Lolitech - TBI sarl
24, rue Pierre Evrat
88100 Saint-Dié-des-Vosges
Tel : +33 3 29 52 95 67

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".

LOLI Tech
Logiciel Libre et Technologie

Table des matières

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.

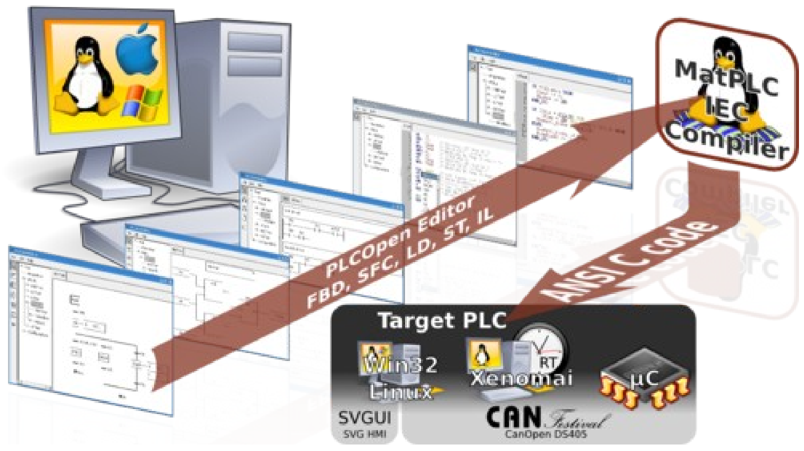
With Beremiz, you can :

- 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:

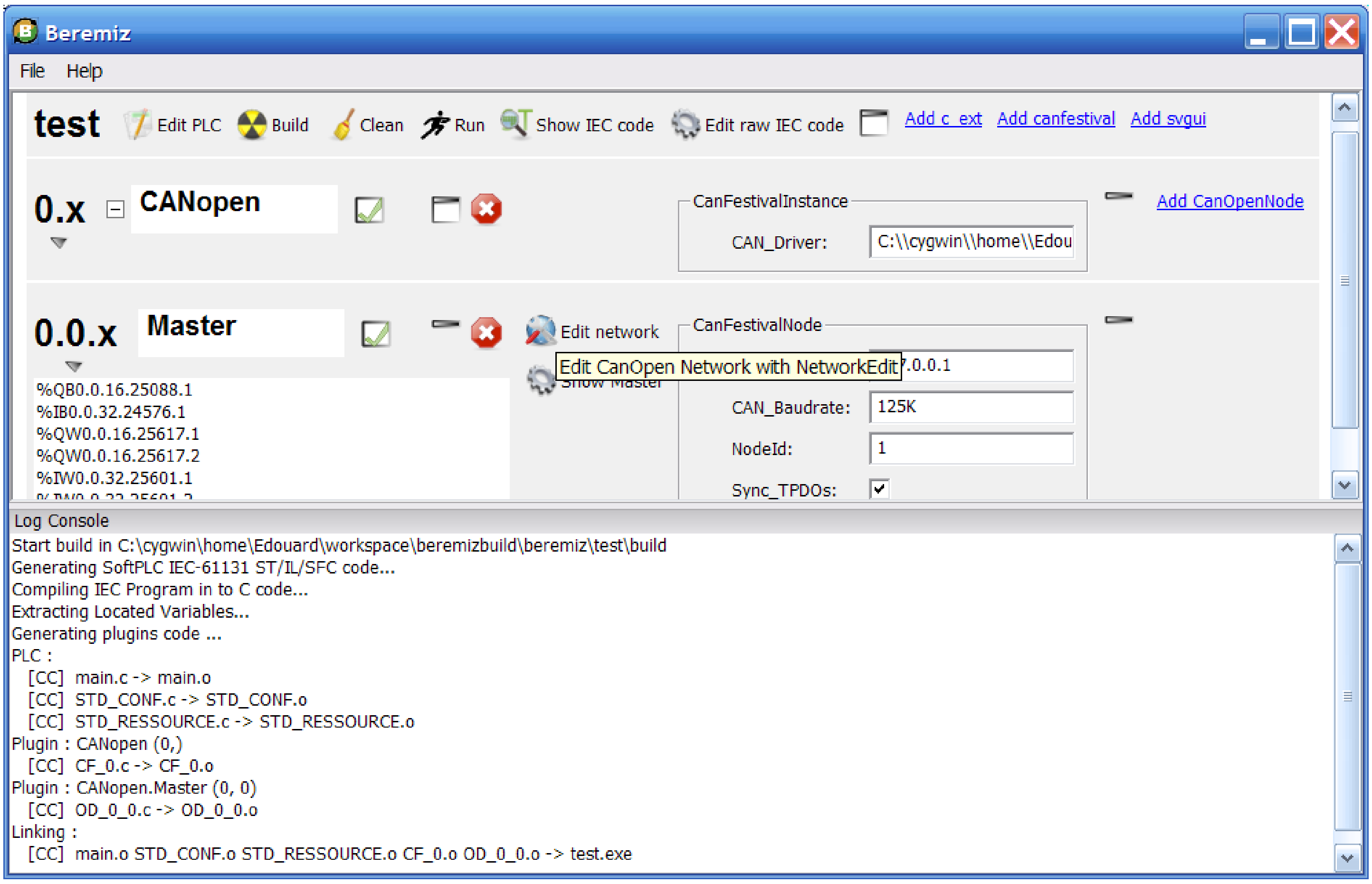
1. PLCOpen Editor : Multi-platform automation IDE
2. MatPLC's IEC compiler : IEC 61131-3 compiler
3. CanFestival : CANOpen interface to physical I/O
4. SVGUI : automated HMI tool, based on SVG

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 let PLC programmer 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



CanFestivalInstance
CAN_Driver: C:\\cygwin\\home\\Edou

CanFestivalNode
.0.0.1
CAN_Baudrate: 125K
NodeId: 1
Sync_TPDOs: ✓

```
%QB0.0.16.25088.1
%IB0.0.32.24576.1
%QW0.0.16.25617.1
%QW0.0.16.25617.2
%IW0.0.32.25601.1
```

```
Start build in C:\cygwin\home\Edouard\workspace\beremizbuild\beremiz\test\build
Generating SoftPLC IEC-61131 ST/IL/SFC code...
Compiling IEC Program in to C code...
Extracting Located Variables...
Generating plugins code ...
PLC :
   [CC]  main.c -> main.o
   [CC]  STD_CONF.c -> STD_CONF.o
   [CC]  STD_RESSOURCE.c -> STD_RESSOURCE.o
Plugin : CANopen (0,)
   [CC]  CF_0.c -> CF_0.o
Plugin : CANopen.Master (0, 0)
   [CC]  OD_0_0.c -> OD_0_0.o
Linking :
   [CC]  main.o STD_CONF.o STD_RESSOURCE.o CF_0.o OD_0_0.o -> test.exe
```

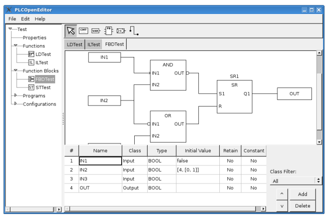
2° PLCopen Editor

The PLCopen Editor saves and loads XML projects accordingly to PLCopen TC6-XML Schemes.

Lolitech - TBI sarl
24, rue Pierre Evrat
88100 Saint-Dié-des-Vosges
Tel : +33 3 29 52 95 67

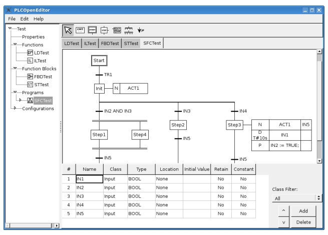
LOLI Tech
Logiciel Libre et Technologie

Function Block Diagram - FBD



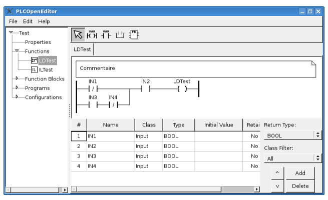
#	Name	Class	Type	Initial Value	Retain	Constant
1	IN1	Input	BOOL	false	No	No
2	IN2	Input	BOOL	[4, [0, 1]]	No	No
3	IN3	Input	BOOL		No	No
4	OUT	Output	BOOL		No	No

Sequential Function Chart - SFC



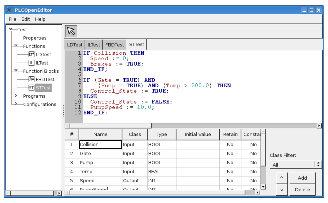
#	Name	Class	Type	Location	Initial Value	Retain	Constant
1	IN1	Input	BOOL	None		No	No
2	IN2	Input	BOOL	None		No	No
3	IN3	Input	BOOL	None		No	No
4	IN4	Input	BOOL	None		No	No
5	IN5	Input	BOOL	None		No	No

Ladder Diagram - LD



#	Name	Class	Type	Initial Value	Retai
1	IN1	Input	BOOL		No
2	IN2	Input	BOOL		No
3	IN3	Input	BOOL		No
4	IN4	Input	BOOL		No

Structured Text - ST

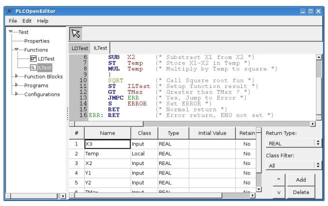


```
IF Collision THEN
  Speed := 0;
  Brakes := TRUE;
END_IF;

IF (Gate = TRUE) AND
   (Pump = TRUE) AND (Temp > 200.0) THEN
  Control_State := TRUE;
ELSE
  Control_State := FALSE;
  PumpSpeed := 10.0;
END_IF;
```

#	Name	Class	Type	Initial Value	Retain	Consta
1	Collision	Input	BOOL		No	No
2	Gate	Input	BOOL		No	No
3	Pump	Input	BOOL		No	No
4	Temp	Input	REAL		No	No
5	Speed	Output	INT		No	No

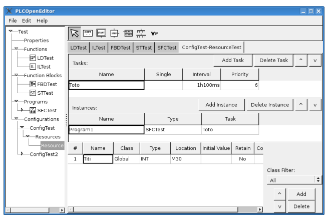
Instruction List - IL



```
        SUB   X2      (* Substract X1 from X2 *)
        ST    Temp    (* Store X1-X2 in Temp *)
        MUL   Temp    (* Multiply by Temp to square *)
        )
        SQRT          (* Call Square root fun *)
        ST    ILTest  (* Setup function result *)
        GT    TMax    (* Greater than TMax ? *)
        JMPC  ERR     (* Yes, Jump to Error *)
        S     ERROR   (* Set ERROR *)
        RET           (* Normal return *)
ERR:    RET           (* Error return, ENO not set *)
```

#	Name	Class	Type	Initial Value	Retain
1	X3	Input	REAL		No
2	Temp	Local	REAL		No
3	X2	Input	REAL		No
4	Y1	Input	REAL		No
5	Y2	Input	REAL		No

Configurations, Resources,Tasks



Tasks:

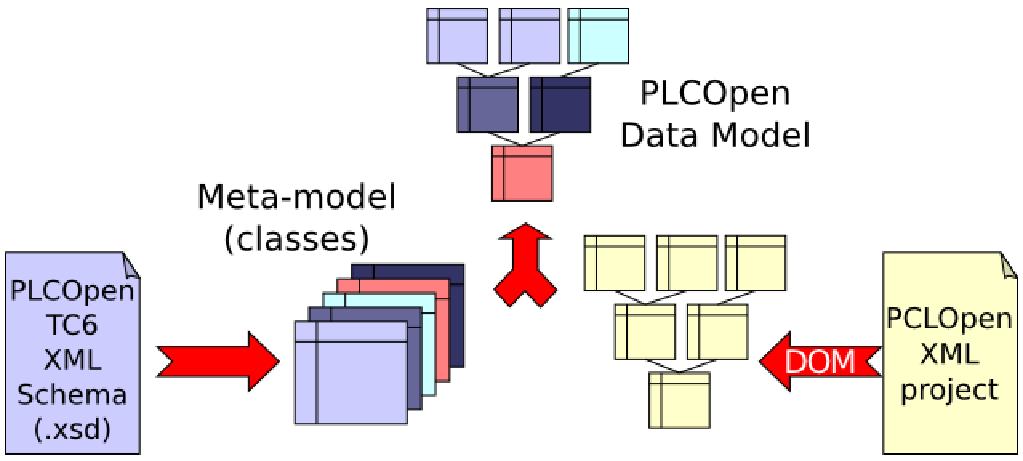
Name	Single	Interval	Priority
Toto		1h100ms	6

Instances:

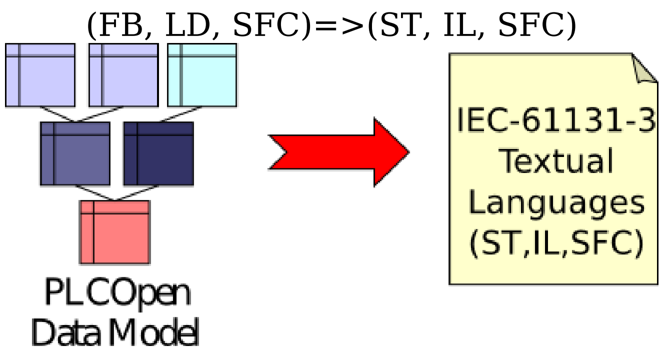
Name	Type	Task
Program1	SFCTest	Toto

#	Name	Class	Type	Location	Initial Value	Retain	Co
1	Titi	Global	INT	M30		No	

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

4° CANopen Stack

CanFestival is an OpenSource CANOpen framework, licensed with GPLv2 and LGPLv2.

With CanFestival you CAN :

- Turn any µC 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

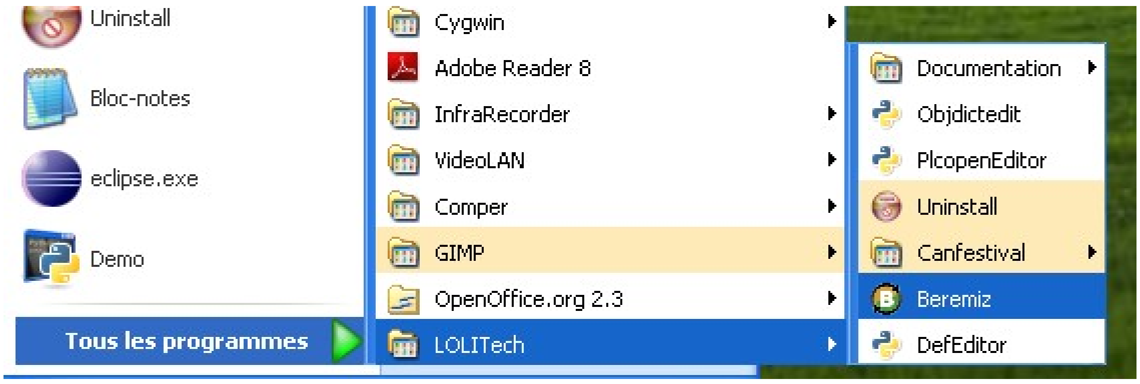
5° SVG HMI toolkit

PRE-ALPHA : Broken, documentation to be written.

C - Start with Beremiz

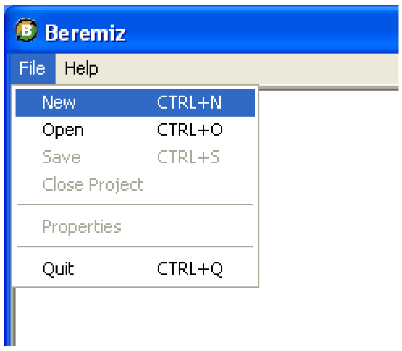
1° Launch Beremiz

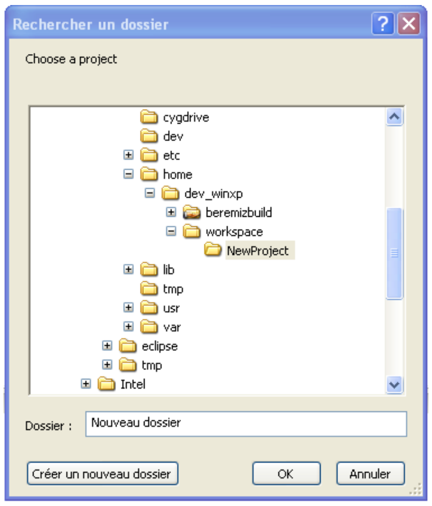
Beremiz installer provides many shortcuts in the Windows 's “Start Menu”. So browse your start menu as the following and launch Beremiz :



2° Create a new project

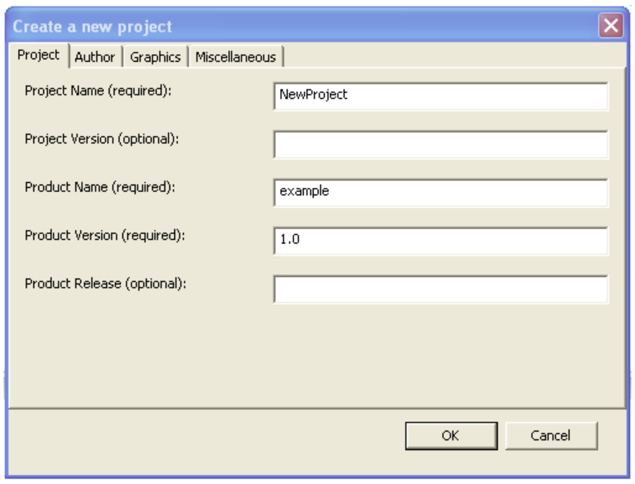
Beremiz project need an empty directory to be created. This directory will contain all the necessary files to develop and build your PLC.





Dossier : Nouveau dossier

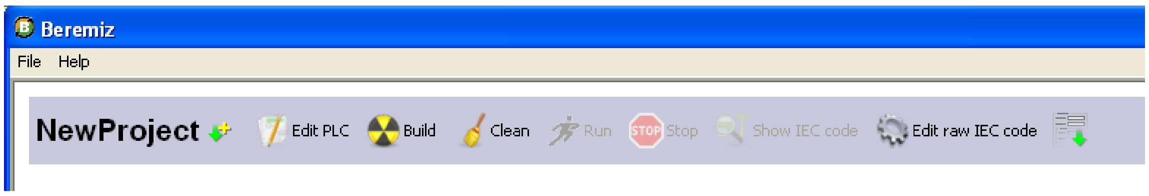
Choose where to create project, and create your project directory with project name



Project Name (required): NewProject
Project Version (optional):
Product Name (required): example
Product Version (required): 1.0
Product Release (optional):

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

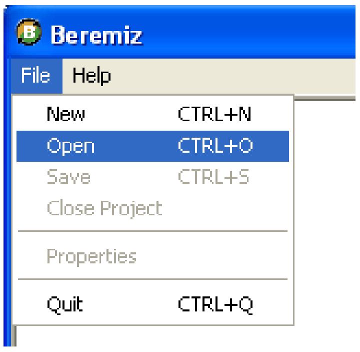
Once the project is created, the project's toolbar appeared

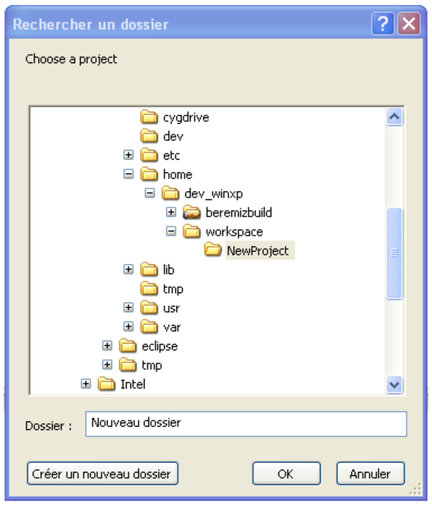


Your project directory now contain two files :

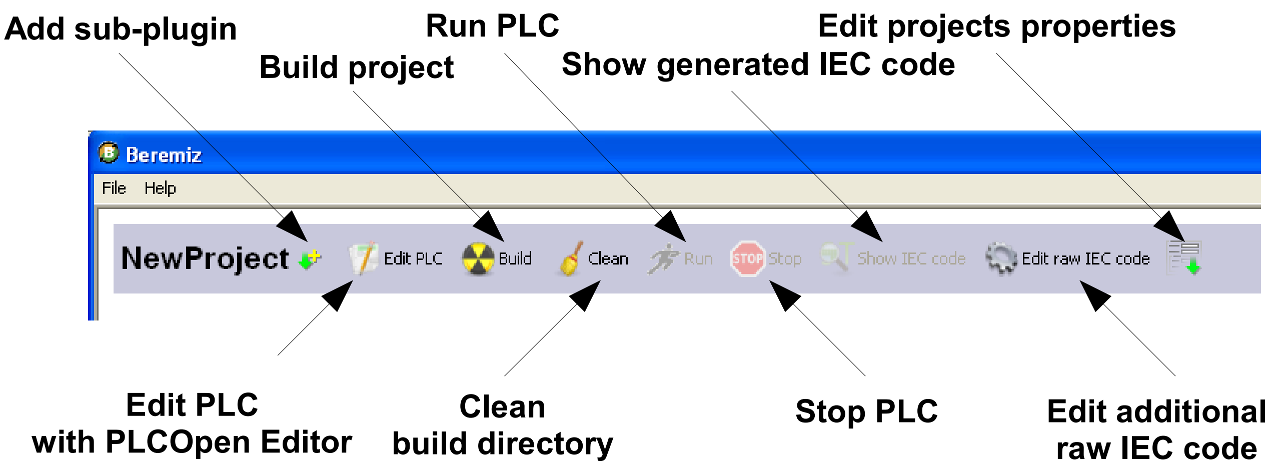
- plc.xml : contain project properties
- beremiz.xml : contain Beremiz default properties

3° Open an existing project





4° Project's tool bar

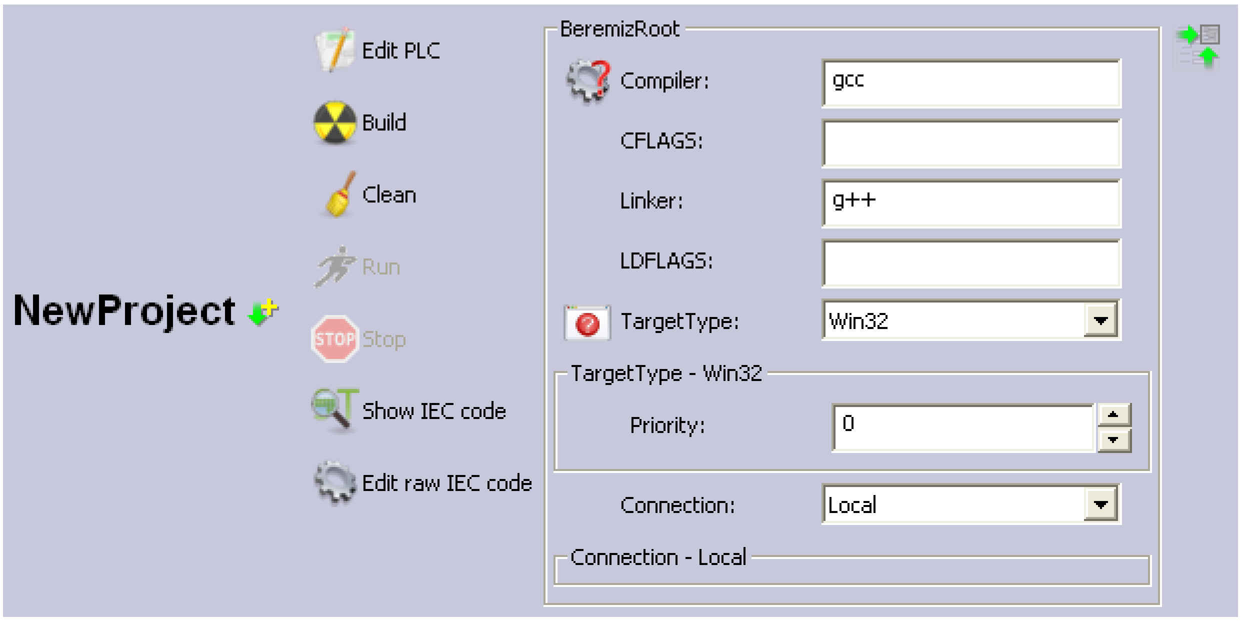


5° Configure your project

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 .



BeremizRoot

Compiler: gcc
CFLAGS:
Linker: g++
LDFLAGS:
TargetType: Win32

TargetType - Win32

Priority: 0

Connection: Local

Connection - Local

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

b)Linux projects

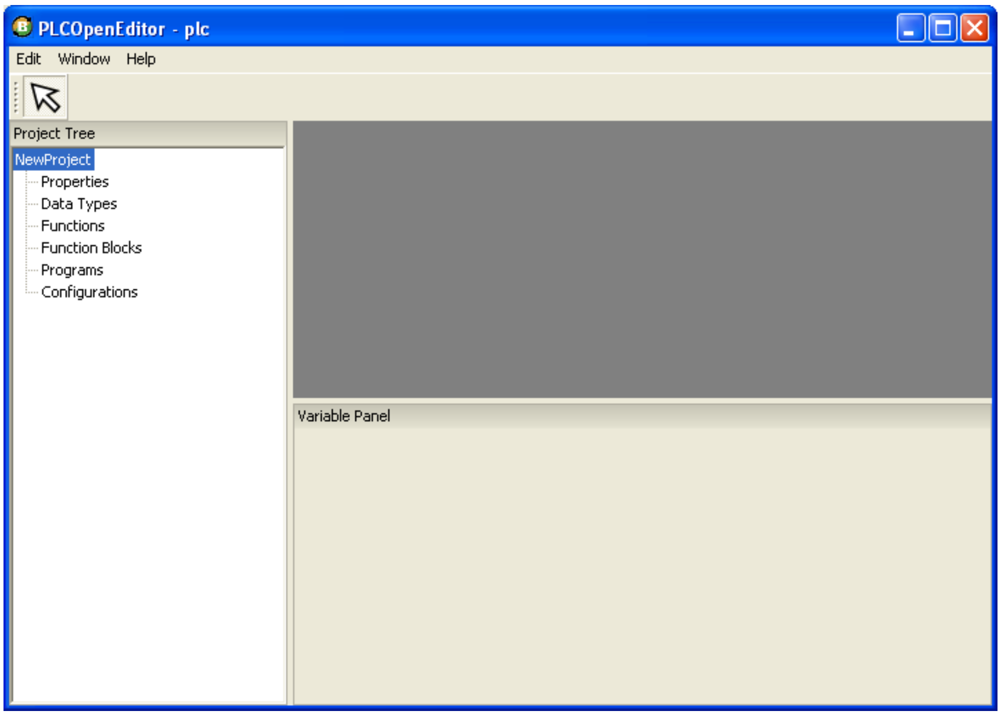
PRE-ALPHA : To be written

Lolitech - TBI sarl
24, rue Pierre Evrat
88100 Saint-Dié-des-Vosges
Tel : +33 3 29 52 95 67

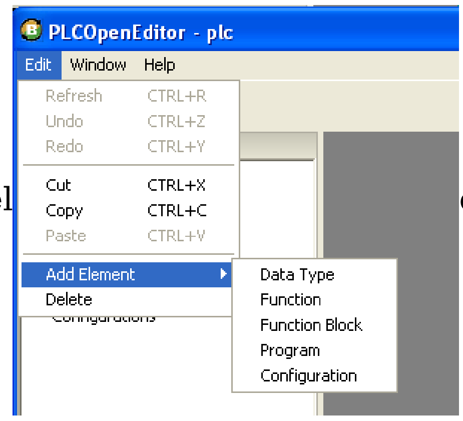
LOLI Tech
Logiciel Libre et Technologie

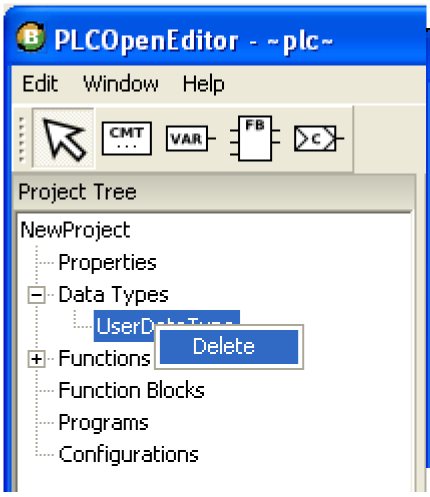
6° PLCopen Editor

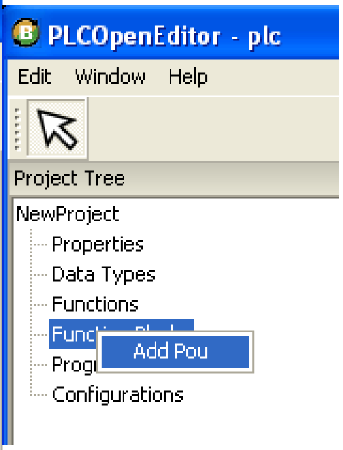
PlcOpen Editor Interface:



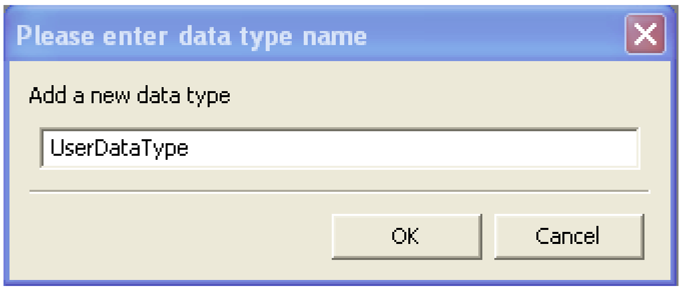
Add / Del ete POU:





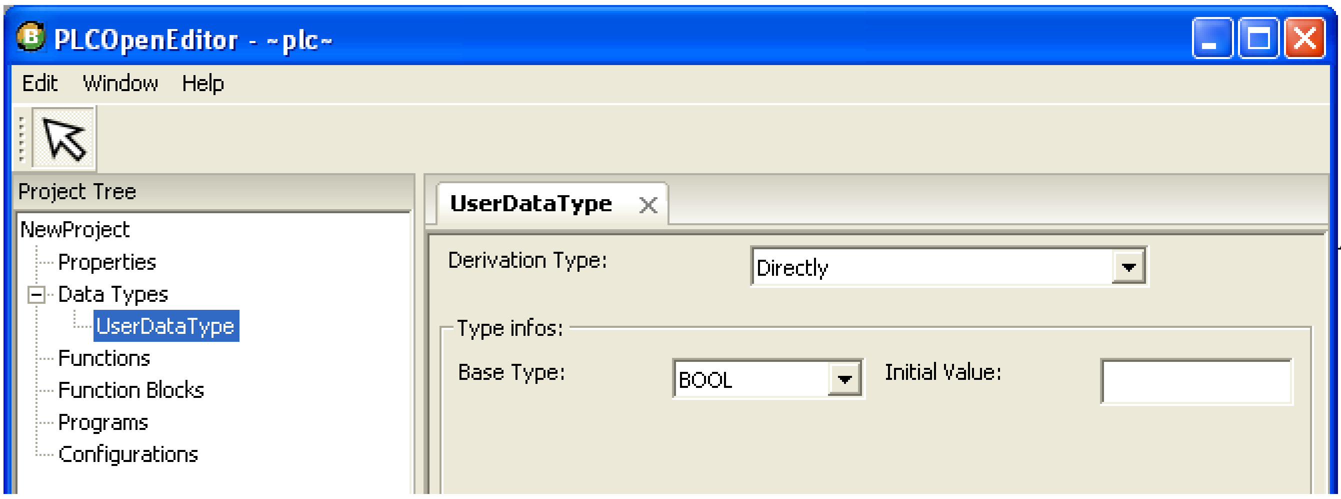


Add Data Types :


Please enter data type name

Add a new data type

UserDataType



Derivation Type: Directly

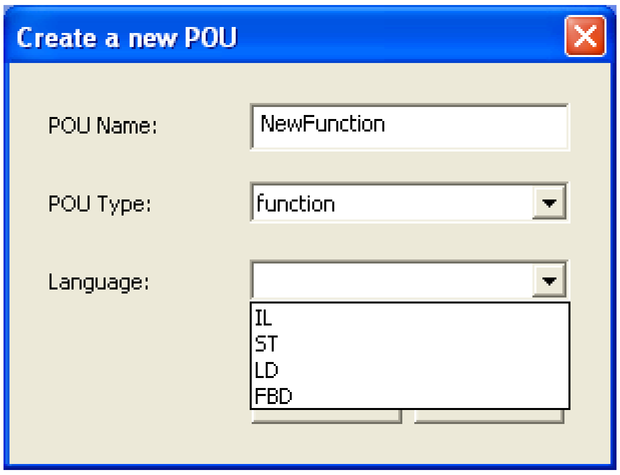
Type infos:

Base Type: BOOL

Initial Value:

Add

Functions :

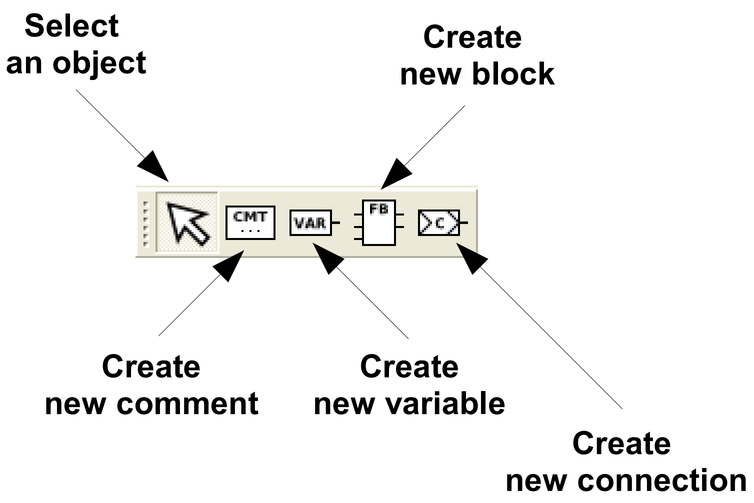

Create a new POU

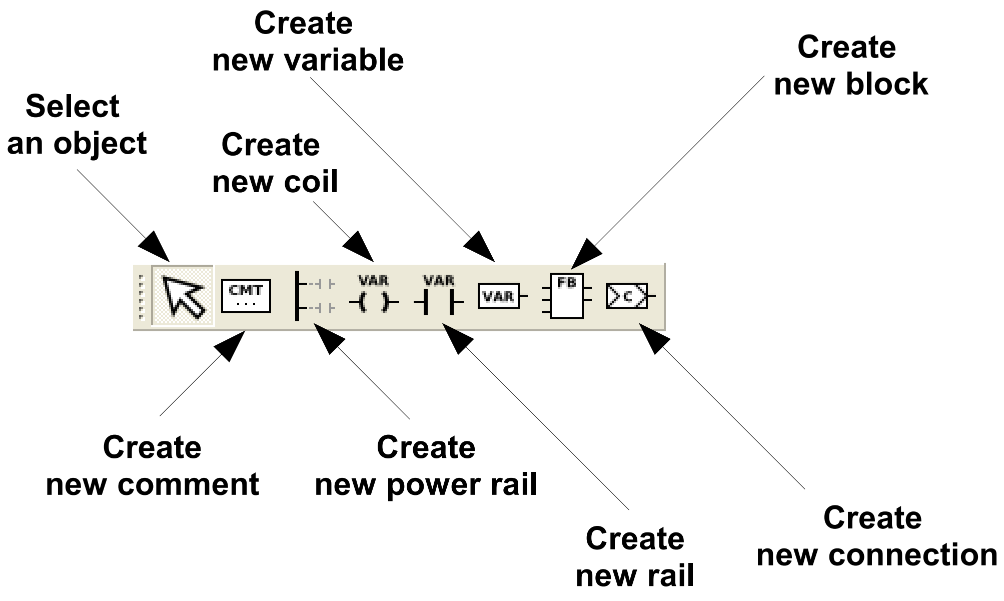
POU Name: NewFunction

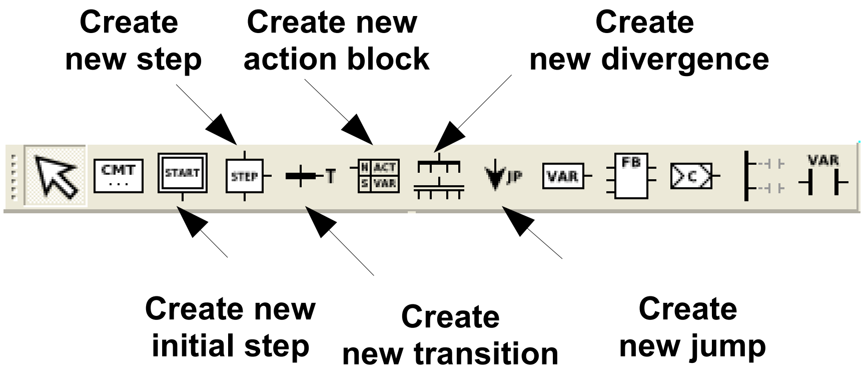
POU Type: function

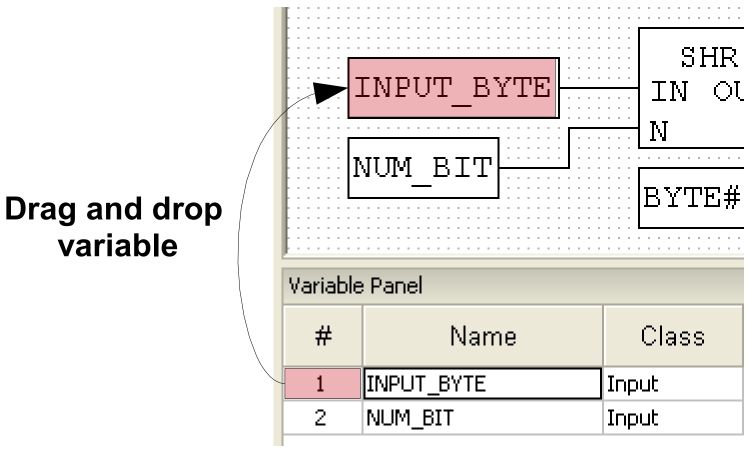
Language:

Function Diagram tool bar :





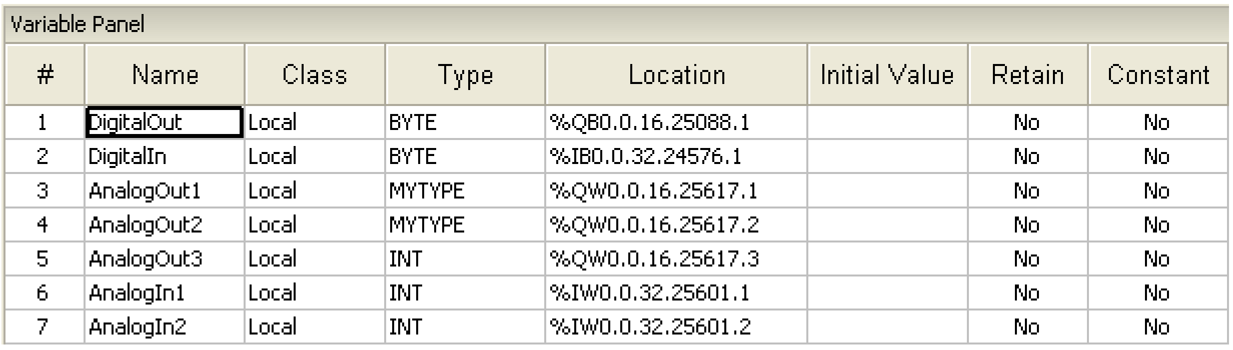




Variable Panel

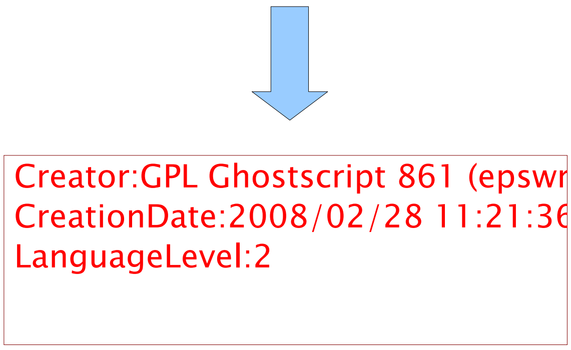
#	Name	Class
1	INPUT_BYTE	Input
2	NUM_BIT	Input

Program Variable Panel :



Variable Panel

#	Name	Class	Type	Location	Initial Value	Retain	Constant
1	DigitalOut	Local	BYTE	%QB0.0.16.25088.1		No	No
2	DigitalIn	Local	BYTE	%IB0.0.32.24576.1		No	No
3	AnalogOut1	Local	MYTYPE	%QW0.0.16.25617.1		No	No
4	AnalogOut2	Local	MYTYPE	%QW0.0.16.25617.2		No	No
5	AnalogOut3	Local	INT	%QW0.0.16.25617.3		No	No
6	AnalogIn1	Local	INT	%IW0.0.32.25601.1		No	No
7	AnalogIn2	Local	INT	%IW0.0.32.25601.2		No	No


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

7° Plugins

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

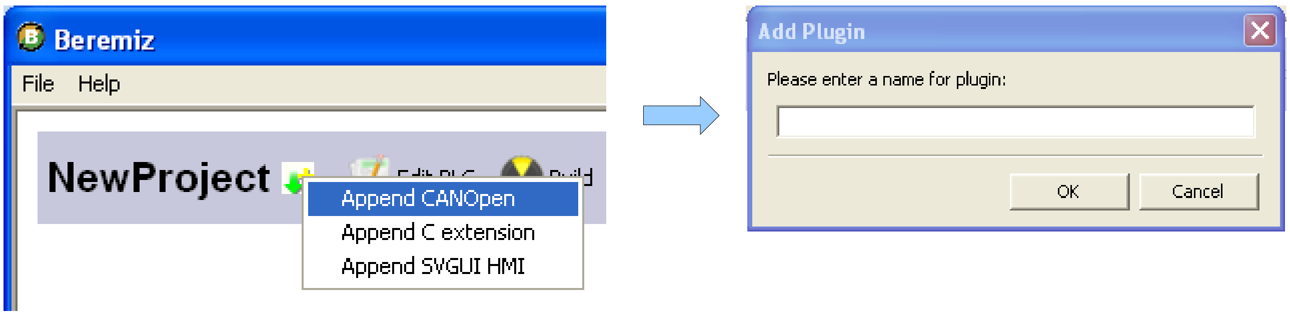
Each plugin is associated with a range of IEC-61131-3 “directly represented variables”. Nested plugins are mapped to sub-ranges. As an example:

```
                        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
```

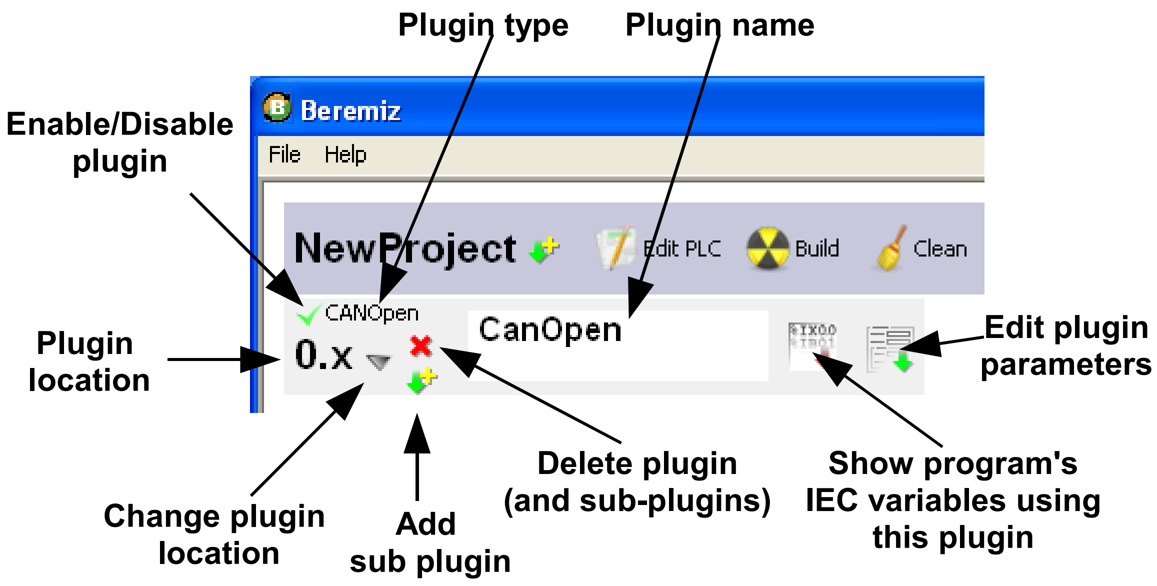
During build, “directly represented variables” declared in PLC program are dispatched in plugin tree according to their location, an consumed by plugins to produce corresponding C code.

a)Adding a plugin to your project

Add a plugin to your project :



Plugin appears now in the project's tree :

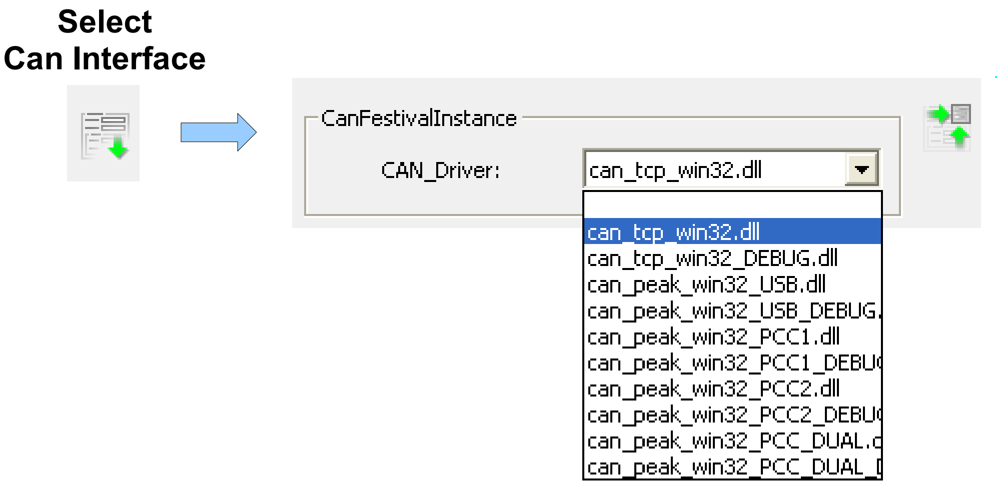


b)CANopen plugin

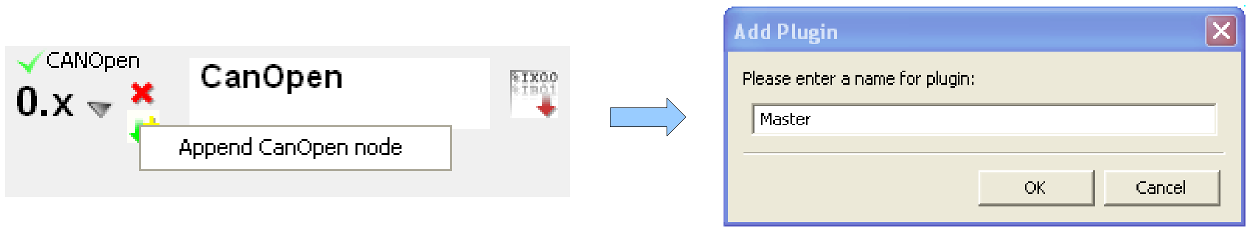
Thanks to this plug-in, your PLC can act as a CanOpen Master and control slaves Slave Nodes on different CAN buses.

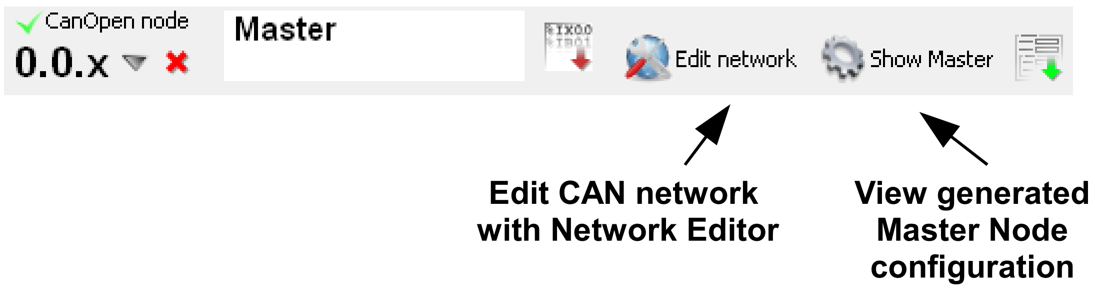
You have to choose the CAN driver corresponding to your Hardware. PRE-ALPHA: At that time only Peak-System CAN hardware is supported.

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

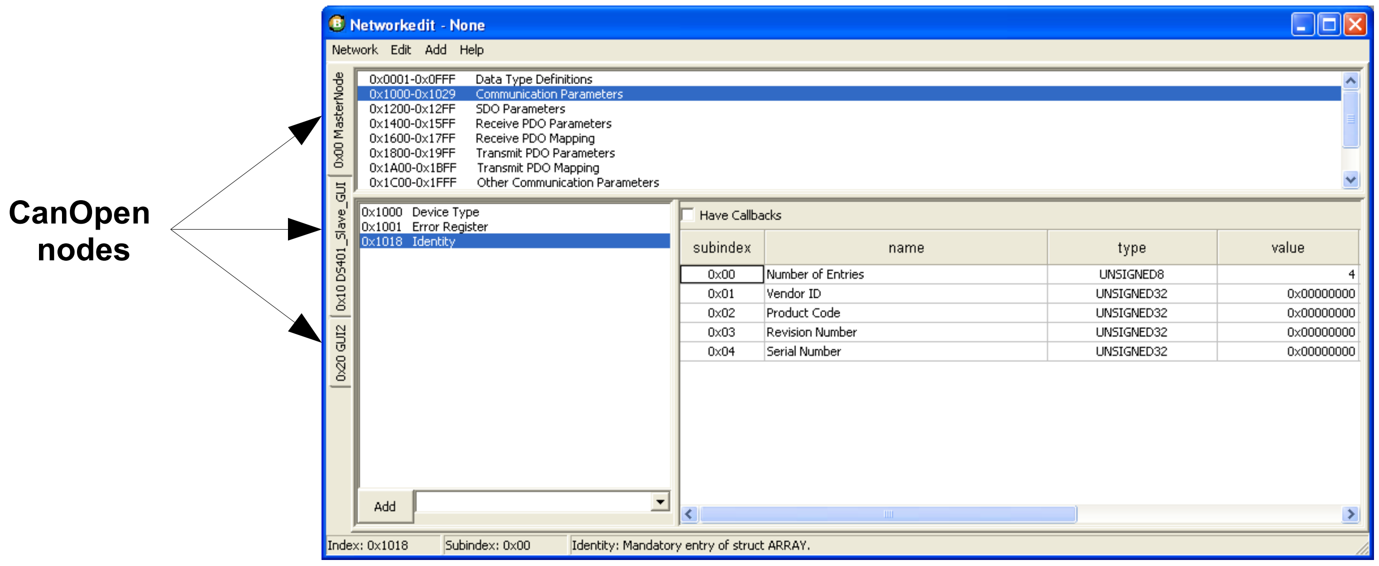


Once CanFestival instance is defined, add a master node :





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



0x0001-0x0FFF	Data Type Definitions
0x1000-0x1029	Communication Parameters
0x1200-0x12FF	SDO Parameters
0x1400-0x15FF	Receive PDO Parameters
0x1600-0x17FF	Receive PDO Mapping
0x1800-0x19FF	Transmit PDO Parameters
0x1A00-0x1BFF	Transmit PDO Mapping
0x1C00-0x1FFF	Other Communication Parameters

0x1000	Device Type
0x1001	Error Register
0x1018	Identity

subindex	name	type	value
0x00	Number of Entries	UNSIGNED8	4
0x01	Vendor ID	UNSIGNED32	0x00000000
0x02	Product Code	UNSIGNED32	0x00000000
0x03	Revision Number	UNSIGNED32	0x00000000
0x04	Serial Number	UNSIGNED32	0x00000000

Index: 0x1018 Subindex: 0x00 Identity: Mandatory entry of struct ARRAY.

XXXX Explain variable drag'n'drop here !

8° The DEMO

Beremiz use the DS401 virtual slave nodes of CanFestival to emulate I/O blocks, and

a)

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.

1° PLC builder

2° PLCopen Editor

3° IEC-61131-3 compiler

4° CANopen Stack

5° SVG HMI toolkit