

Automatisme libre avec Beremiz

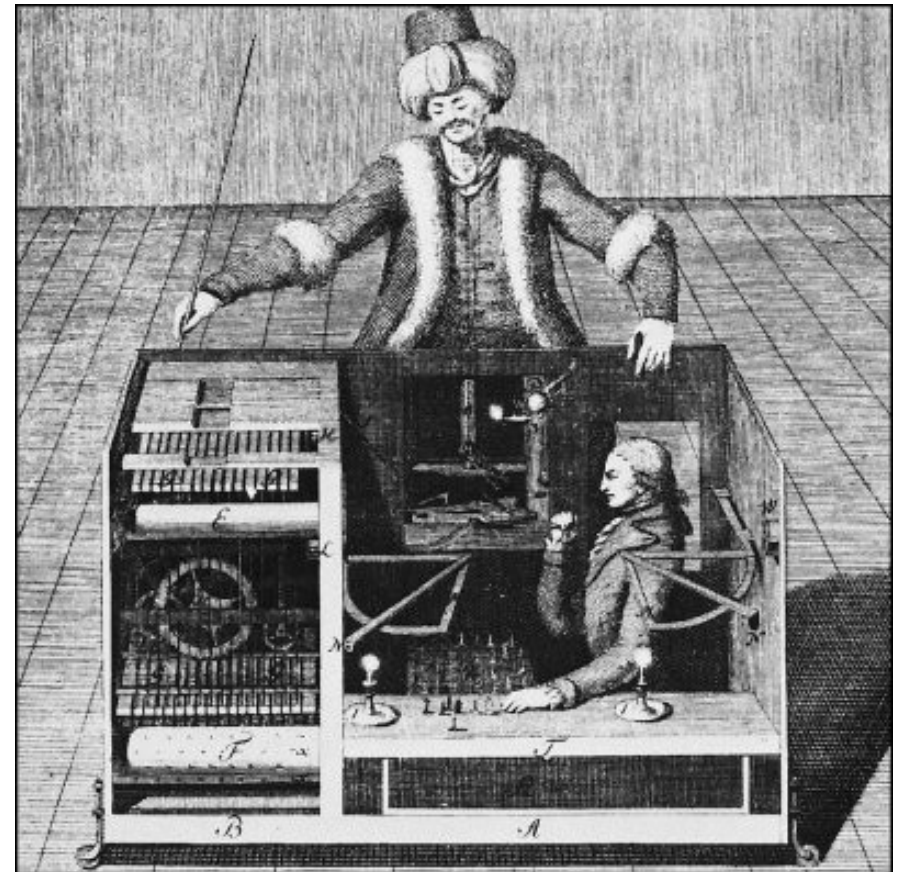
Free as a Beer ?

- L'essentiel de ce que nous consommons est conditionné par des systèmes automatisés.
- Existe-t-il des logiciels libres pour construire des automatismes ?



L'automatique est l'ancêtre de l'informatique

- L'informatique (information automatique) désigne l'automatisation du traitement de l'information.
- Aujourd'hui, les automates sont eux mêmes des ordinateurs.

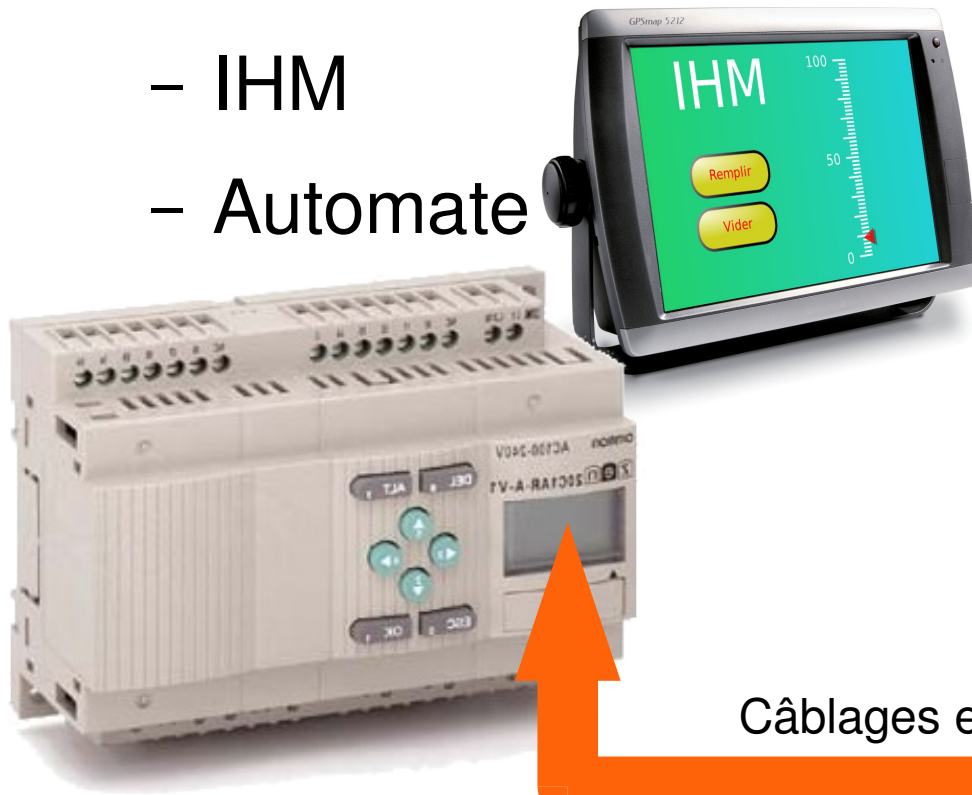


Qu'est ce qu'un automatisme ?

- Partie commande
- Partie opérative

- IHM

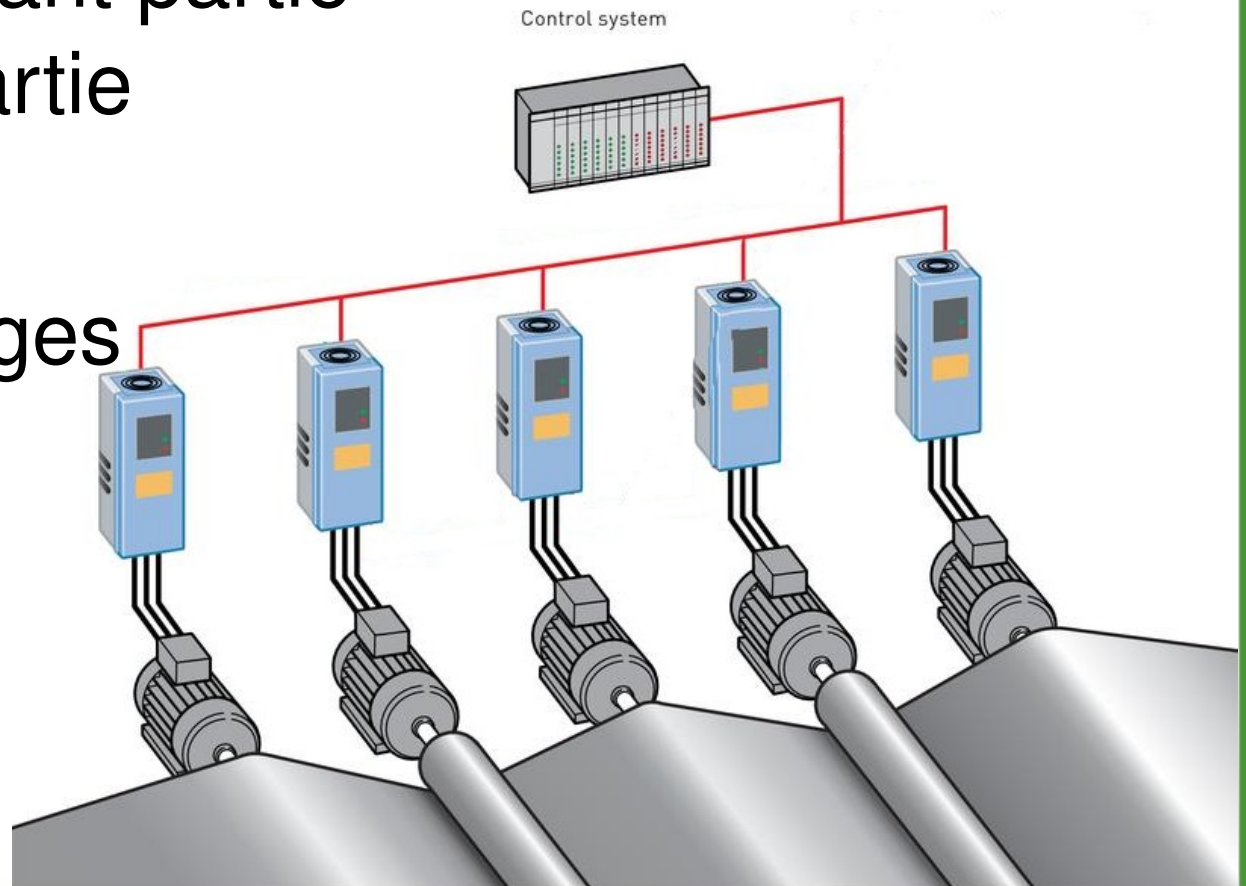
- Automate



Câblages et/ou Bus de terrain

Bus de terrain

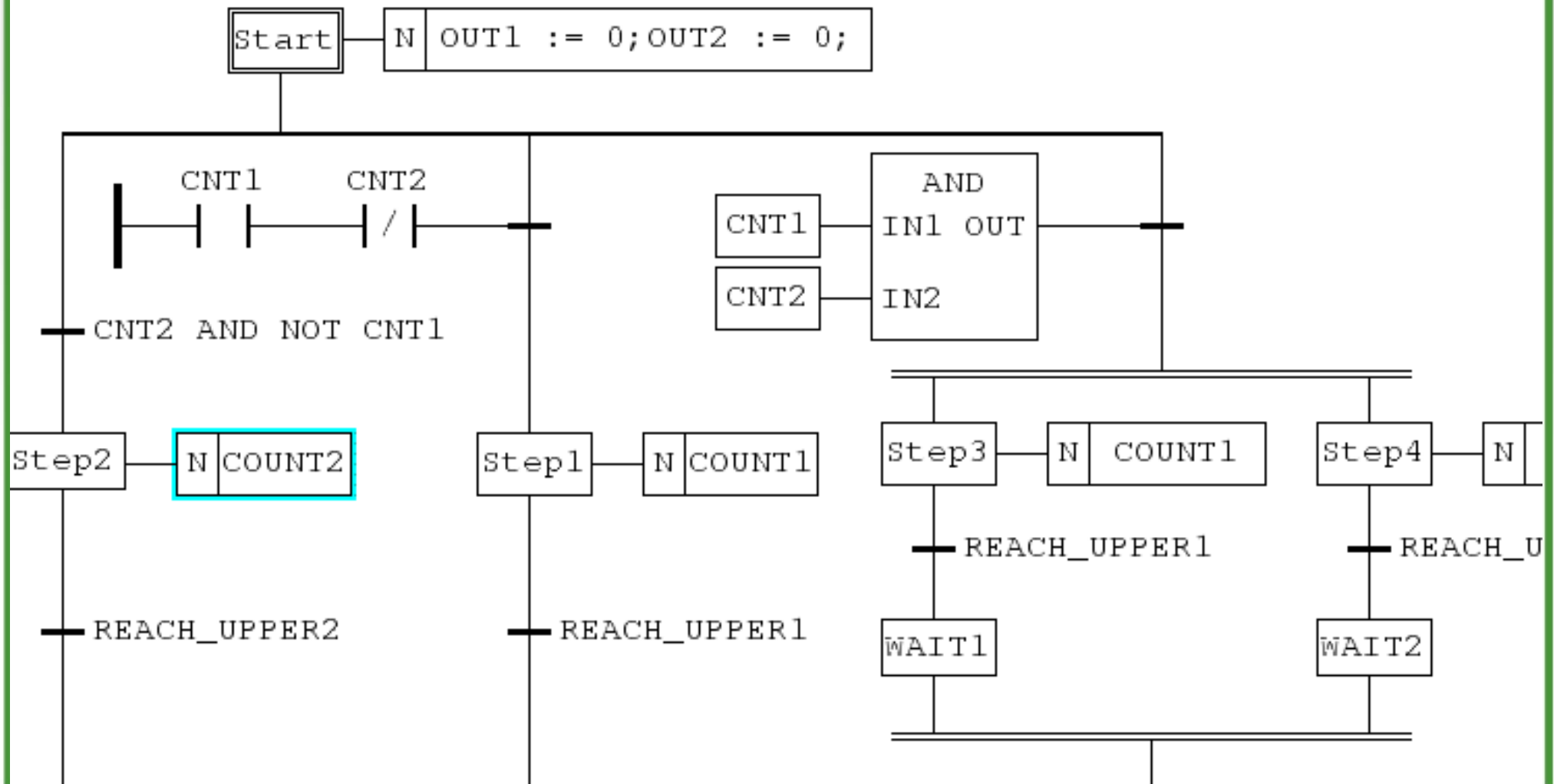
- Réseau connectant partie commande et partie opérative
- Réduit les câblages
- Normalise les échanges



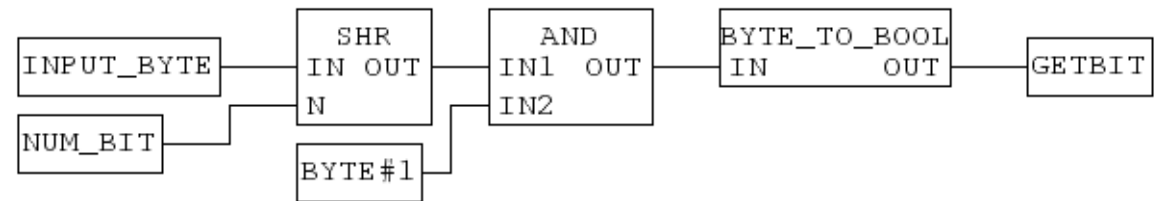
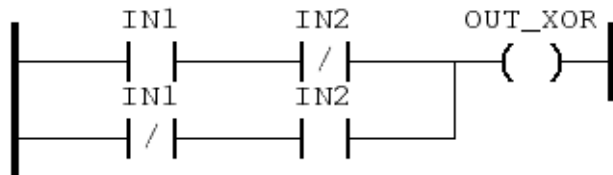
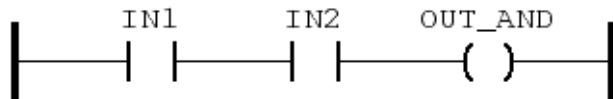
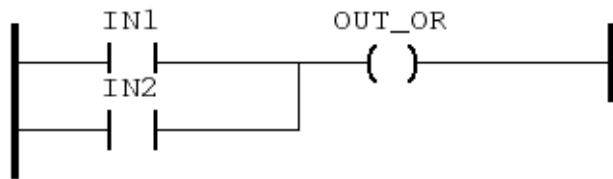
Principaux standards de l'automatique

- Bus de terrain
 - RS-485
 - Profibus
 - ModBus
 - CAN
 - DeviceNet
 - CANopen
- Ethernet
 - ModBus TCP
 - Powerlink
 - EtherCat
- Programmation
 - IEC-61131
 - PLCopen XML
 - IEC-61499

IEC-61131-3 (SFC)



IEC-61131-3 (LD, FBD)

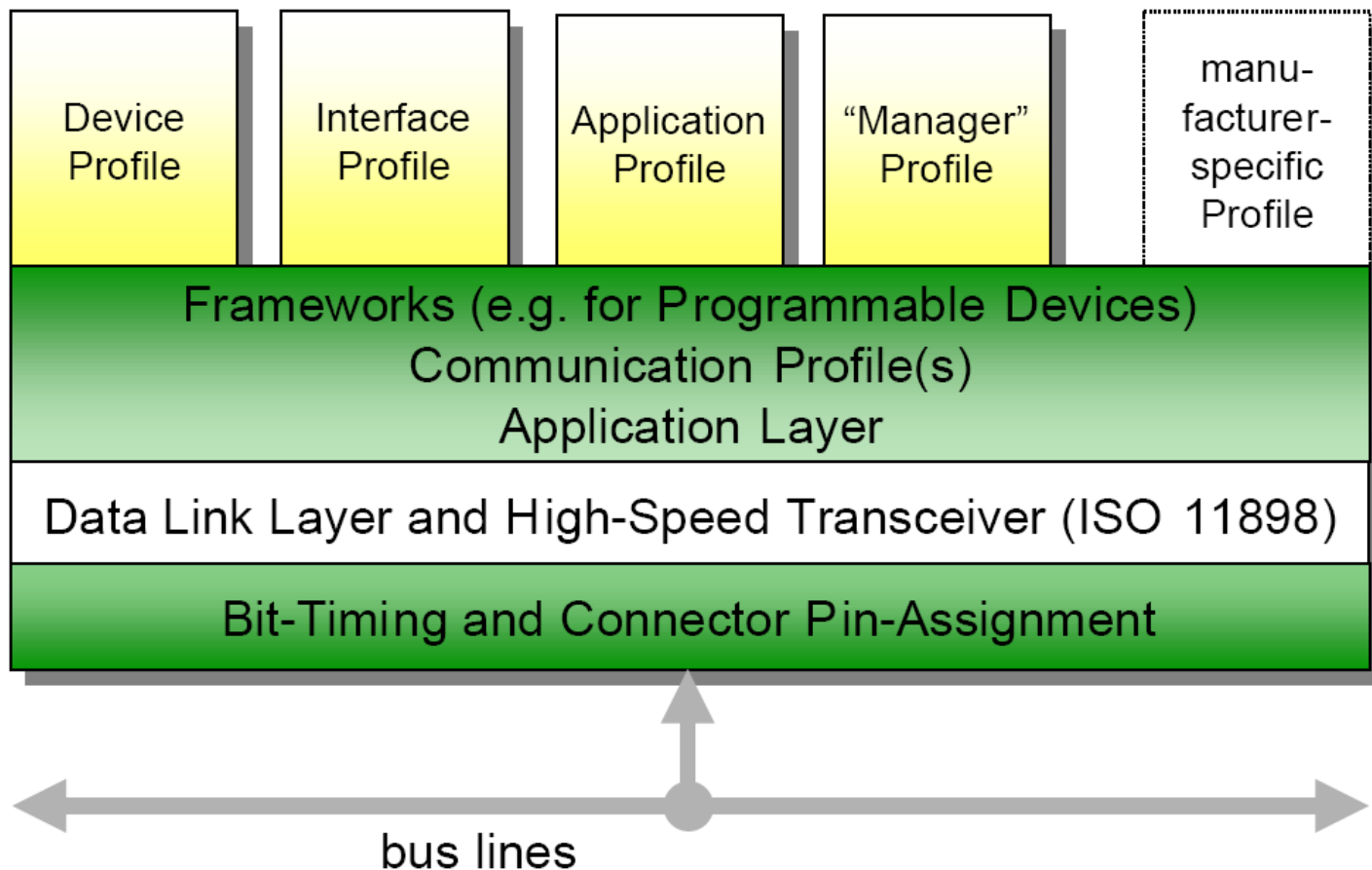


IEC-61131-3 (ST, IL)

```
1 IF Collision THEN
2   Speed := 0;
3   Brakes := ON;
4 END_IF;
5
6 IF (Gate = CLOSED) AND
7   (Pump = ON) AND (Temp > 200.0) THEN
8   Control_State := Active;
9 ELSE
10  Control_State := Hold;
11  PumpSpeed := 10.0;
12 END_IF;
```

```
2 LD Y1
3 SUB Y2 (* Subtract Y2 from Y1 *)
4 ST Temp (* Store Y1-Y2 in Temp *)
5 MUL Temp (* Multiply by Temp to square *)
6 ADD ( X1 (* Defer ADD *)
7 SUB X2 (* Subtract X1 from X2 *)
8 ST Temp (* Store X1-X2 in Temp *)
9 MUL Temp (* Multiply by Temp to square *)
10 )
11 SQR T (* Call Square root fun *)
12 ST ILTest (* Setup function result *)
13 GT TMax (* Greater than TMax ? *)
14 JMPC ERR (* Yes, Jump to Error *)
15 S ERROR (* Set ERROR *)
16 RET (* Normal return *)
17 ERR: RET (* Error return, ENO not set *)
```

CANopen



Libérer l'automatisme

- Indépendance des automaticiens
- Pérennité des automatismes
- Favoriser l'application de standards ouverts
- Favoriser le partage des connaissances



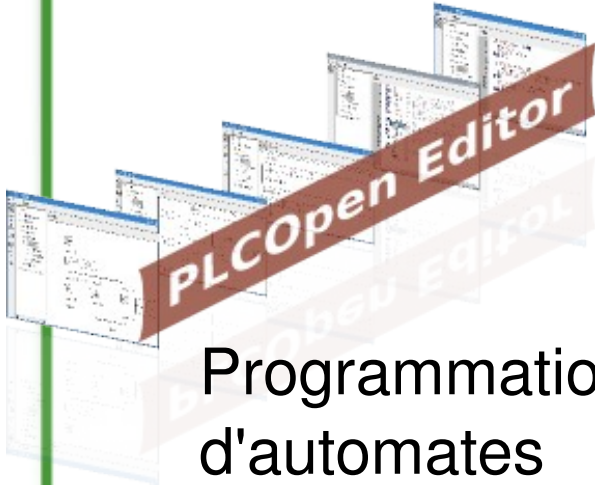


Compilateur
IEC-61131-3

CAN *Festival*
Pile CANopen

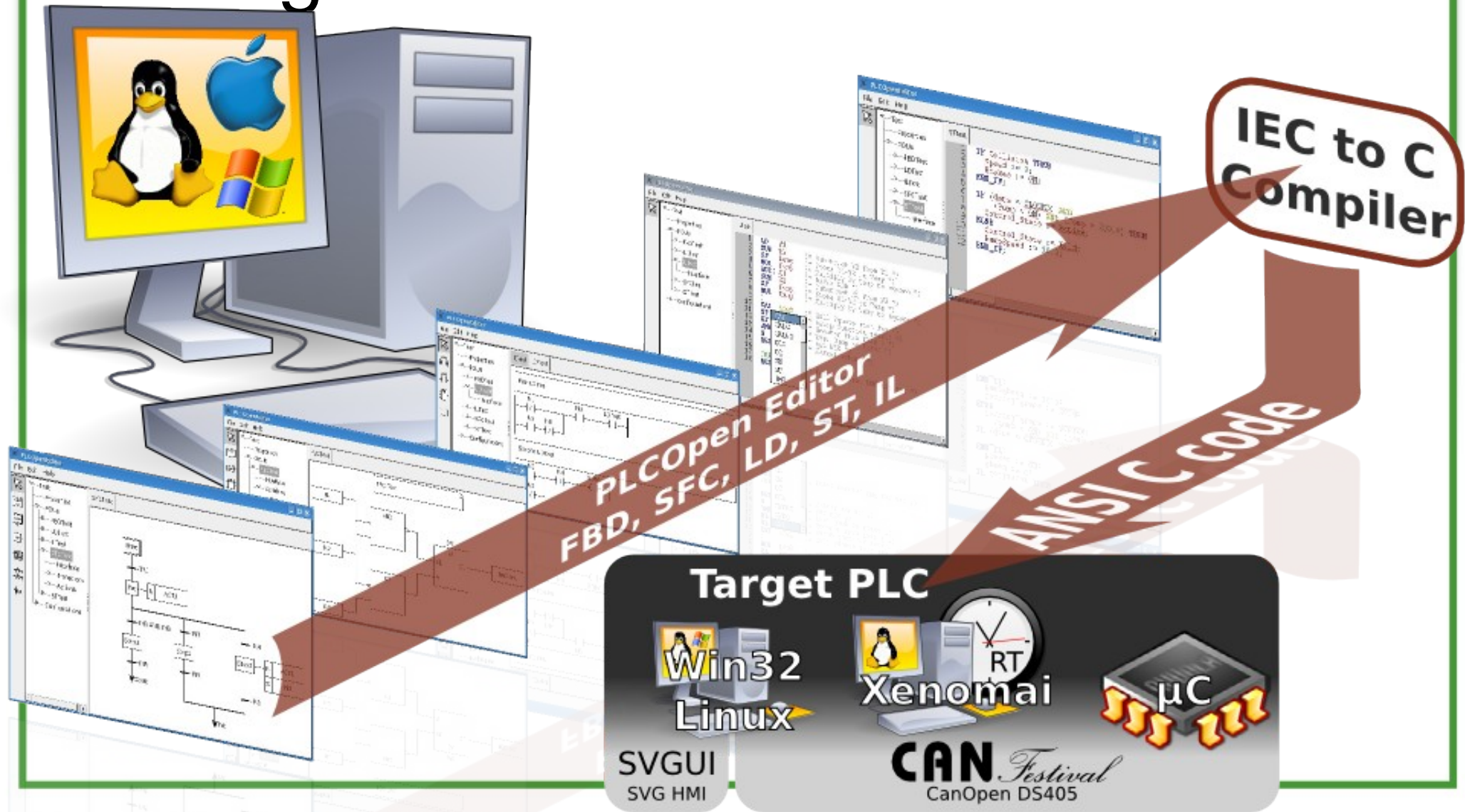


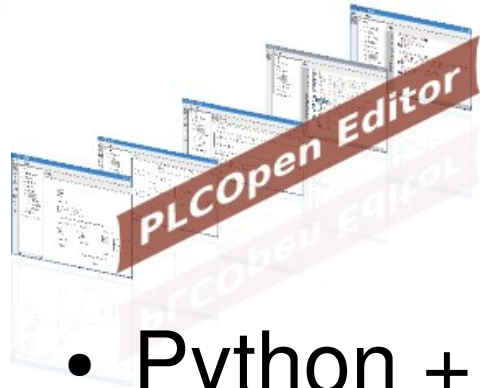
Construction
d'IHM

The logo for PLCOpen Editor, featuring a stack of windows and a red banner with the text 'PLCOpen Editor'.

Programmation
d'automates

Programmation d'automatismes





Editeur PLCopen

- Python + wxPython.
- Modèle Vue Contrôleur.
- Méta-modèle basé sur la XSD officielle.
- Travail sur fichiers XML PLCopen.
- Export textuel des diagrammes FBD,LD en ST.
- Génération et impression de documentation.

PLCOpenEditor - plc

Edit Window Help

Project Tree

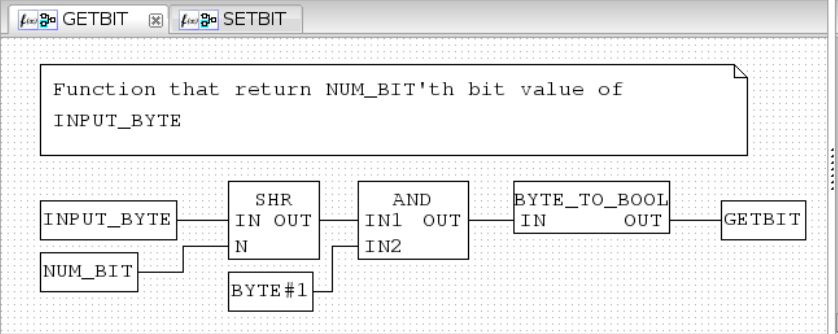
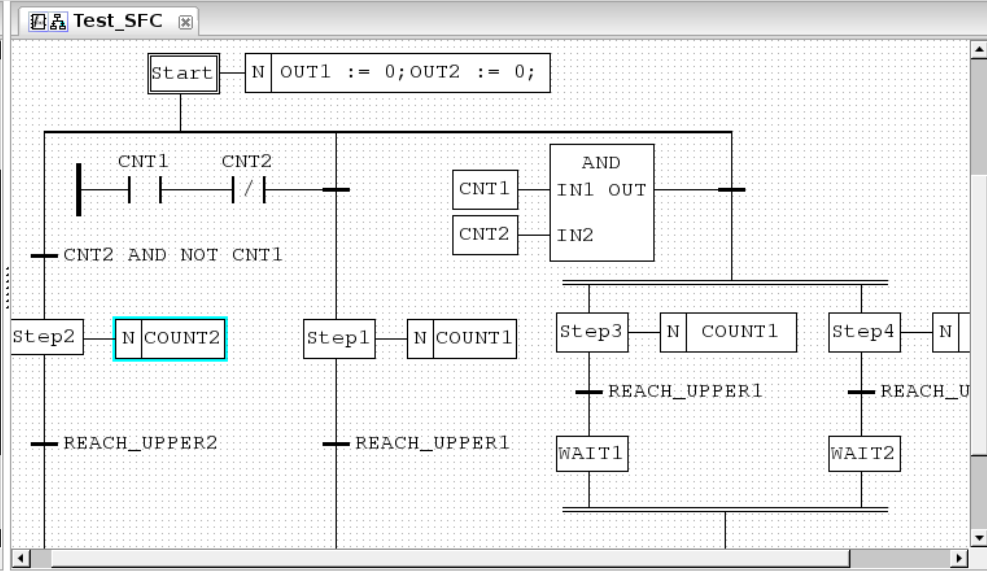
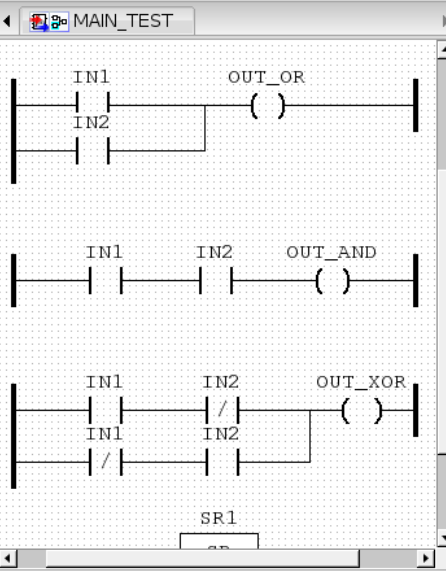
- STD_TEST
 - Properties
 - Data Types
 - MYTYPE
 - MYTYPE2
 - MYTYPE3
 - MYTYPE4
 - MYTYPE5
 - Functions
 - GETBIT
 - SETBIT
 - Function Blocks
 - Bitwise_Block
 - Test_SFC
 - Transitions
 - REACH_UPP
 - REACH_UPP
 - Actions
 - COUNT1
 - COUNT2
 - Programs
 - MAIN_TEST
 - Configurations
 - STD_CONF
 - Resources
 - STD_RESSO

Tasks: Add Task Delete Task

Name	Single
STD_TASK	

Instances: Add Instance Delete Instance

Name	Type
MAIN_INSTANCE	MAIN_TEST



```

1
2 OUT2 := IN2 + 1;
3 TEST_ARRAY[0,0] := TEST_ARRAY[0,1] + 50;
4

```

Variable Panel

#	Name	Class	Type	Initial Value	Retain	Constant
1	CNT1	Input	BOOL		No	No
2	CNT2	Input	BOOL		No	No
3	IN1	Input	MYTYPE		No	No
4	IN2	Input	MYTYPE		No	No
5	UPPER1	Input	INT		No	No
6	UPPER2	Input	INT		No	No
7	OUT1	Output	MYTYPE		No	No

Class Filter:

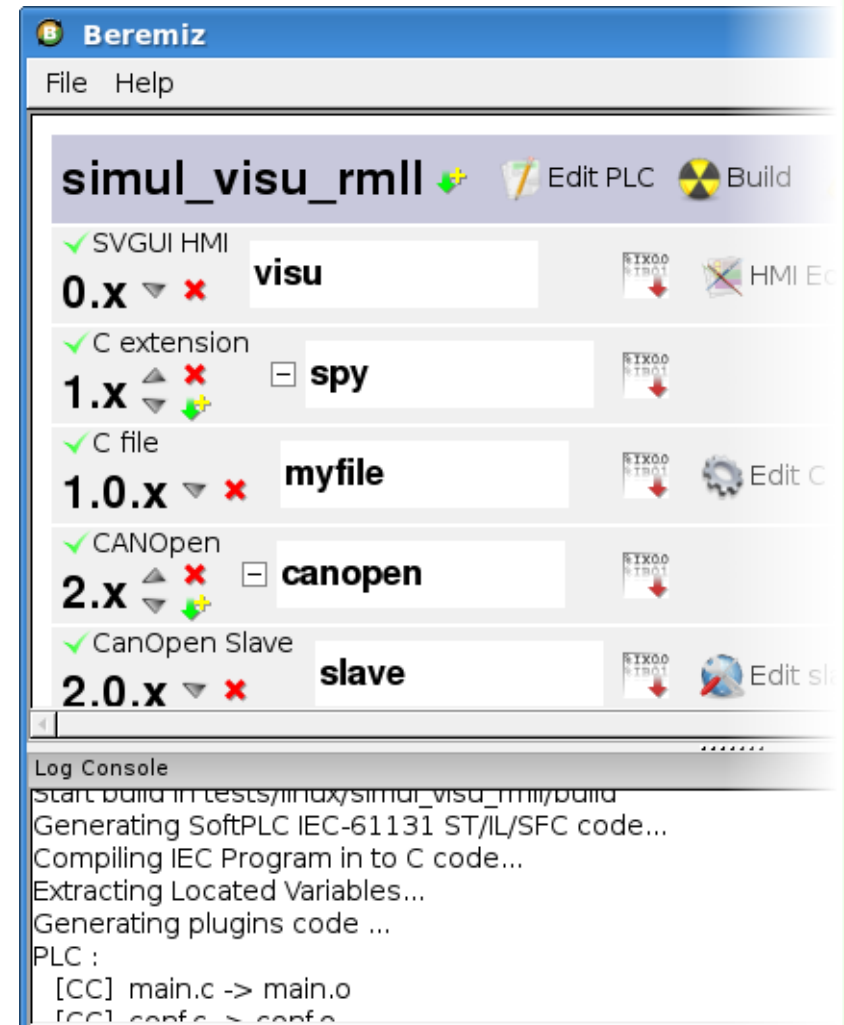
All

^ Add

v Delete

Gestion de projet

- Python + wxPython
- Arbre à greffons (plugins)
 - Hiérarchie IEC-61131
- Génération du runtime C
- Outils de mise au point
 - Contrôle d'exécution
 - Monitoring



Toolchain IEC-61131



Target PLC configuration

Edit PLC
 Build
 Clean
 Run
 Stop
 Show IEC code
 Edit raw IEC code

BeremizRoot
 Compiler: gcc
 CFLAGS:
 Linker: g++
 LDFLAGS:
 Sync_Align_Ra: 50
 TargetType: Linux
 TargetType - Linux:
 Nice: 0
 Connection: Local
 Connection - Local:

IEC-61131-3
Textual
Languages
(ST, IL, SFC)

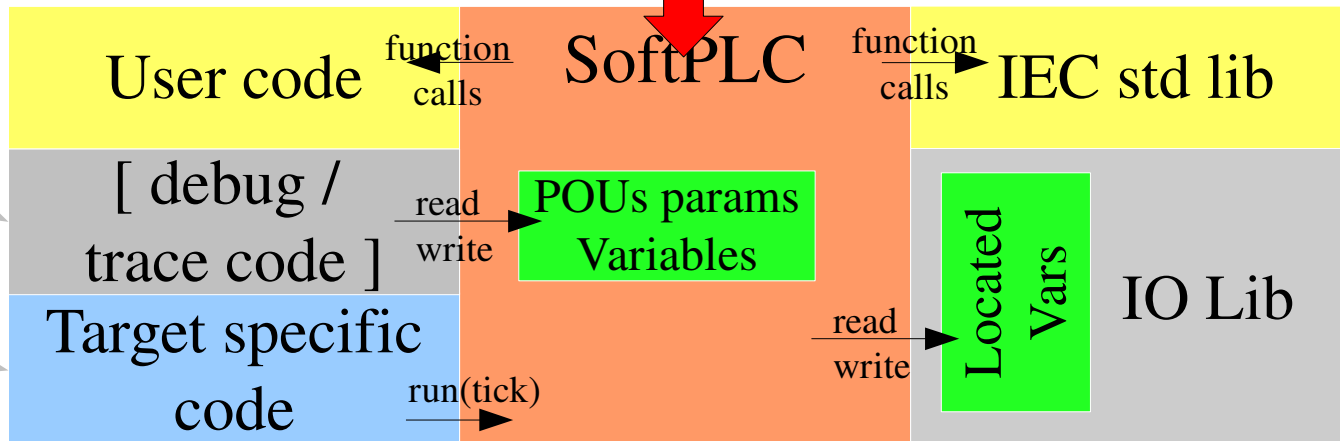


ANSI-C
code

C
compiler

Plugins

- SVGUI HMI
0.x **ihm**
- CANOpen
1.x **canopen**
- CanOpen Master
1.0.x **master**



Plugin CanFestival : Bus CANopen

- Création de maîtres
 - Déclaration des nœuds esclaves CANopen
 - Configuration du réseau selon les besoins du PLC
 - Transmissions synchrones et asynchrones
- Création d'esclaves
 - Déclaration des variables échangées
 - Exportation d'EDS (Electronic Data Sheet)



Networkedit - None

Network Edit Add Help

0x00 MasterNode	0x1C00-0x1FFF	Other Communication Parameters
0x10 DS401_Slave_GUI	0x2000-0x5FFF	Manufacturer Specific
0x20 GUI2	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 B
0x6206	Error Mode Outputs 8 Bit

subindex	name	type	value	access
0x00	Number of Input 8 bit	UNSIGNED8	1	Read O
0x01	Read Inputs 0x1 to 0x8	UNSIGNED8	0x00	Read O

Have Callbacks

struct REC.

PLCOpenEditor - plc

Edit Window Help

Project Tree

- autom_ism_r
 - Properties
 - Data Types
 - Functions
 - Function B
 - Programs
 - main
 - Configurat

main

full_in — full

empty_in — empty

Variable Panel

#	Name	Class	Type	Location
4	gauge	Local	Transform	
5	level	Local	INT	%IW1.0.3.8196.0
6	pump_out	Local	BOOL	%QX1.0.3.8192.0
7	sink_out	Local	BOOL	%QX1.0.3.8193.0
8	full_in	Local	BOOL	%IX1.0.3.8194.0
9	empty_in	Local	BOOL	%IX1.0.3.8195.0

Class Filter: All

Add Delete

Drag'n'drop des variables CANopen

Paramètres du plugin

CanFestivalInstance

CAN_Driver:

CanFestivalNode

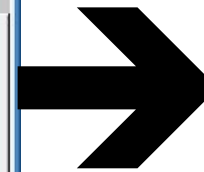
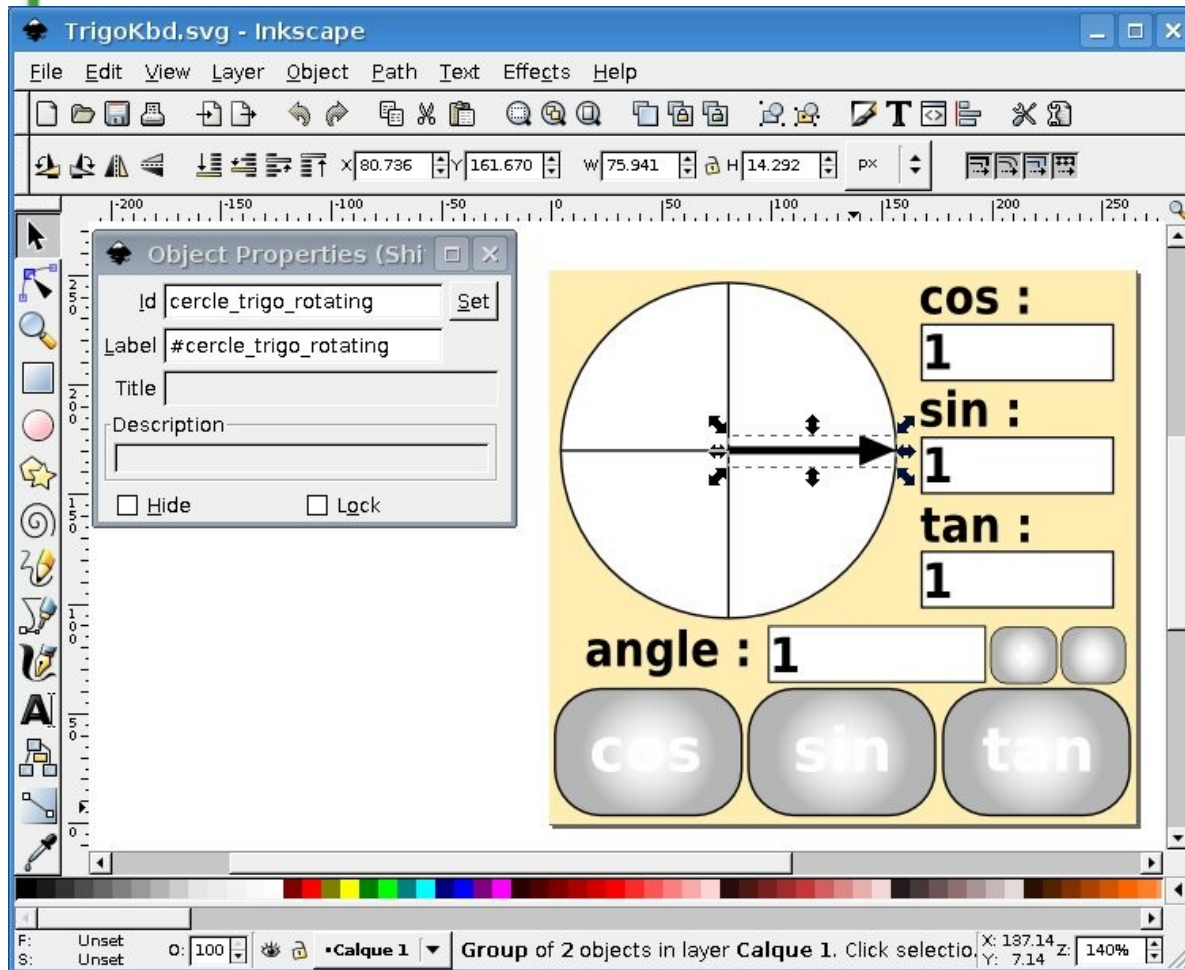
CAN_Device:

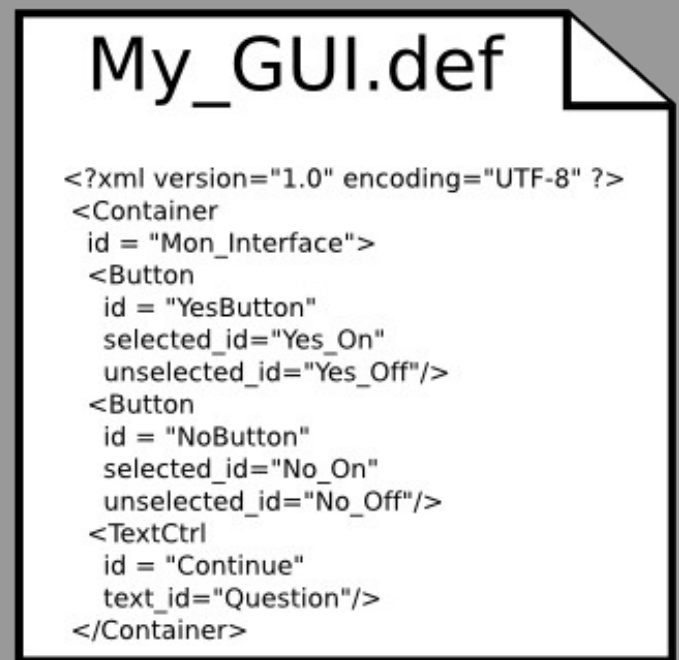
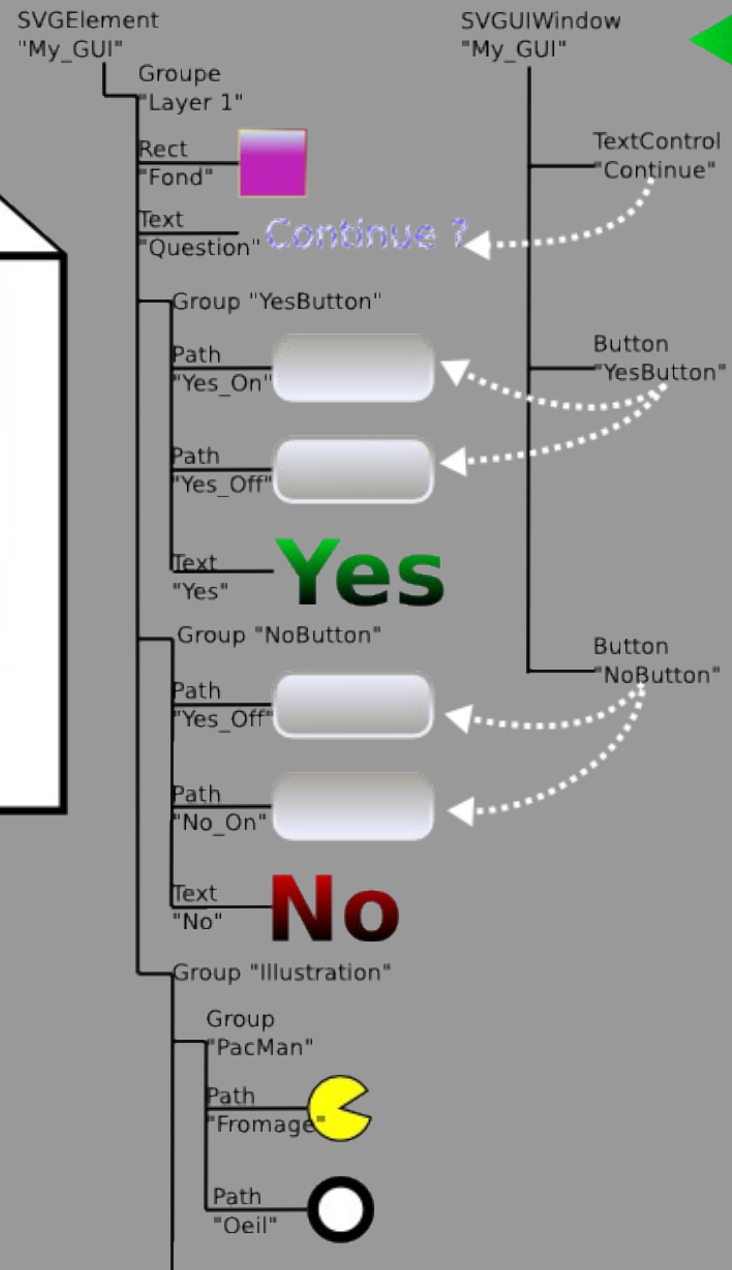
CAN_Baudrate:

NodeId:

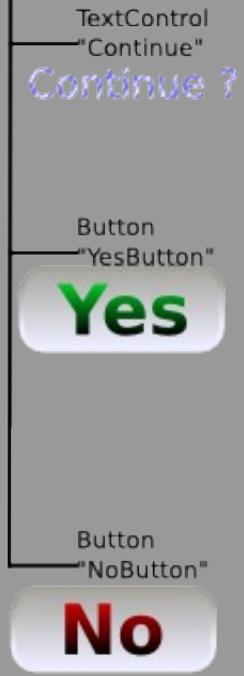
Sync_TPDOs:

Plugin SVGUI : Construction d'IHM



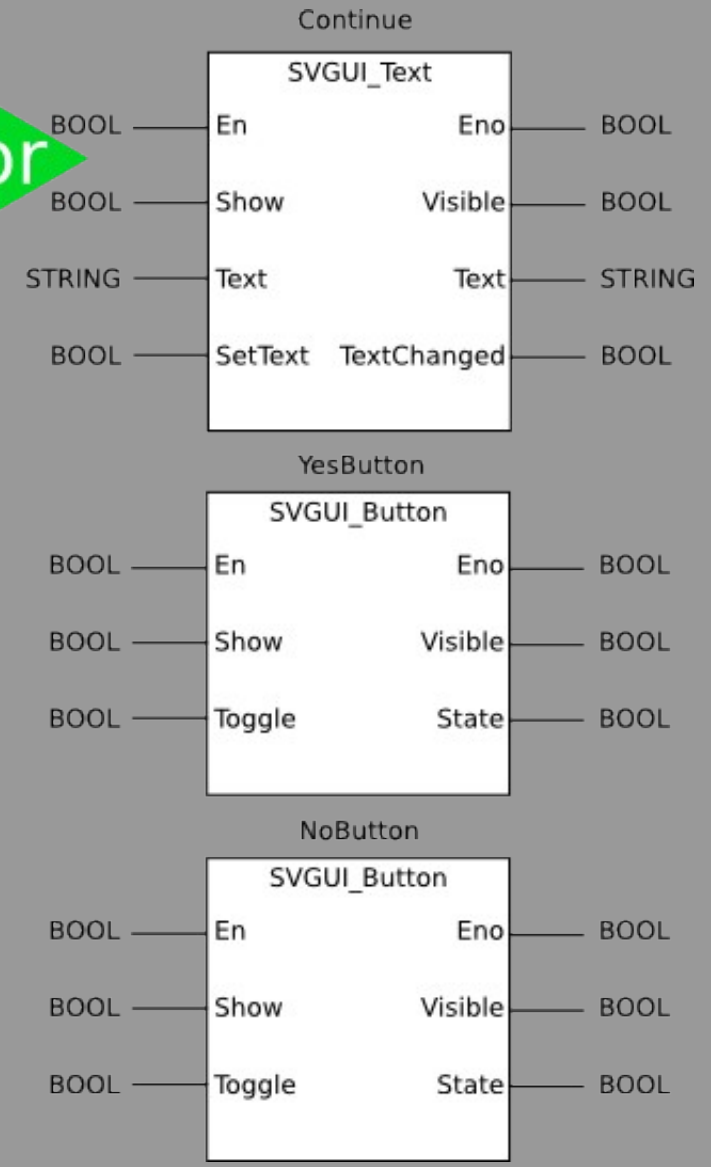


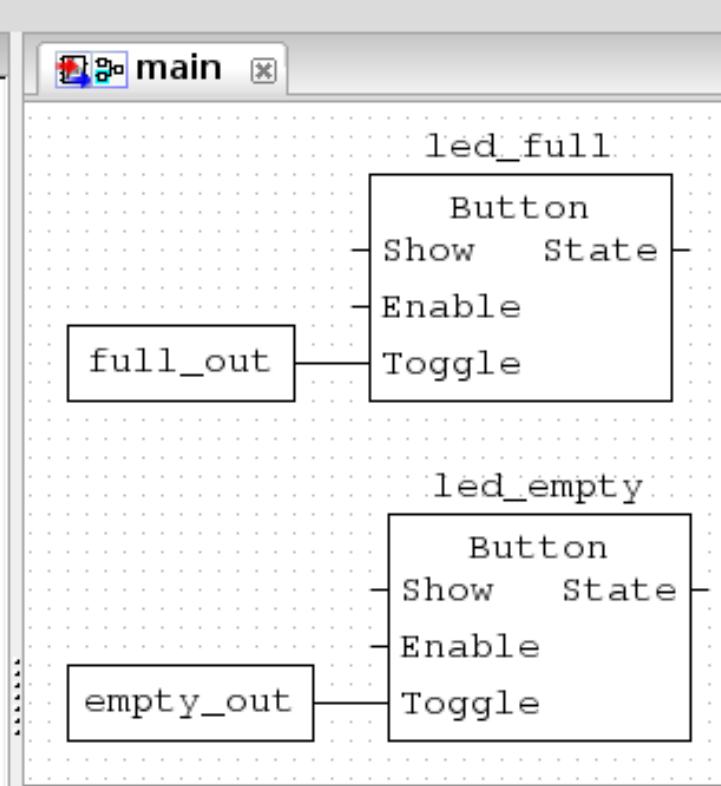
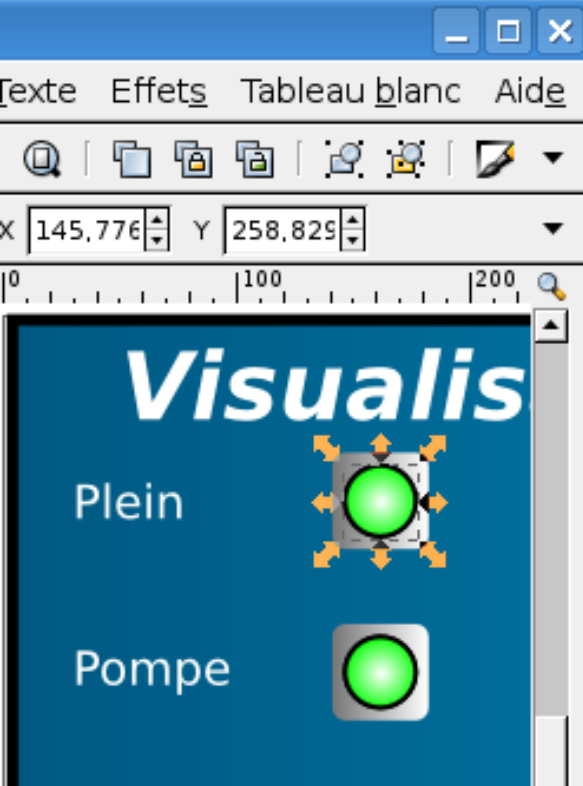
SVGUIWindow
"My_GUI"



```

My_GUI.def
<?xml version="1.0" encoding="UTF-8" ?>
<Container
  id = "Mon_Interface">
  <Button
    id = "YesButton"
    selected_id="Yes_On"
    unselected_id="Yes_Off"/>
  <Button
    id = "NoButton"
    selected_id="No_On"
    unselected_id="No_Off"/>
  <TextCtrl
    id = "Continue"
    text_id="Question"/>
</Container>
    
```

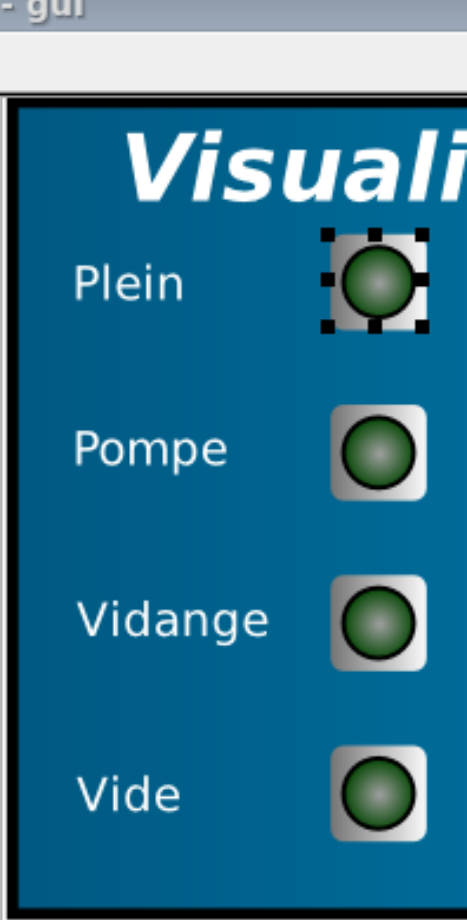




Edit

Interface Tree

- Interface
 - led_full
 - led_pump
 - led_sink
 - led_empty
 - movewater



gui.svg: Éditeur XML (Maj+Ctrl+X)

xml abc xml

```

  > <svg:text id="text3451">
  > <svg:text id="text3455">
  > <svg:text id="text3461">
  <svg:path id="led_full_on">
  <svg:path id="led_pompe_on">
  <svg:path id="led_vidange_off">
  <svg:path id="led_empty_off">
  <svg:path id="led_empty_on">
  <svg:path id="led_vidange_on">
  
```

Attribut	Valeur
id	led_full_on
sodipodi:cx	155.56349
sodipodi:cy	87.453148
sodipodi:r	15.152288

id

led_full_on

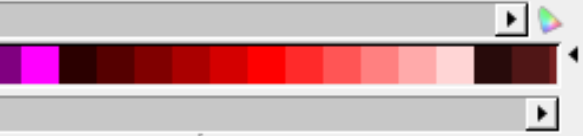
Définir

Attribut **id** sélectionné. Appuyer sur **Ctrl+Enter** après édition pour valider.

Editor Panel

Attribute	Value
name	led_full
enable	No
background_id	rect3479
toggle	Yes
selected_id	led_full_on
unselected_id	led_full_off

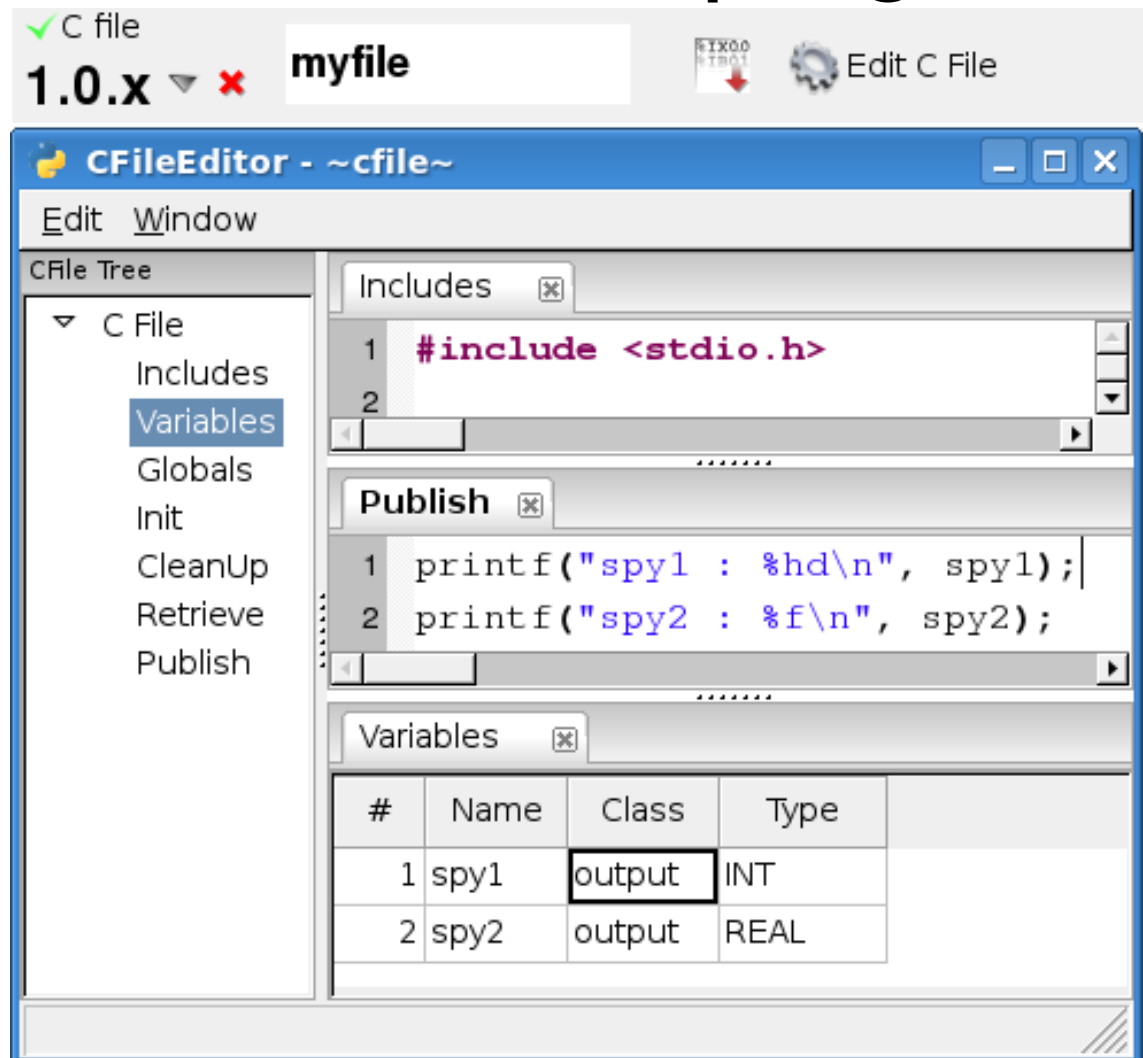
Pick up an element in the SVG View to set the v



magin : canopen.master (1, 0)
 [CC] OD_1_0.c -> OD_1_0.o
 Linking :

Plugin C_extension : votre plugin

- Ré-utilisation de code C/C++ existant
- Intégration de périphériques exotiques
- Accès direct aux ressources



The screenshot shows the CFileEditor interface with the following components:

- Top Bar:** Shows a green checkmark, "C file", version "1.0.x", a red 'x' icon, the filename "myfile", and an "Edit C File" button.
- CFileEditor - ~cfile~:** The main window title.
- Menu:** "Edit" and "Window".
- CFile Tree:** A tree view on the left with "C File" expanded, showing sub-items: "Includes", "Variables" (highlighted), "Globals", "Init", "CleanUp", "Retrieve", and "Publish".
- Includes:** A code editor showing:

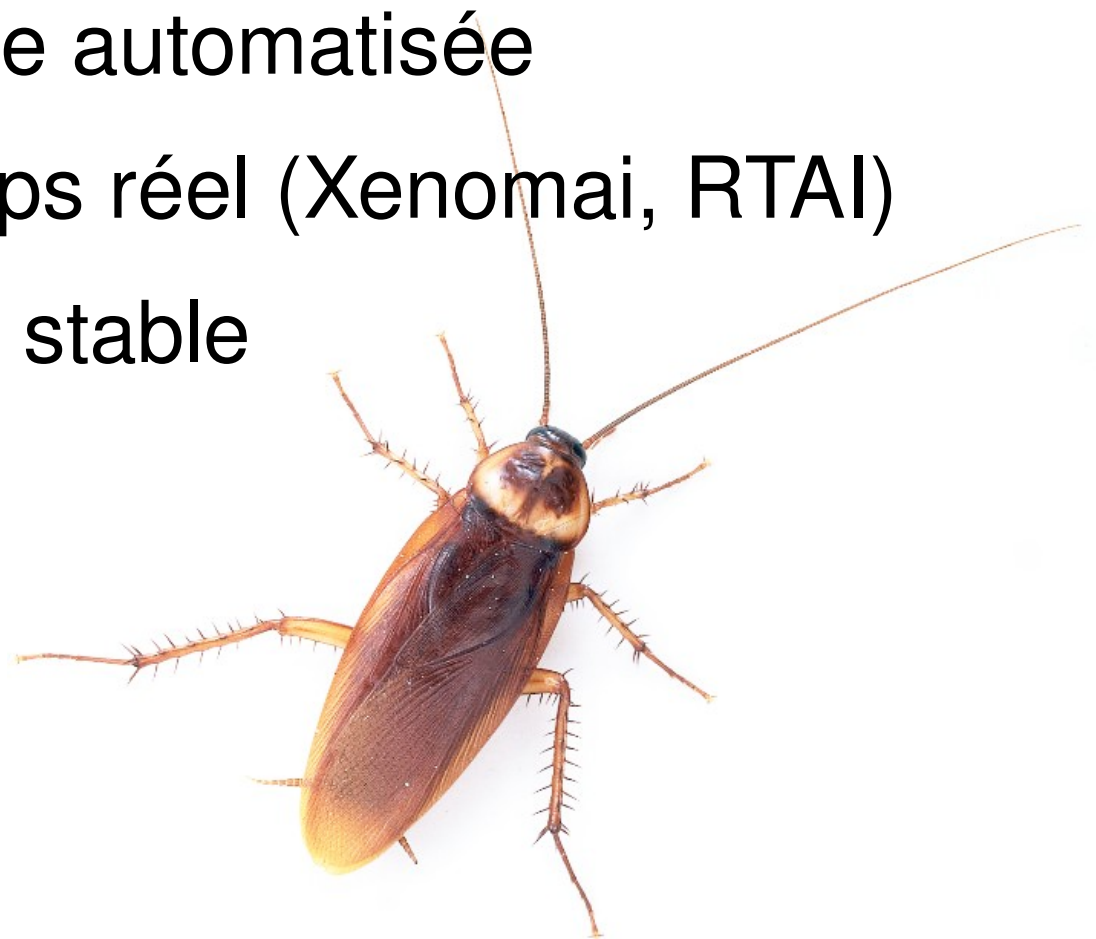

```
1 #include <stdio.h>
```
- Publish:** A code editor showing:


```
1 printf("spy1 : %hd\n", spy1);|
2 printf("spy2 : %f\n", spy2);
```
- Variables:** A table listing variables:

#	Name	Class	Type
1	spy1	output	INT
2	spy2	output	REAL

Etat d'avancement

- 04/08 - 1ère machine automatisée
- 07/08 - Support temps réel (Xenomai, RTAI)
- 08/08 - 1ère version stable



Plan de route

- 09/08 - Support commercial
- 10/08 - Compilation croisée, cibles sans OS



- xx/09 - PowerLink, ModBus, ...
- yy/09 - Analyse de code (GGCC, Framac-C,...)

CANopen Days 2008

Saint-Dié-des-Vosges



Journées portes ouvertes CANopen

Journées de formation à CANopen.
À Saint-Dié-des-Vosges, à 2h20 de Paris.
Les 9 et 10 septembre 2008.

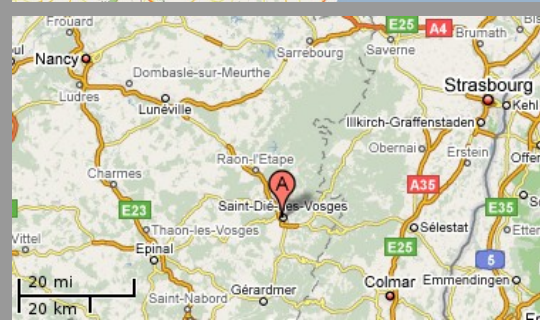
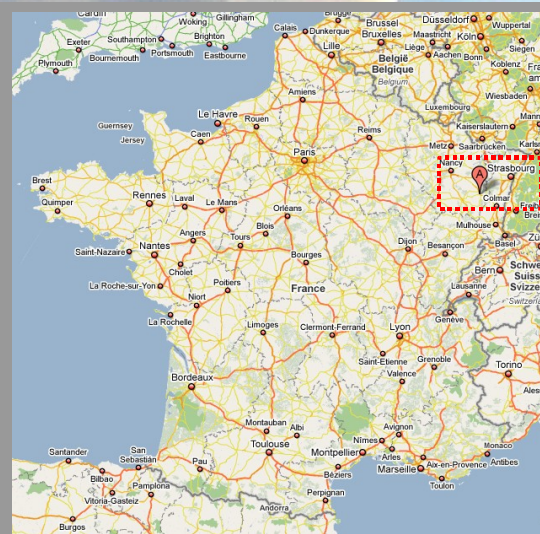
Programme :

- Initiation à CAN et CANopen.
- Ateliers de mise en pratique.
- Rencontres professionnelles.
- Expositions.

Objectifs :

- Comprendre et utiliser CANopen.
- Produire des nœuds CANopen.
- Choisir parmi les solutions CANopen.

Participation **gratuite**, réservation obligatoire :
canopendays@canfestival.org
<http://canopendays.canfestival.org/>



LOLITech
Logiciel Libre et Technologie

PEAK
System



FESTO



LeCroy

I.S.I.T

s a i n t
D    
DES-VOSGES