The Sardana System

How to export a control system (maybe)

Introduction

User View

Starting a procedure

A talk given by Jörg and Tiago at the WP10 ESRF UP 13/1/2011

Taurus as a toolkit for applications

Configure – don't program

How to write your own procedure

How to adapt it to your own hardware

Yesterday (2004)

Please build Synchrotron HERE

We have (had?) our experience

I like SPEC ♥

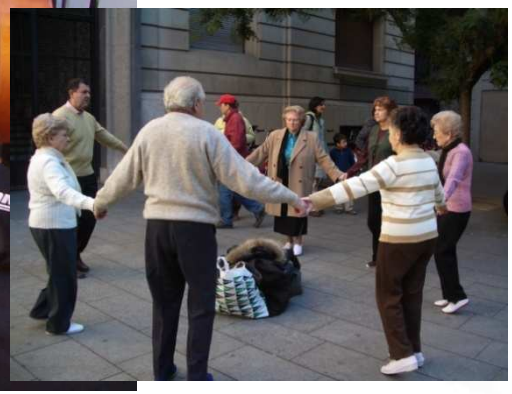
and I don't

- Well suited for large support groups
- Generic system (procedures) on all beam lines
- Involvement of users
- Easy prototyping

- IMPOSSIBLE(?) to collaborate / export
- Maintain a domain specific language
- Not well structured and maintainable

We chose a name

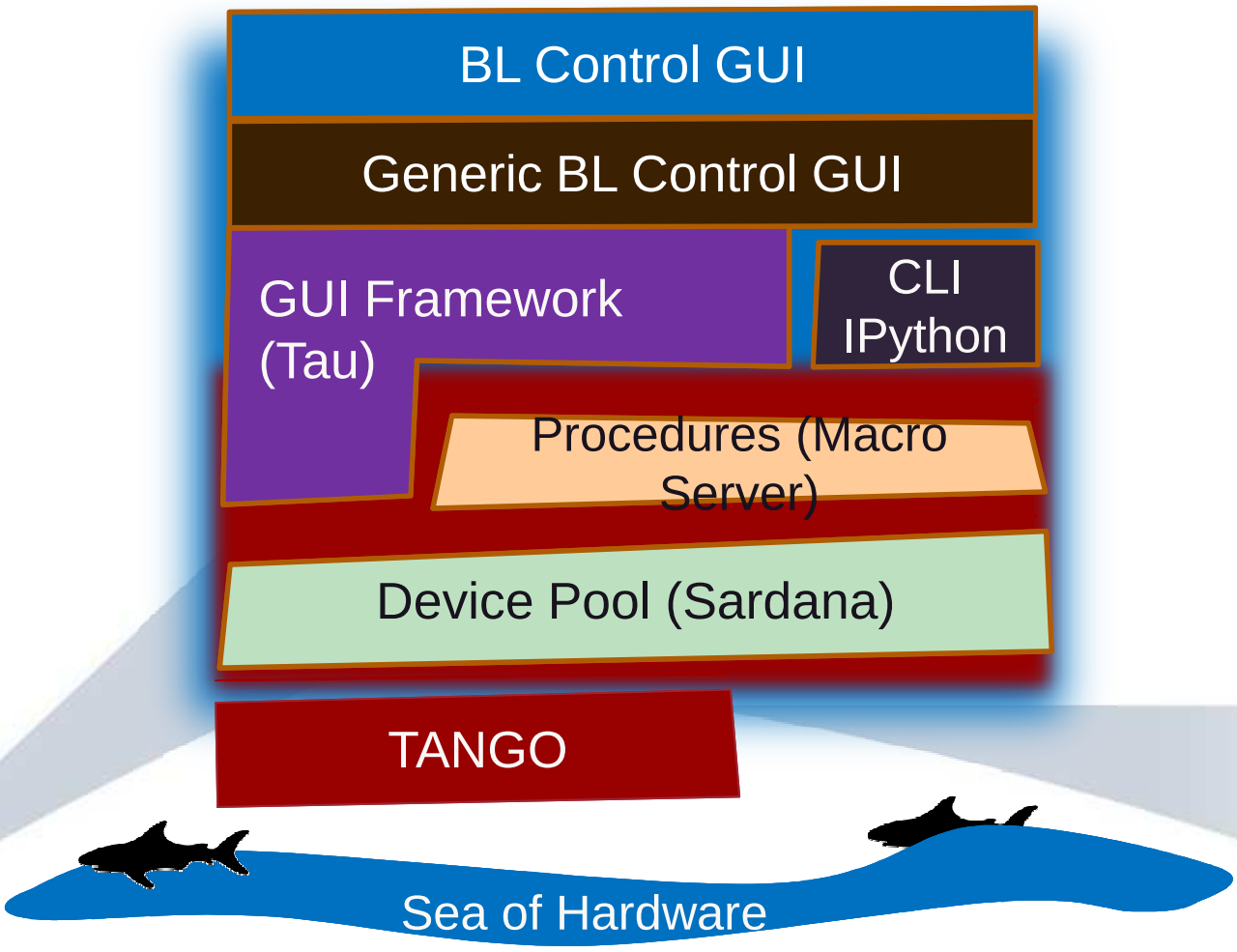
sardana



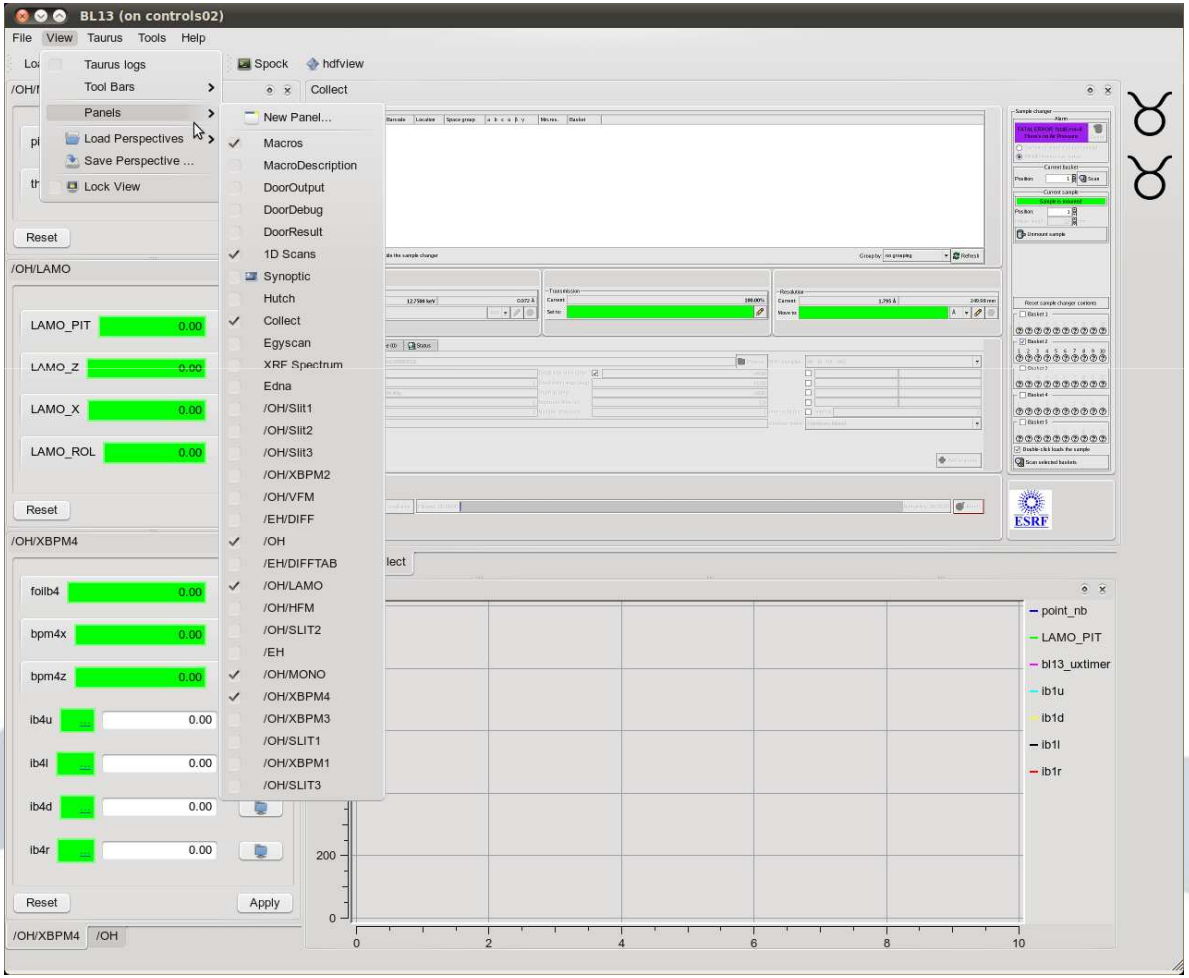
We selected some (great?) tools

- Mainly Qt, C++, Python are our tools
- Use TANGO (even if it hurts sometimes)

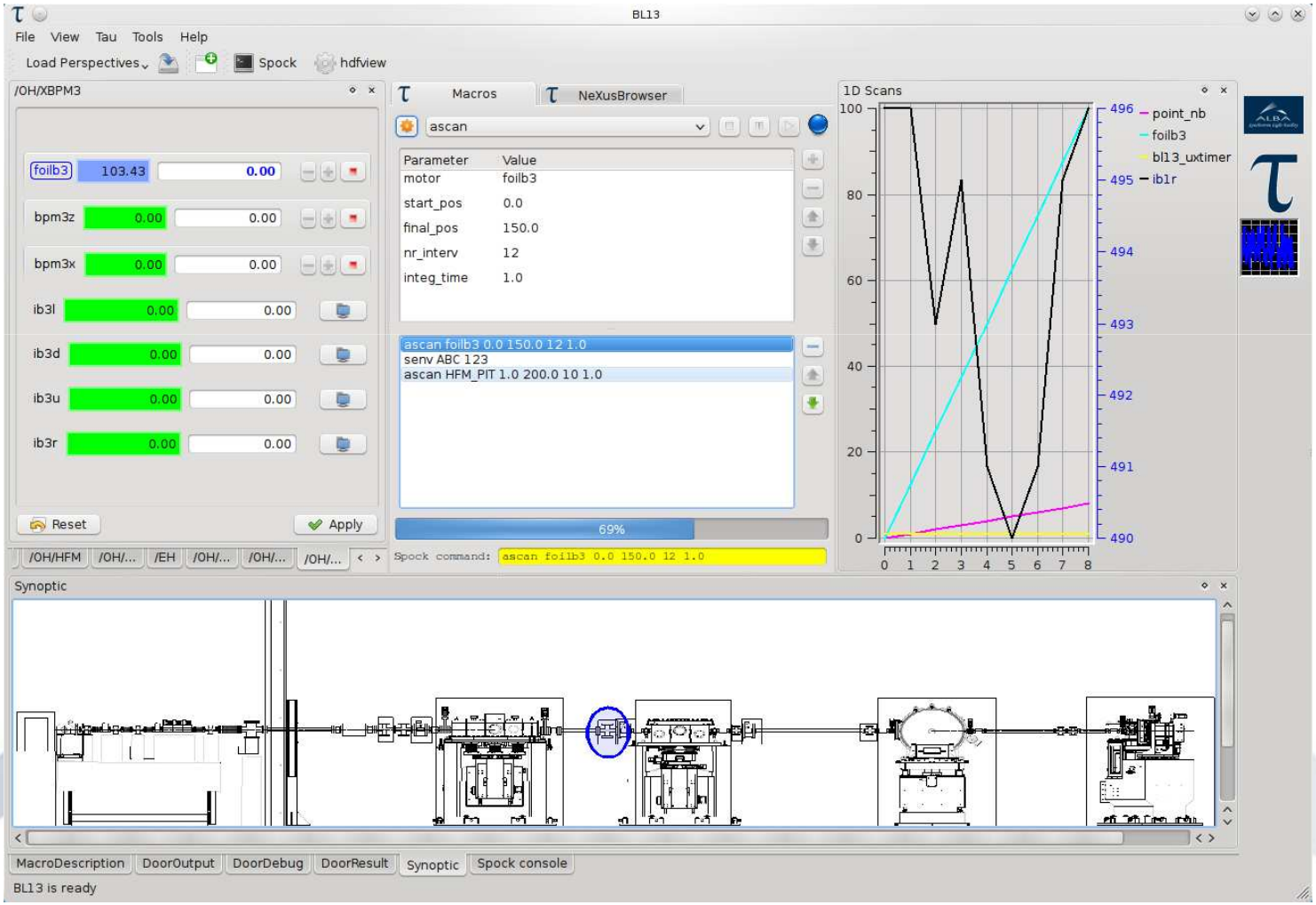
And we did (some) design ...



What do we sell to the user?



Standard (parts of a)BL GUI

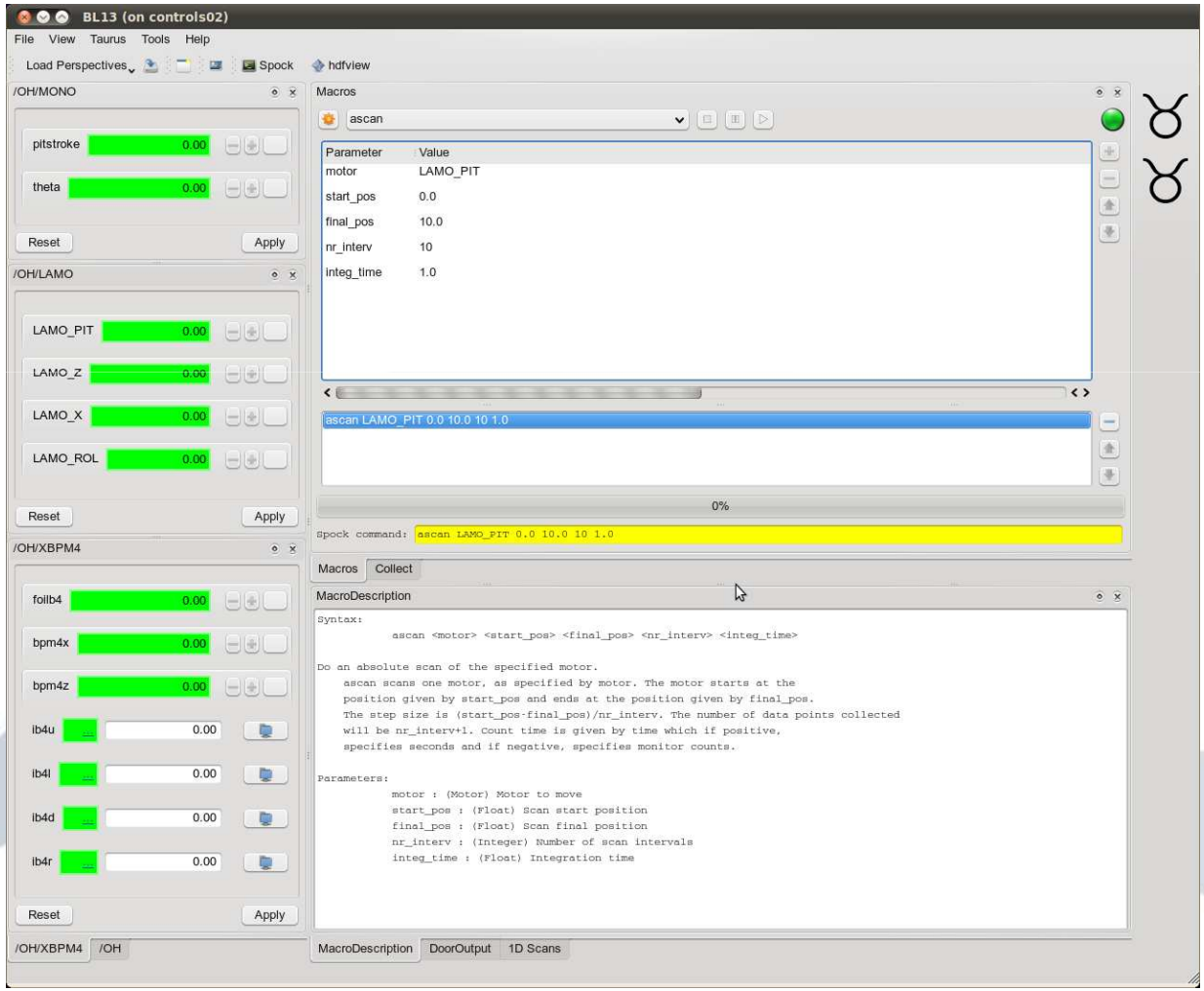


Starting a procedure

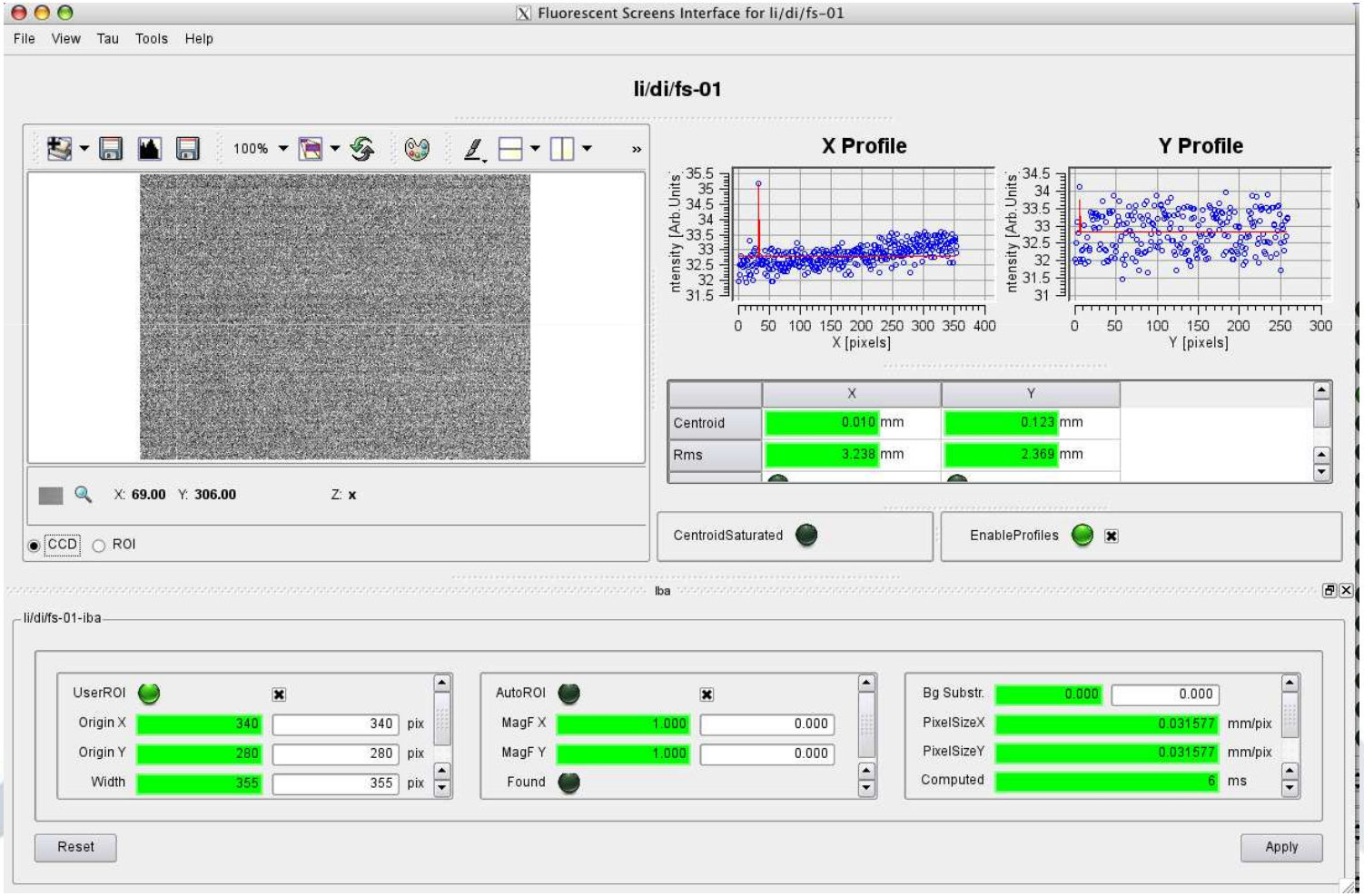
None - Terminator

```
tcoutinho@PC151:~$ spock -p BL98
Setting BL98 environment... [DONE]
Setting global environment... [DONE]
Connecting to door...
80 new macro(s) available

Spock 0.1.0 -- An interactive Macro Server client.

Running on top of Python 2.5.2 and IPython 0.9.1

Using Door BL98/Door/001 to access Macro Server BL98/MacroServer/001.

1.BL98: wm lt01
lt01_bending1     lt01_quadrupole1  lt01_quadrupole2  lt01_quadrupole3

1.BL98: wm lt01_quadrupole1
          lt01_quadrupole1
          lt01_quadrupole1
User
 High      1900.0 km
 Current    200.0 km
 Low      -1900.0 km
Dial
 High       1900.00
 Current     200.00
 Low       -1900.00

2.BL98: umv lt01_quadrupole1 100
lt01_quadrupole1
  100.0000

3.BL98: pdoc ascan
Class Docstring:
    Syntax:
            ascan  <motor> <start_pos> <final_pos> <nr_interv> <integ_time>

    Do an absolute scan of the specified motor

    Parameters:
            motor (Motor) - Motor to move
            start_pos (Float) - Scan start position
            final_pos (Float) - Scan final position
            nr_interv (Integer) - Number of scan intervals
            integ_time (Float) - Integration time
Calling Docstring:
    x.__call__(...) <==> x(...)

4.BL98: ascan lt01_quadrupole1 100 400 20 2.5
```

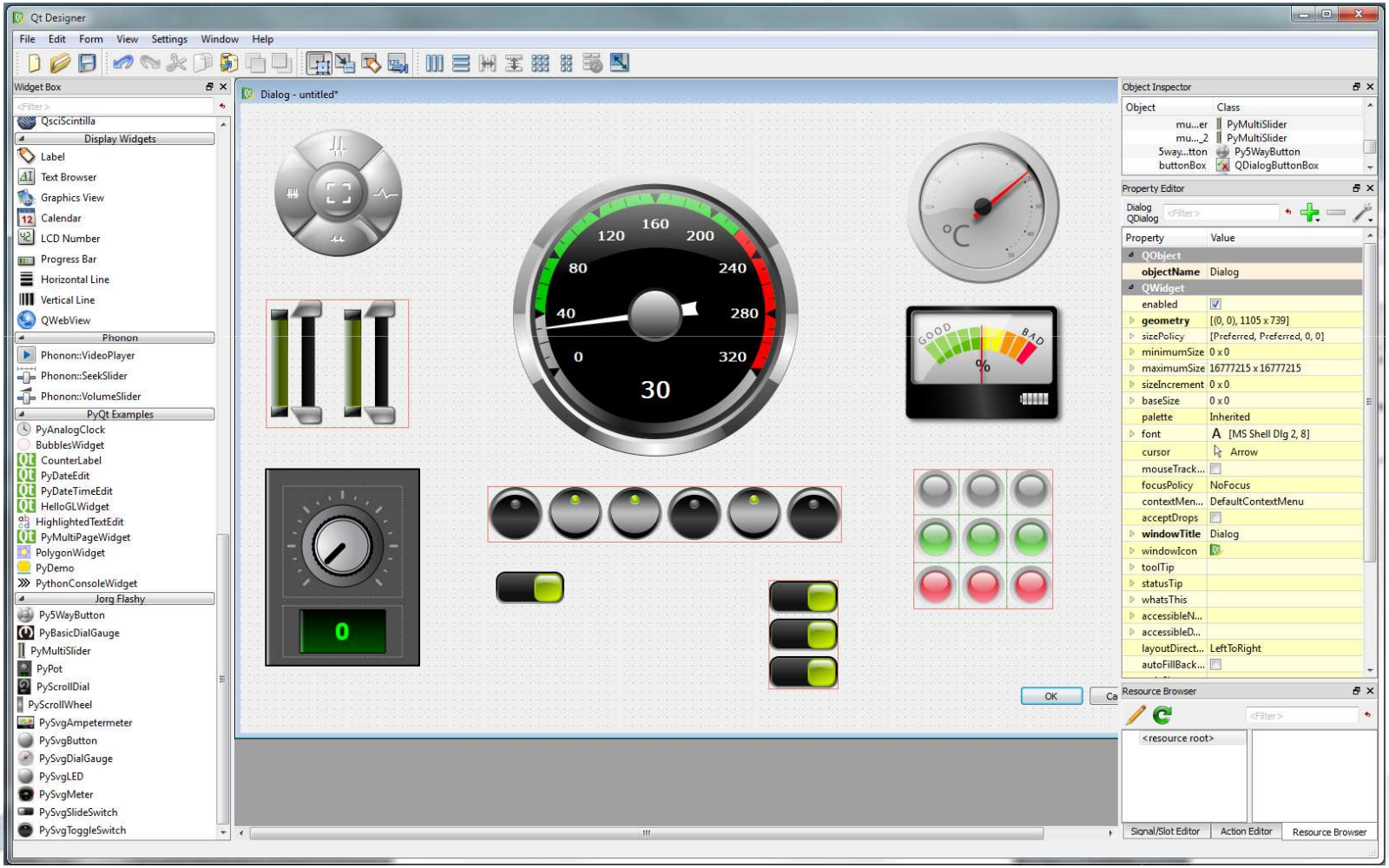
Start it from the GUI



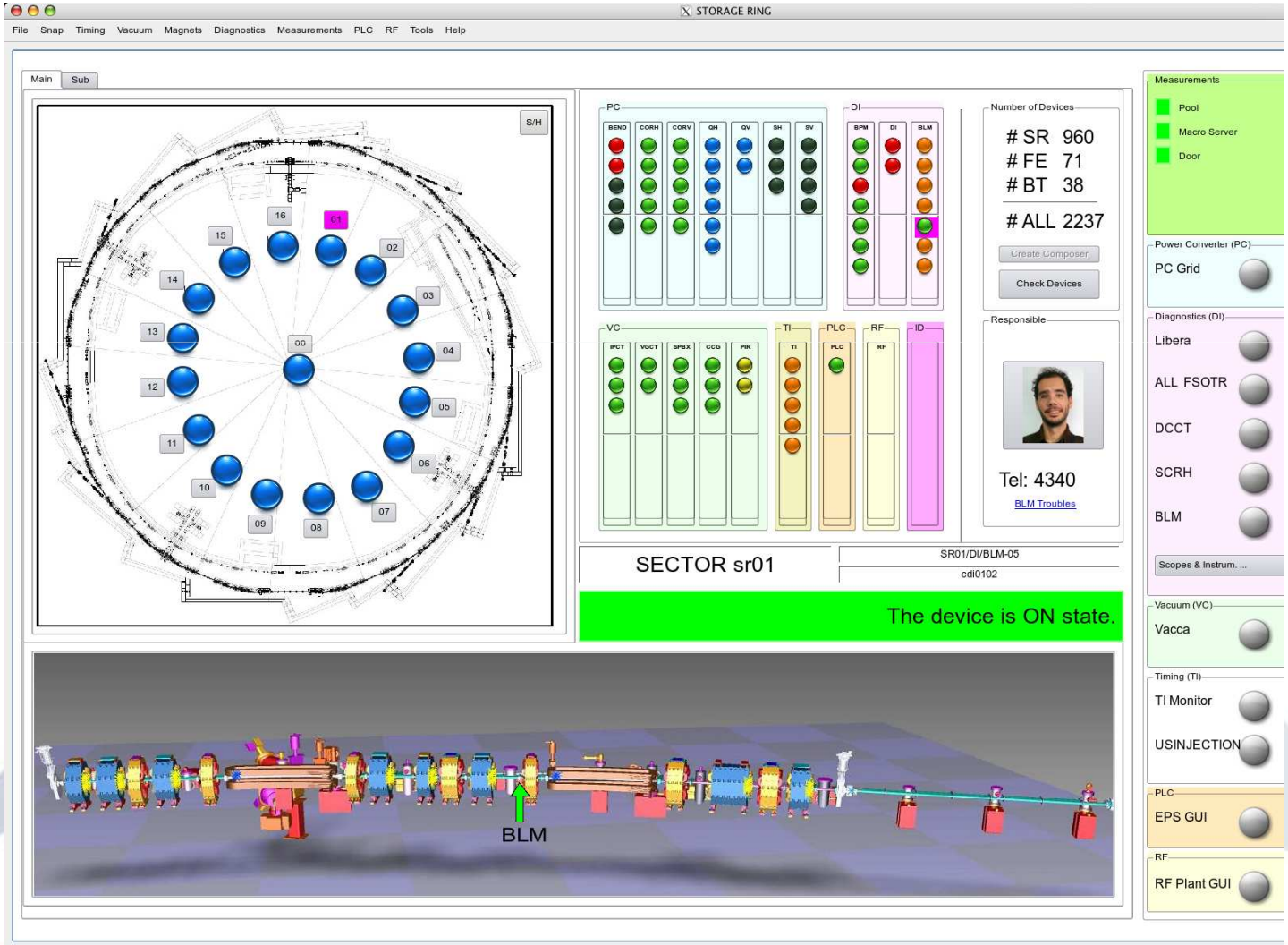
Generic ➔ Specific

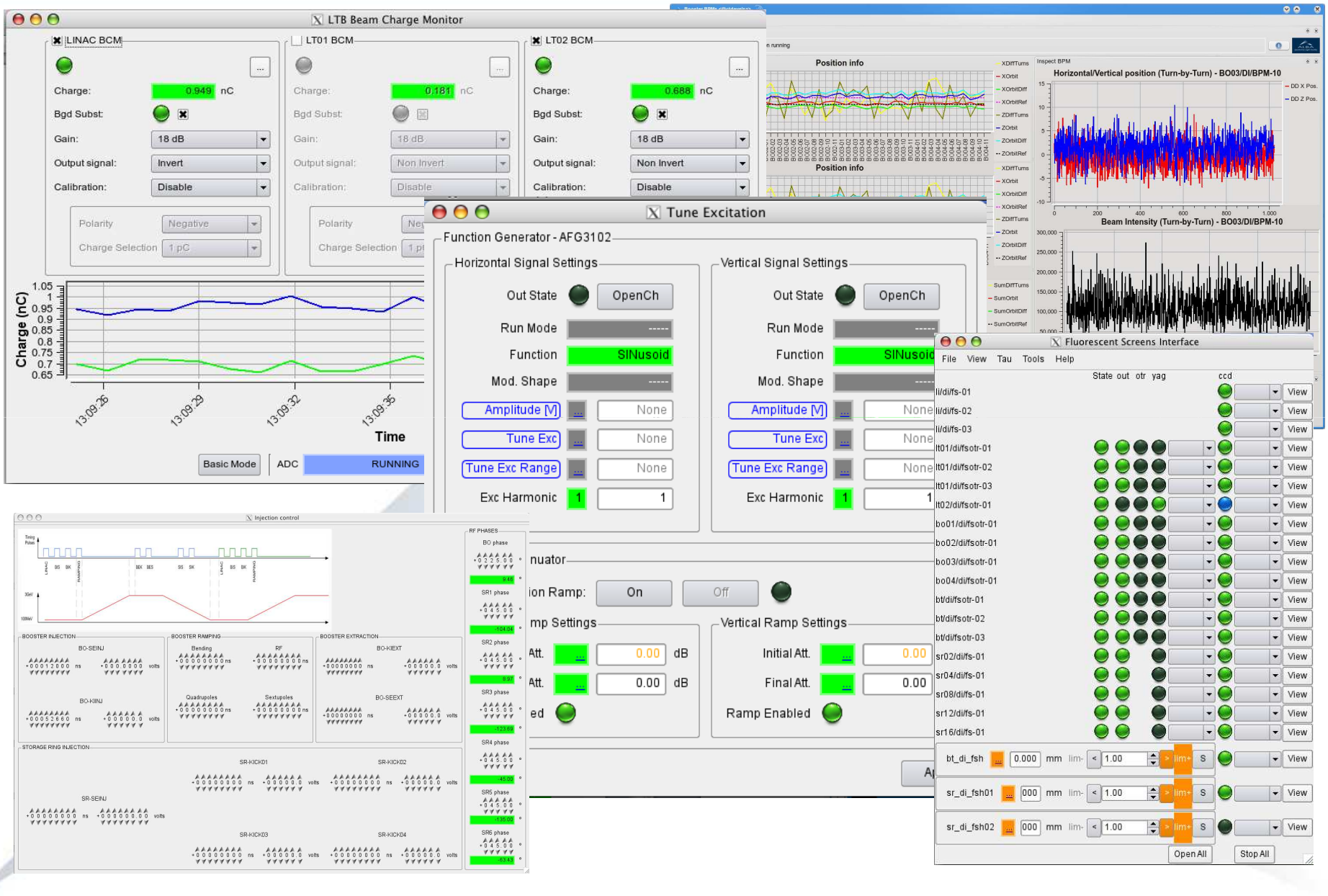


Built-in editor

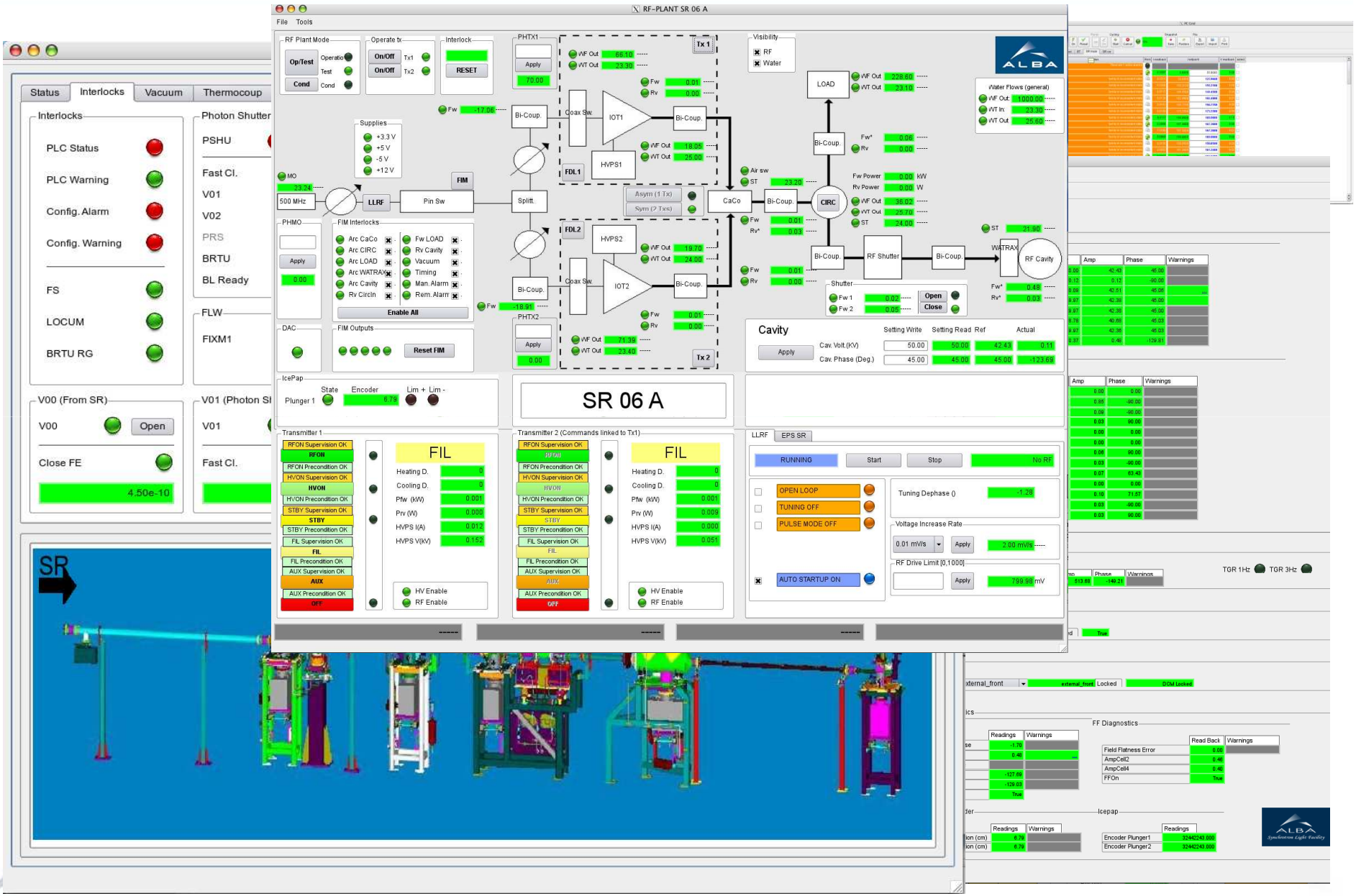


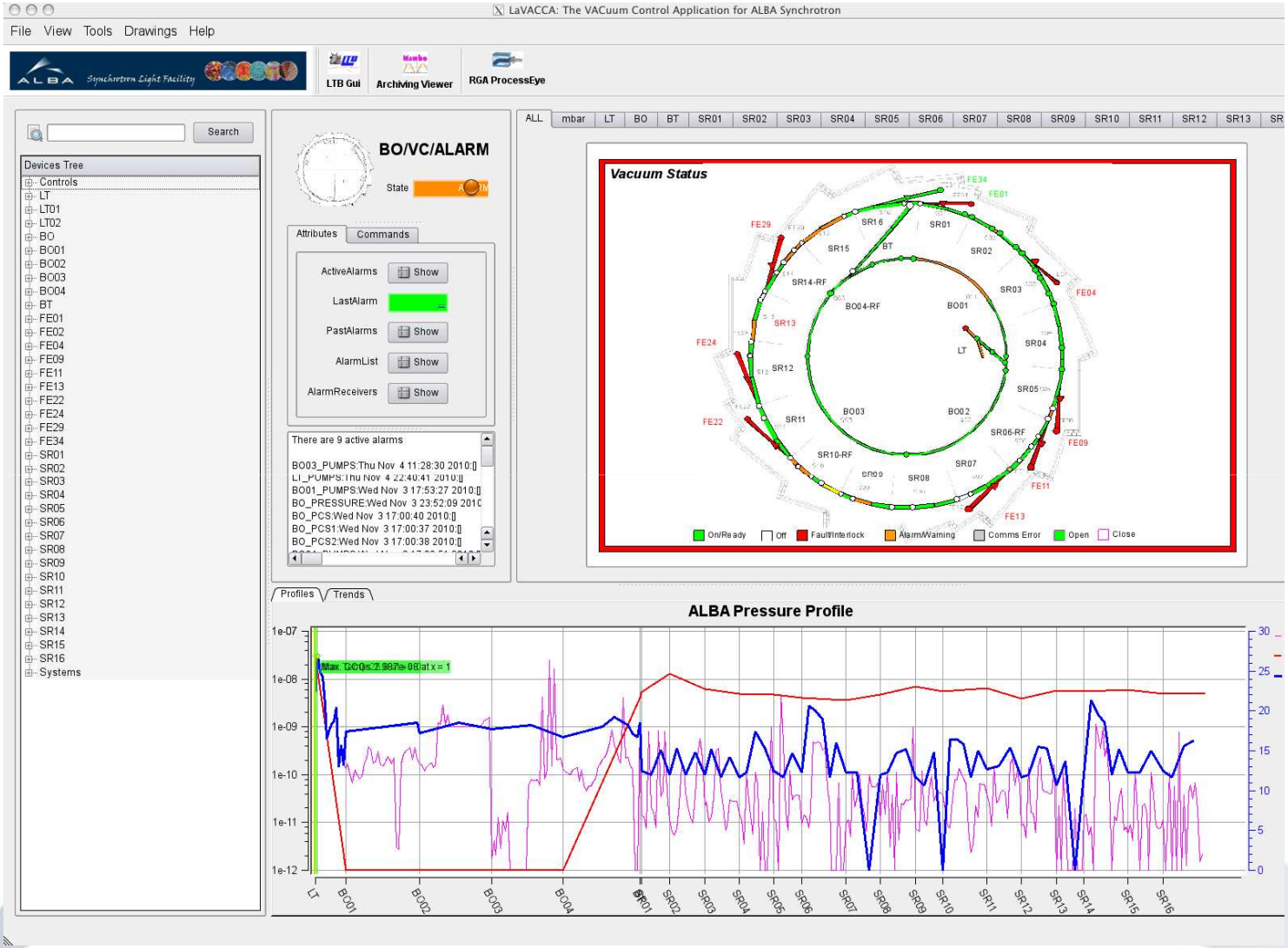
Taurus Widgets ➔ Applications



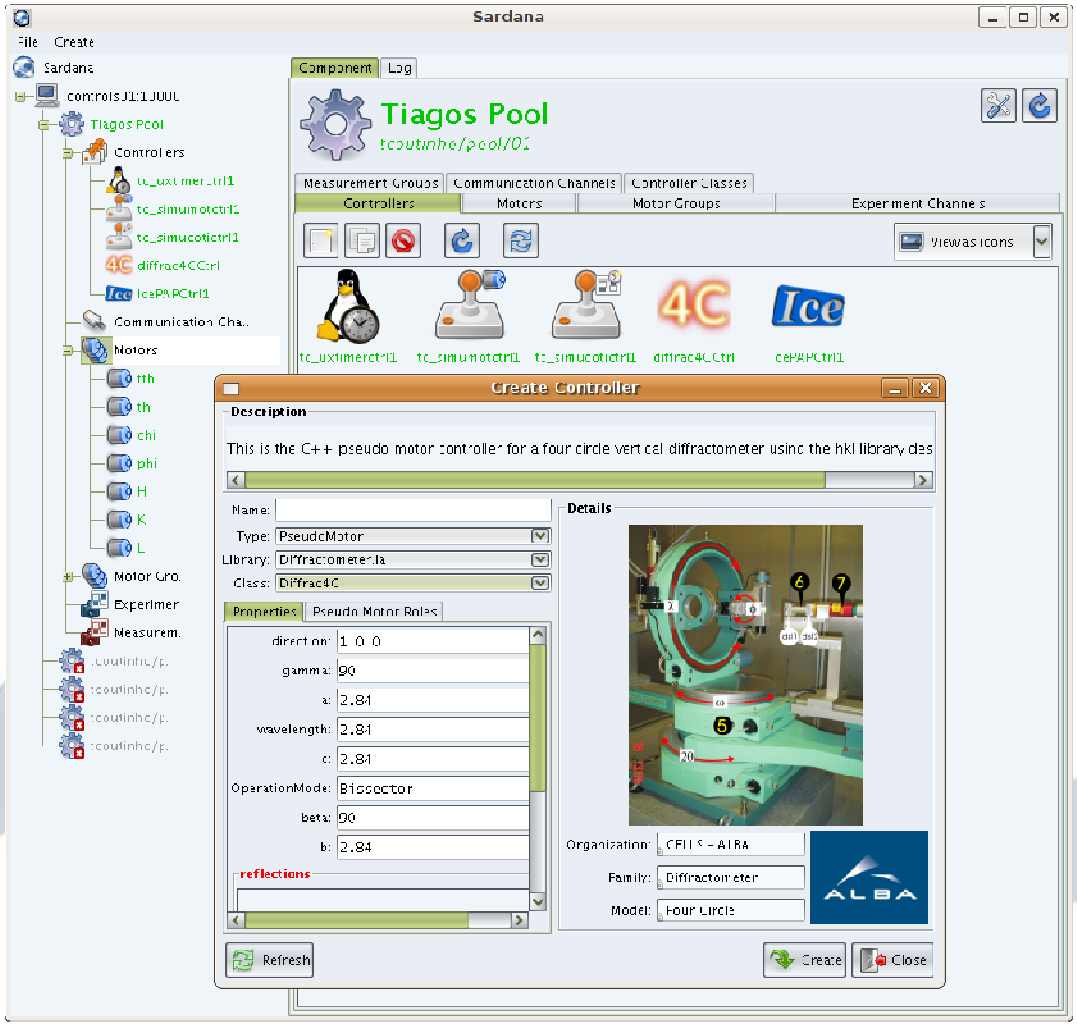

LTB Beam Charge Monitor
LINAC BCM
LT01 BCM
LT02 BCM
Charge:
0.949 nC
0.181 nC
0.688 nC
Bgd Subst:
Gain:
18 dB
Output signal:
Invert
Non Invert
Calibration:
Disable
Polarity
Negative
Charge Selection
1 pC
Charge (nC)
Time
13:09:26
13:09:29
13:09:32
13:09:35
Basic Mode
ADC
RUNNING
Tune Excitation
Function Generator - AFG3102
Horizontal Signal Settings
Vertical Signal Settings
Out State
OpenCh
Run Mode
Function
SINusoid
Mod. Shape
Amplitude [V]
Tune Exc
Tune Exc Range
None
Exc Harmonic
Vertical Ramp Settings
Initial Att.
Final Att.
Ramp Enabled
On
Off
Position info
Horizontal/Vertical position (Turn-by-Turn) - BO03/DI/BPM-10
Beam Intensity (Turn-by-Turn) - BO03/DI/BPM-10
Fluorescent Screens Interface
File View Tau Tools Help
State out otr yag ccd
li/di/fs-01
li/di/fs-02
li/di/fs-03
lt01/di/fsotr-01
lt01/di/fsotr-02
lt01/di/fsotr-03
lt02/di/fsotr-01
bo01/di/fsotr-01
bo02/di/fsotr-01
bo03/di/fsotr-01
bo04/di/fsotr-01
bt/di/fsotr-01
bt/di/fsotr-02
bt/di/fsotr-03
sr02/di/fs-01
sr04/di/fs-01
sr08/di/fs-01
sr12/di/fs-01
sr16/di/fs-01
bt_di_fsh
sr_di_fsh01
sr_di_fsh02
View
Open All
Stop All
Injection control
BOOSTER INJECTION
BOOSTER RAMPING
BOOSTER EXTRACTION
STORAGE RING INJECTION
RF PHASES

WP10 2011

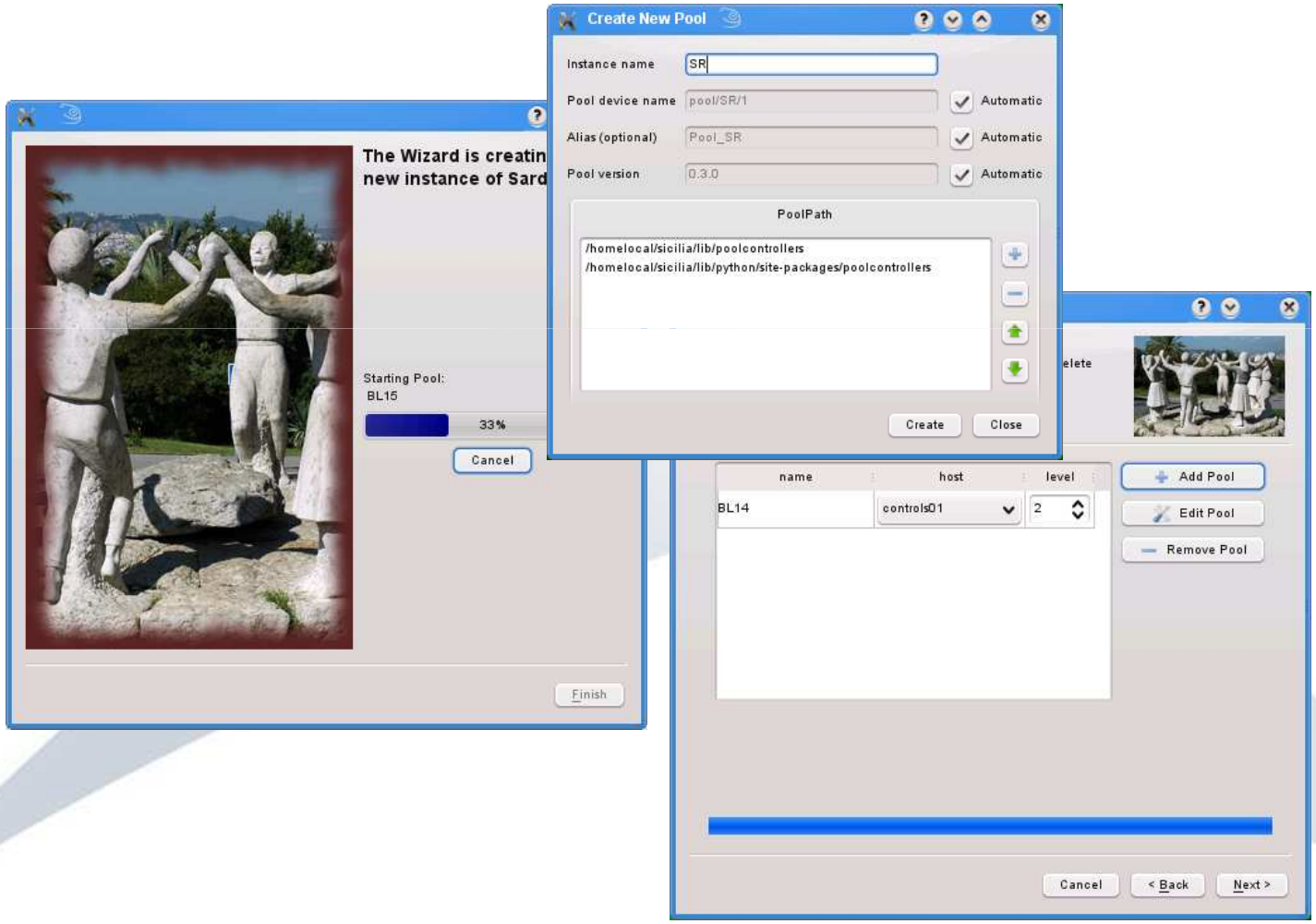

RF-PLANT SR 06 A
SR 06 A
Status
Interlocks
Vacuum
Thermocoup
PLC Status
PLC Warning
Config. Alarm
Config. Warning
FS
LOCUM
BRTU RG
Photon Shutter
PSHU
Fast Cl.
V01
V02
BRTU
BL Ready
V00 (From SR)
Open
Close FE
Cavity
FIL
SR

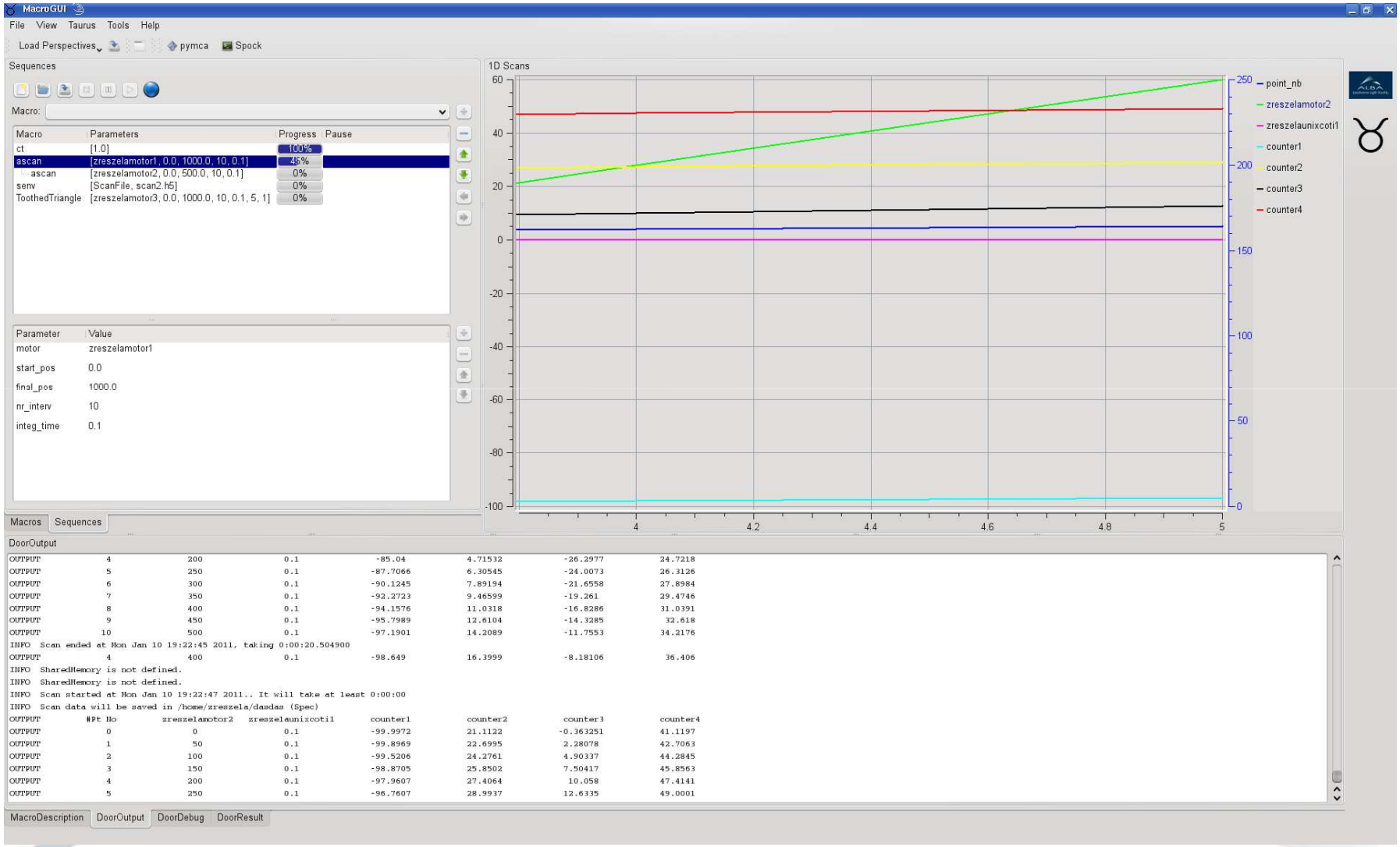

LaVACCA: The VACuum Control Application for ALBA Synchrotron
File View Tools Drawings Help
LTB Gui
Archiving Viewer
RGA ProcessEye
Search
Devices Tree
Controls
LT
LT01
LT02
BO
BO01
BO02
BO03
BO04
BT
FE01
FE02
FE04
FE09
FE11
FE13
FE22
FE24
FE29
FE34
SR01
SR02
SR03
SR04
SR05
SR06
SR07
SR08
SR09
SR10
SR11
SR12
SR13
SR14
SR15
SR16
Systems
BO/VC/ALARM
State
Attributes
Commands
ActiveAlarms
LastAlarm
PastAlarms
AlarmList
AlarmReceivers
Show
There are 9 active alarms
BO03_PUMPS:Thu Nov 4 11:28:30 2010:[]
BO01_PUMPS:Wed Nov 3 17:53:27 2010:[]
BO_PRESSURE:Wed Nov 3 23:52:09 201
BO_PCS:Wed Nov 3 17:00:40 2010:[]
BO_PCS1:Wed Nov 3 17:00:37 2010:[]
BO_PCS2:Wed Nov 3 17:00:38 2010:[]
ALL
mbar
LT
BO
BT
Vacuum Status
On/Ready
Off
Fault/Interlock
Alarm/Warning
Comms Error
Open
Close
Profiles
Trends
ALBA Pressure Profile

Configuration



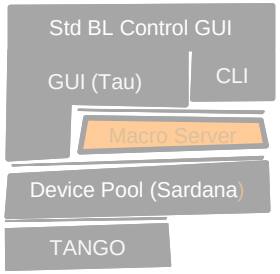
New version


Create New Pool
Instance name
SR
Pool device name
pool/SR/1
Automatic
Alias (optional)
Pool_SR
Automatic
Pool version
0.3.0
Automatic
PoolPath
/homelocal/sicilia/lib/poolcontrollers
/homelocal/sicilia/lib/python/site-packages/poolcontrollers
Create
Close
The Wizard is creatin
new instance of Sard
Starting Pool:
BL15
33%
Cancel
Finish
name
host
level
BL14
controls01
2
Add Pool
Edit Pool
Remove Pool
Cancel
< Back
Next >


MacroGUI
File View Taurus Tools Help
Load Perspectives
pymca
Spock
Sequences
Macro:
Macro Parameters Progress Pause
ct [1.0] 100%
ascan [zreszelamotor1, 0.0, 1000.0, 10, 0.1] 45%
ascan [zreszelamotor2, 0.0, 500.0, 10, 0.1] 0%
senv [ScanFile, scan2.h5] 0%
ToothedTriangle [zreszelamotor3, 0.0, 1000.0, 10, 0.1, 5, 1] 0%
Parameter Value
motor zreszelamotor1
start_pos 0.0
final_pos 1000.0
nr_interv 10
integ_time 0.1
1D Scans
point_nb
zreszelamotor2
zreszelaunixcoti1
counter1
counter2
counter3
counter4
Macros Sequences
DoorOutput
OUTPUT 4 200 0.1 -85.04 4.71532 -26.2977 24.7218
OUTPUT 5 250 0.1 -87.7066 6.30545 -24.0073 26.3126
OUTPUT 6 300 0.1 -90.1245 7.89194 -21.6558 27.8984
OUTPUT 7 350 0.1 -92.2723 9.46599 -19.261 29.4746
OUTPUT 8 400 0.1 -94.1576 11.0318 -16.8286 31.0391
OUTPUT 9 450 0.1 -95.7989 12.6104 -14.3285 32.618
OUTPUT 10 500 0.1 -97.1901 14.2089 -11.7553 34.2176
INFO Scan ended at Mon Jan 10 19:22:45 2011, taking 0:00:20.504900
OUTPUT 4 400 0.1 -98.649 16.3999 -8.18106 36.406
INFO SharedMemory is not defined.
INFO SharedMemory is not defined.
INFO Scan started at Mon Jan 10 19:22:47 2011.. It will take at least 0:00:00
INFO Scan data will be saved in /home/zreszela/dasdas (Spec)
OUTPUT #Pt No zreszelamotor2 zreszelaunixcoti1 counter1 counter2 counter3 counter4
OUTPUT 0 0 0.1 -99.9972 21.1122 -0.363251 41.1197
OUTPUT 1 50 0.1 -99.8969 22.6995 2.28078 42.7063
OUTPUT 2 100 0.1 -99.5206 24.2761 4.90337 44.2845
OUTPUT 3 150 0.1 -98.8705 25.8502 7.50417 45.8563
OUTPUT 4 200 0.1 -97.9607 27.4064 10.058 47.4141
OUTPUT 5 250 0.1 -96.7607 28.9937 12.6335 49.0001
MacroDescription DoorOutput DoorDebug DoorResult

Macro Server

- Central reusable procedures
- Controlled environment (edit, run) with a clean (tango) interface
- Procedures are python classes but this is hidden (one could write a macro server for another language, but we won't ….)
- Main programming is done in specific python classes not macro classes !!! Ex. Scan framework



Central Procedures

```
class mymv(Macro):
    """Move motor to position"""

    param_def = [
       ['motor', Type.Motor, None, 'Motor to move'],
       ['pos',   Type.Float, None, 'Position to move to']
    ]

    def run(self, motor, pos):
        motor.startMove(pos)
        motor.waitMove()
```

POINT 1

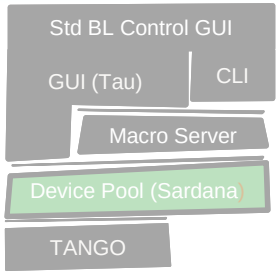
A macro is a class which inherits from Macro

POINT 2

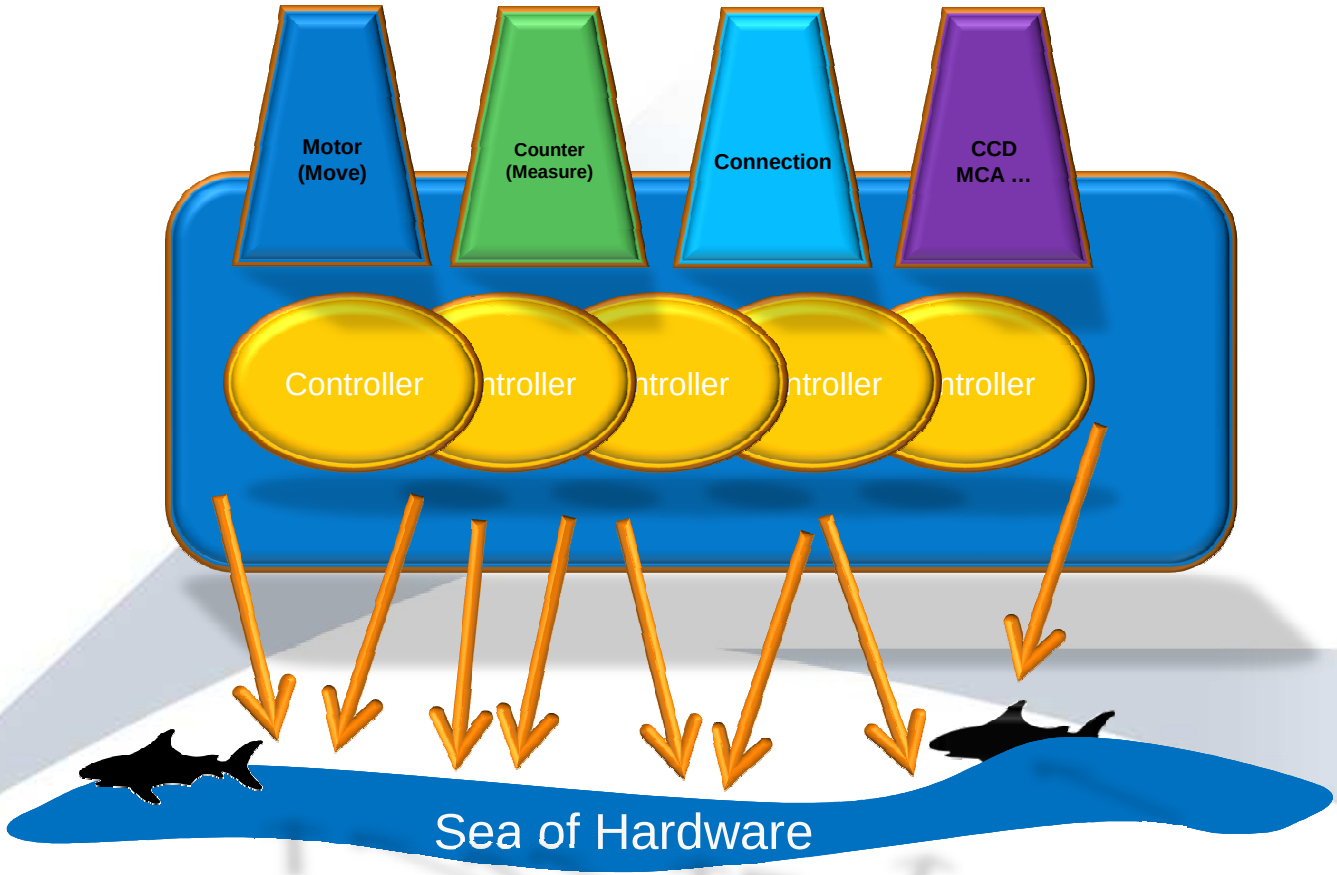
A macro has a description of itself and its parameters

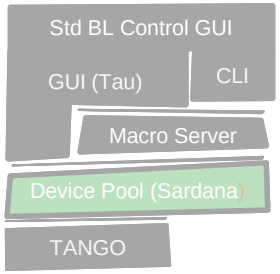
POINT 3

A macro has a method run which will be started in a separate thread



Device Abstraction

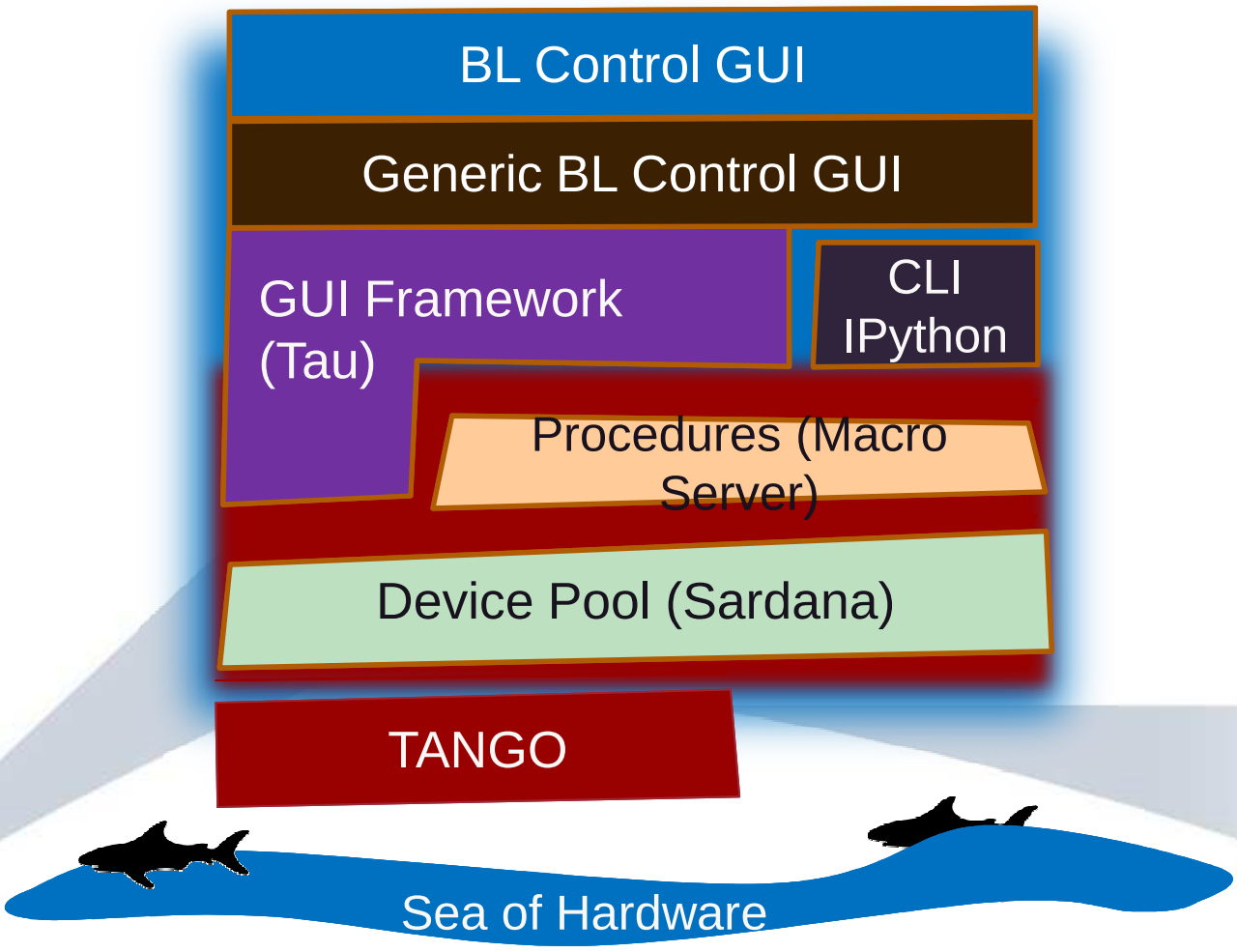




Device Abstraction and Pooling – Pseudo Devices

```
from PseudoMotorController import PseudoMotorController

class Slit(PseudoMotorController):
  """A Slit pseudo motor system """
  gender , model, organisation  = "Pseudo motor", "Slit", "CELLS - ALBA"
  image, logo  = "slit.png", "ALBA_logo.png"
  pseudo_motor_roles = ("Gap", "Offset")
  motor_roles = ("top blade", "bottom blade")

  class_prop = { 'resolution' : {
       'Description' : 'The encoder resolution',
       'Type' : 'PyTANGO.DevFloat,
       'DefaultValue' : 1.0 },
  }

  def calc_physical(self,index,pseudo_pos):
     half_gap = pseudo_pos[0]/2.0
     if index == 0:
       return pseudo_pos[1] - half_gap
     else:
       return pseudo_pos[1] + half_gap

  def calc_pseudo(self,index,physical_pos):
      if index == 0:
        return physical_pos[1] - physical_pos[0]
      else:
        return (physical_pos[0] + physical_pos[1])/2.0
```

Summary


BL Control GUI
Generic BL Control GUI
GUI Framework (Tau)
CLI IPython
Procedures (Macro Server)
Device Pool (Sardana)
TANGO
Sea of Hardware

Acknowledgements

- **THANK YOU** - ESRF (Manu, Bixente, Alejandro, Gilles, Laurent, Mathias, Andy, Seb), DESY (Teresa, Thorsten), Elettra (Claudio), ALBA (David, Guifre, Zibi, …..) ….. ☺
- We promise to continue to work hard to improve the system – we have our in the project

- Continue work on configuration editor, BL GUIs, installation and documentation, continuous scans, debugging, …..