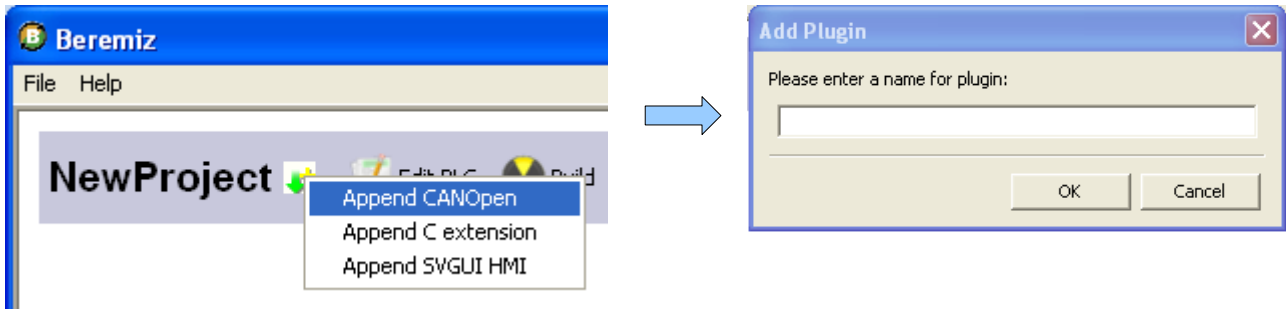
Each plugin is associated with a range of IEC-61131-3 “directly represented variables”. Nested plugins are mapped to sub-ranges. As an example:

	Plugin Location	Possible variable
CANopen plugin :	0	
First CANopen Network :	0.0	%IX0.0.3.323.1
Second CANopen Network :	0.1	%IX0.1.3.323.1
HMI plugin :	1	
First Display :	1.0	%IX1.1.3.323.1
Second Display :	1.1	%IX1.1.3.323.1

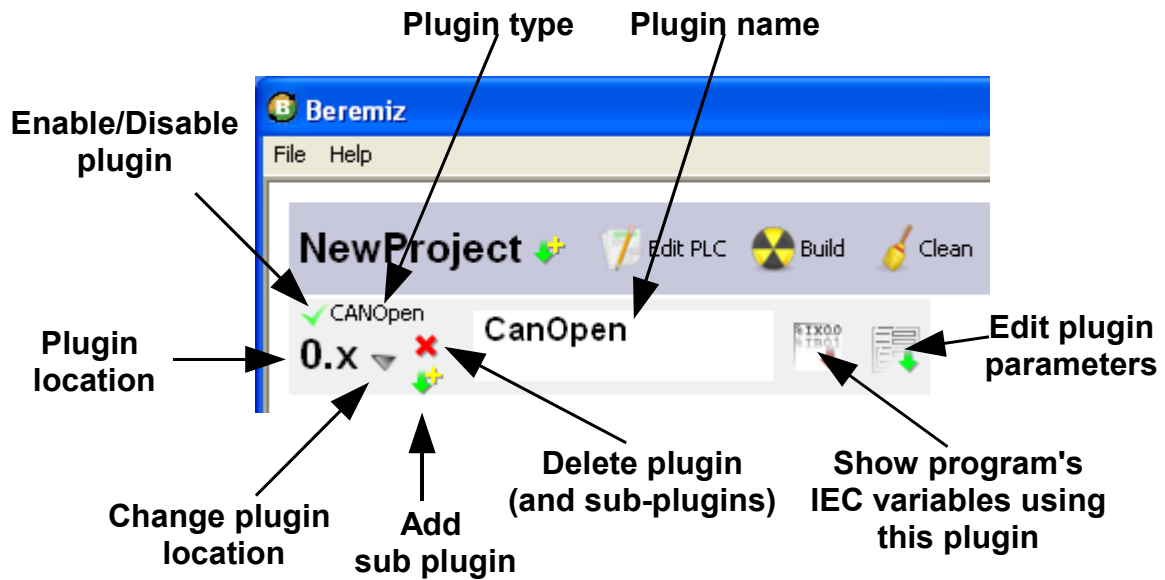
During build, “directly represented variables” declared in PLC program are dispatched in plugin tree according to their location, and consumed by plugins to produce corresponding C code.

a) Adding a plugin to your project

Add a plugin to your project :



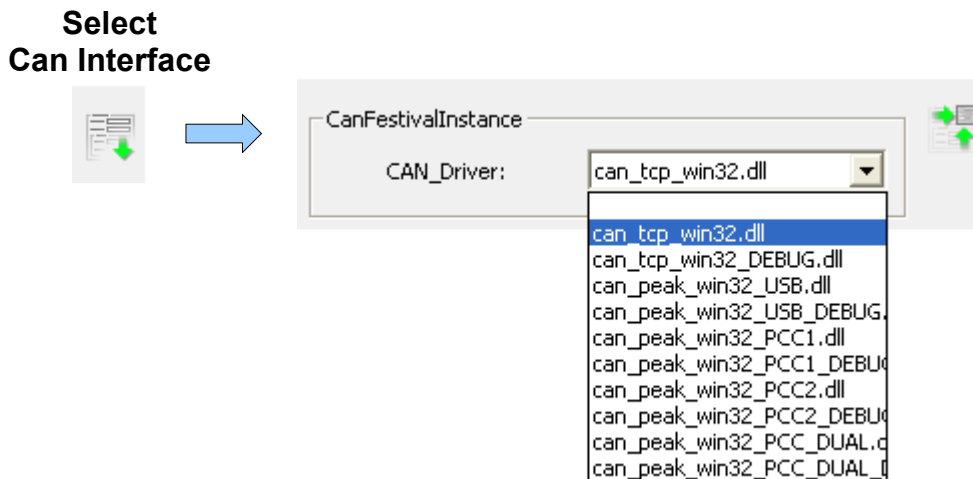
Plugin appears now in the project's tree :



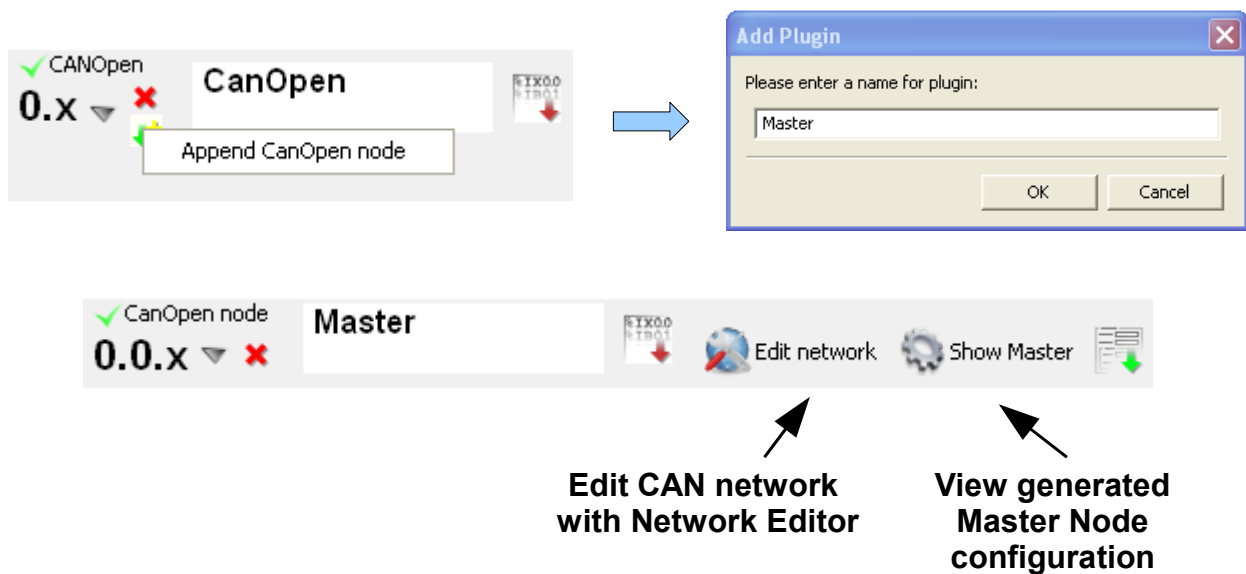
b)CANopen plugin

Thanks to this plugin, your PLC can act as a CanOpen Master and control Slave Nodes on different CAN buses. You have to choose the CAN driver corresponding to your Hardware. PRE-ALPHA: At that time only Peak-System CAN hardware is supported.

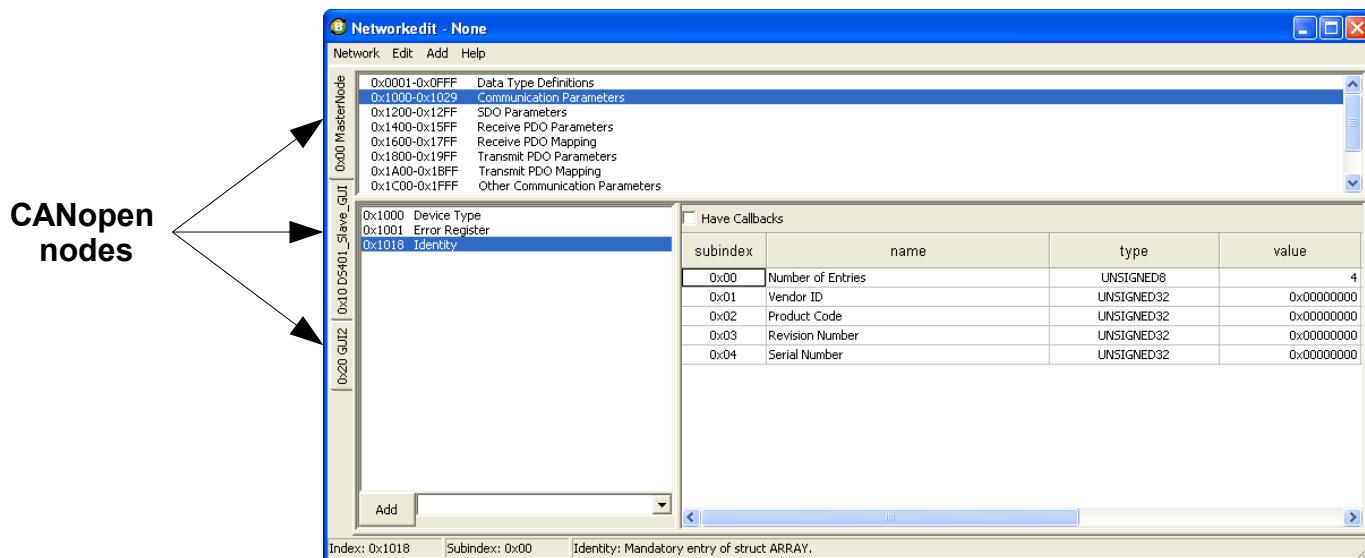
For testing purpose, use the can_tcp_win32 driver to simulate a CAN network over TCP/IP.



Once CanFestival instance is defined, add a master node :



In the Network editor, declare the nodes that will make your CANopen network. You have to provide EDS files, and give Node-ID.



Once EDS files are imported, you can now use nodes variables with your PLC.

Correspondence between IEC variables and Object dictionary entries of a CANopen node:

Creator:GPL Ghostscript 861 (epswr
CreationDate:2008/02/28 11:21:36
LanguageLevel:2

To add a variable, you have to:

- create one new variable in your PLC program
- give it the same type of the aimed CANopen variable
- drag and drop the CANopen variable from the Network Editor to the location colon of your program, as in the following example:

Network Editor

0x000 0x2000-0x5FFF Manufacturer Specific
 0x6000-0x9FFF Standardized Device Profile
 0xA000-0xBFFF Standardized Interface Profile

0x6000 Read Inputs 8 Bit
 0x6002 Polarity Input 8 Bit
0x6200 Write Outputs 8 Bit
 0x6202 Change Polarity Outputs 8 Bit
 0x6206 Error Mode Outputs 8 Bit
 0x6207 Error Value Outputs 8 Bit
 0x6401 Read Analogue Input 16 Bit
 0x6411 Write Analogue Output 16 Bit
 0x6423 Analogue Input Global Interrupt Enable
 0x6426 Analogue Input Interrupt Delta Unsigned
 0x6443 Analogue Output Error Mode
 0x6444 Analogue Output Error Value Integer

Have Callbacks

subindex	name	type
0x00	Number of Output 8 Bit	UNSIGNED8
0x01	Write Outputs 0x1 to 0x8	UNSIGNED8

drag variable

PLCopen Editor Variable Panel

Variable Panel

#	Name	Class	Type	Location	Initial Value	Retain	Constant
1	DigitalOut	Local	BYTE	%QB0.0.16.250		No	No
2	DigitalIn	Local	BYTE	%IB0.0.32.245		No	No
3	AnalogOut1	Local	MYTYPE	%QW0.0.16.25		No	No
4	AnalogOut2	Local	MYTYPE	%QW0.0.16.25		No	No
5	AnalogOut3	Local	INT	%QW0.0.16.25		No	No
6	AnalogIn1	Local	INT	%IW0.0.32.250		No	No
7	AnalogIn2	Local	INT	%IW0.0.32.250		No	No
8	Test	Local	Bitwise_Block			No	No
9	Test2	Local	Test_SFC			No	No
10	t1	Local	TOF			No	No
11	latch	Local	RS			No	No
12	t2	Local	TOF			No	No
13	VARexample	Local	USINT			No	No

Select a variable class:

- Input
- Output**
- Memory

when drop variable :
 a popup window appears



13	VARexample	Local	USINT	%QB0.0.16.25088.1			
----	------------	-------	-------	-------------------	--	--	--

c)Svgui plugin

PRE-ALPHA : Broken, documentation to be written.

8° The DEMO

This example provide a PLC working with the CANopen plugin. It's formed by a Master node and two DS401 virtual slave nodes of CanFestival to emulate I/O blocks.

a)Start the demo

Launch Beremiz with the Beremiz_Demo link in the windows start menu.

Thanks to this, when beremiz start, it automatically launches the “test project” example located in the beremiz directory.

b)Build the project



As this example is already configured, you can press on the build button to start building the project.

The “Log Console” displays, the differents building steps.

The build results by an executable, named as the project name. It's located in the build directory of the test project.

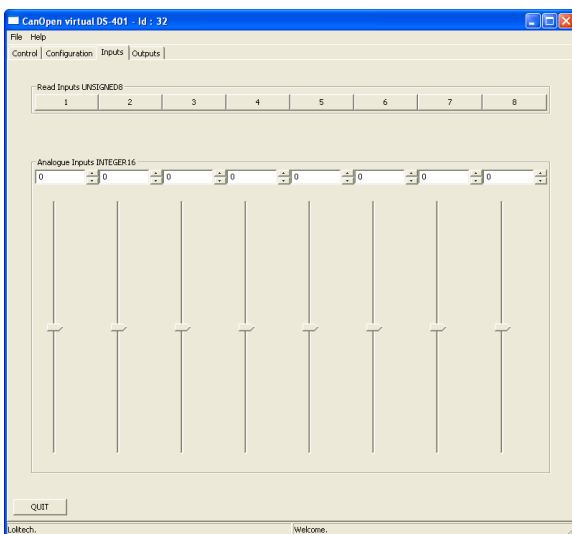
c)Run the PLC



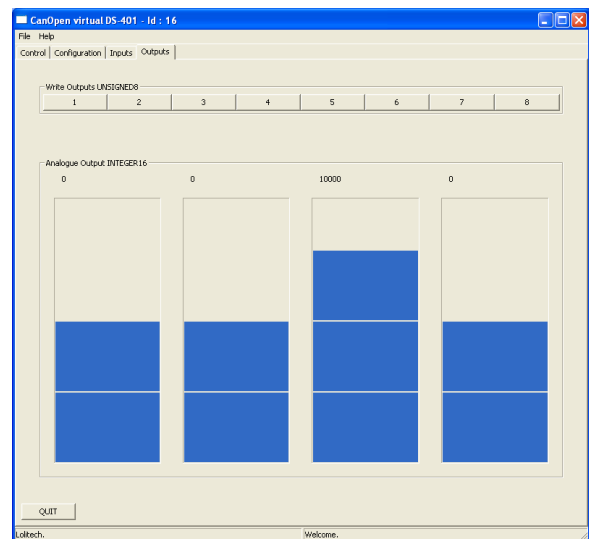
By pressing the “run” button, some external programs are launched before the PLC :

- a CAN tcp server to simulate a CANopen network
- a virtual slave node to simulate input block
- a virtual slave node to simulate output block

DS401 virtual slave nodes



Input block



Ouput block

PLC cycle period is 100 ms.

d) Stop the PLC



Just press the stop button, for stopping all processes.

e) Clean The PLC



By pressing the “clean” button, only the build directory of the project will be deleted.

D - Project Status

The version of Beremiz bundled with this manual is “pre-Alpha”. It means you should not entrust this software any critical mission, and consider it as a preview of some features that will be available in next releases.

E - FAQ

How to ask support to Beremiz project ?

Contact LOLitech at : contact@lolitech.fr

How to be informed about Beremiz status ?

Visit Beremiz website : www.beremiz.org

From there, you can subscribe to the mailing list.

How to use Beremiz on Windows ?

At the moment, Beremiz only runs on Windows Xp. Vista version is broken. It will be created soon.

How to use Beremiz on Linux ?

As Beremiz is developed on linux, it works nicely on this platform. At the moment, you must get from CVS and compile separately all Beremiz's sub-projects.

For CVS instructions refer to www.beremiz.org. In a near future, you will be able to get packages for your favorite distribution.

How to contribute to Beremiz project ?

Contact LOLitech at : contact@lolitech.fr