

Sardana – a Python Based Software Package for Building SCADA Applications

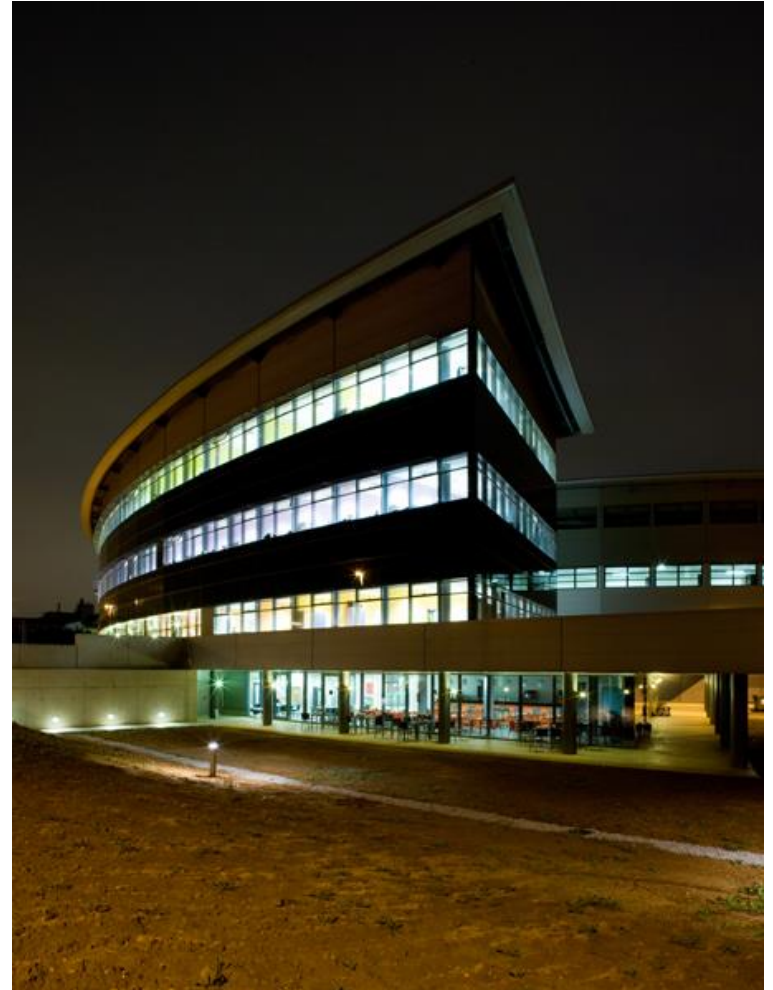
Guifre Cuní

Roberto Homs

Zbigniew Reszela

on behalf of the ALBA Controls Group and the Sardana Community

Tango Meeting 2015



1

Why Sardana?

2

Sardana demo

3

More advanced
topics...

4

Roadmap

Mission

- Produce supervision control and data acquisition software for modern scientific installations.
- Do not program if you do not need to

Sequencer
Macro edition &
Macro execution
environment

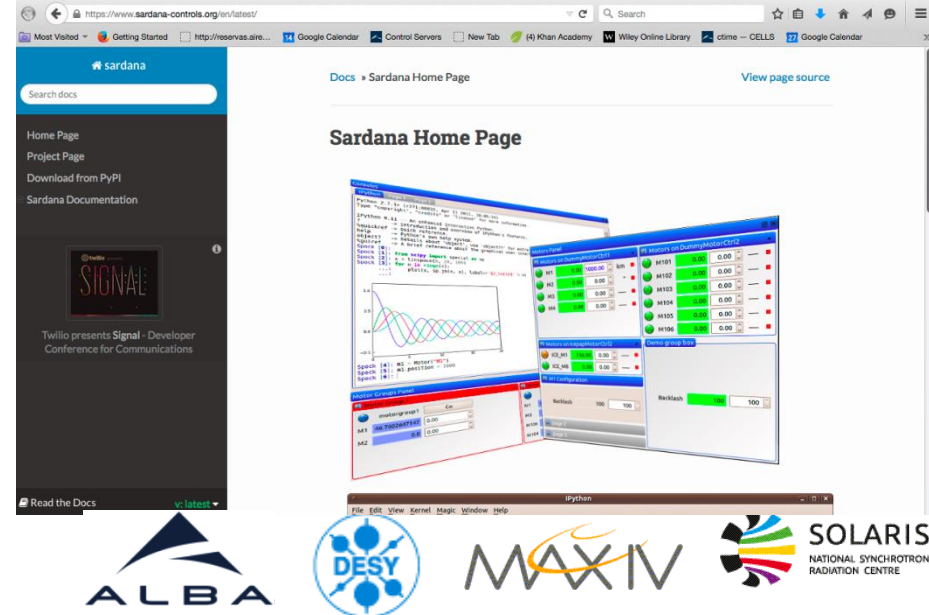
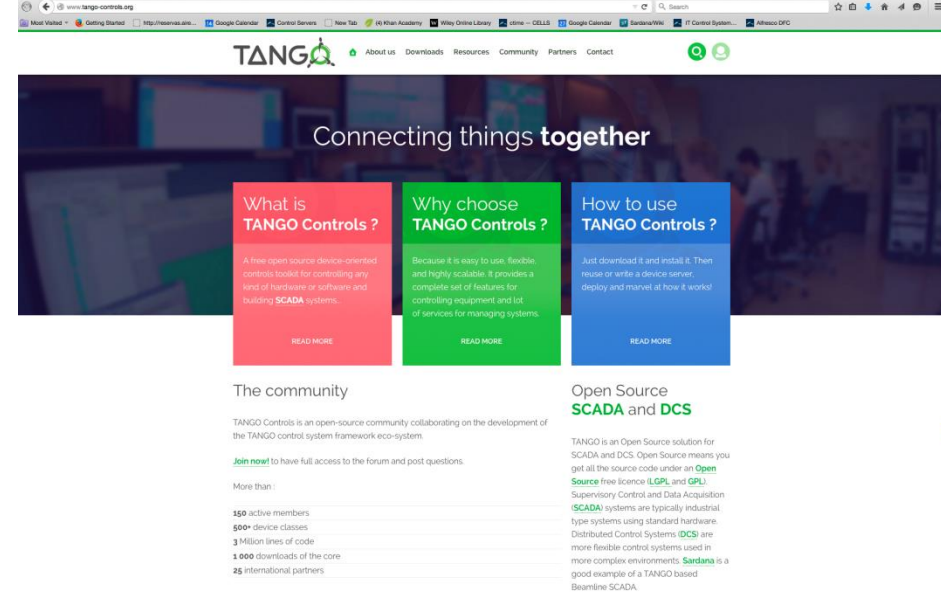
Resilient **Human machine interfaces:**
CLI (spock), GUI (Taurus)

↑ focus on **synchronization**, triggering

↑ Buffering

↑ Detectors, ↑ motors,

Archiving and Alarms integrated
Standardized data formats
Support for a large catalogue of hardware
Configuration centralized
Installation process simplified



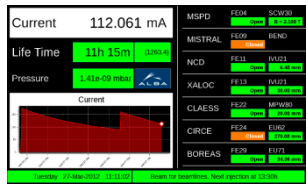
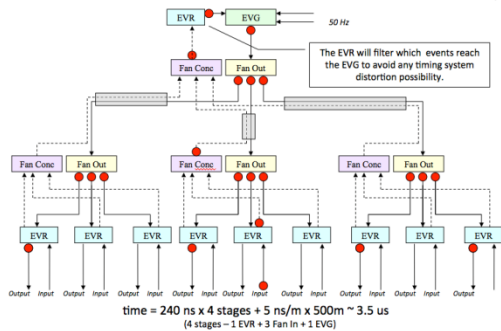
- Toolkit for building distributed control systems
 - Client - Server
 - Device servers managing hardware
 - Full support for Python, C++, Java,
 - Alarms, Archivers...
- 1 Control system for the Accelerators
- 1 Control system per Beamline

- Supervision, Control and Data Acquisition
 - Uses Tango, but has also schemes for others, such as SPEC or EPICS,
 - Sequencer and Macro execution environment
 - Tailored HMI:
 - SPOCK: Command Line Interface (IPython)
 - TAURUS: Graphical Interfaces
- Tailored for Experiment controls



Accelerators

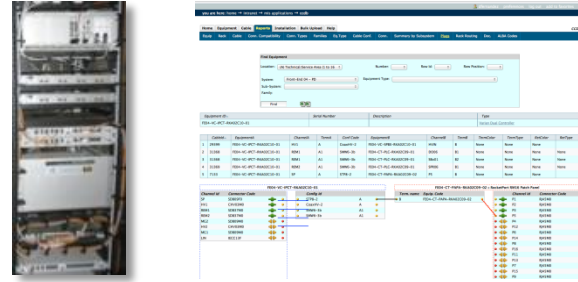
- 1 Tango Control system in the machine (~4500 devices)
- 1 PSS for the Machine (connected with all Beamlines)
- 1 EPS for the Machine (~ 7000 signals 150 CPU/RIO)
- Central **Cabling Database** where equipments cables and connections are stored. Variables and configuration automatically generated
- 1 MRF **Timing** system based on events and bidirectional (Synchronization and Fast Interlocks)
- Standard **Motion** controls including IDs



Personnel Safety System (PSS)



Equipment Protection System (EPS)



CCDB: Equipment Repository

Timing System (Event based)



Motion Controls
Electronics developments



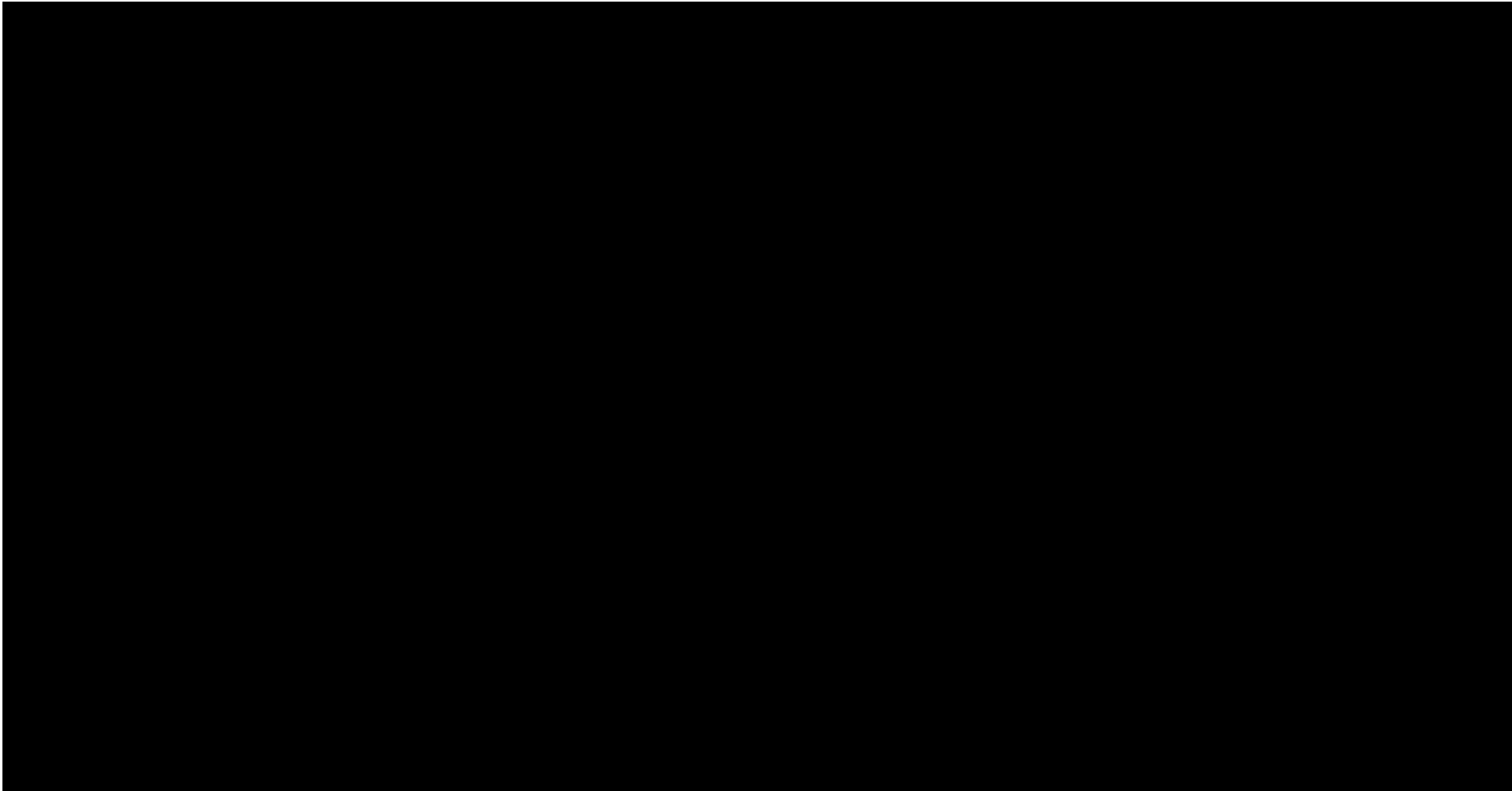
Beamlines

- 1 Tango Control per BL (~600 devices) Communication **BL-MACH** (Tango)
- 1 **PSS** for the BL (modular and extensible)
- 1 **EPS** per BL (~ 250 signals) communicates with the Machine Variables and configuration automatic from Cabling Database
- 1 **Sardana** installation per BL:
 - * Macros and GUIs shared between Beamlines
 - * Shared know how: Simple for Users move from BL to another Hardware & software reused between Beamlines and Machine when possible.
- Timing** system naturally distributed from the Machine to the Beamlines
- Standard **Motion** controls (among 700 axes in all Beamlines and Machine)



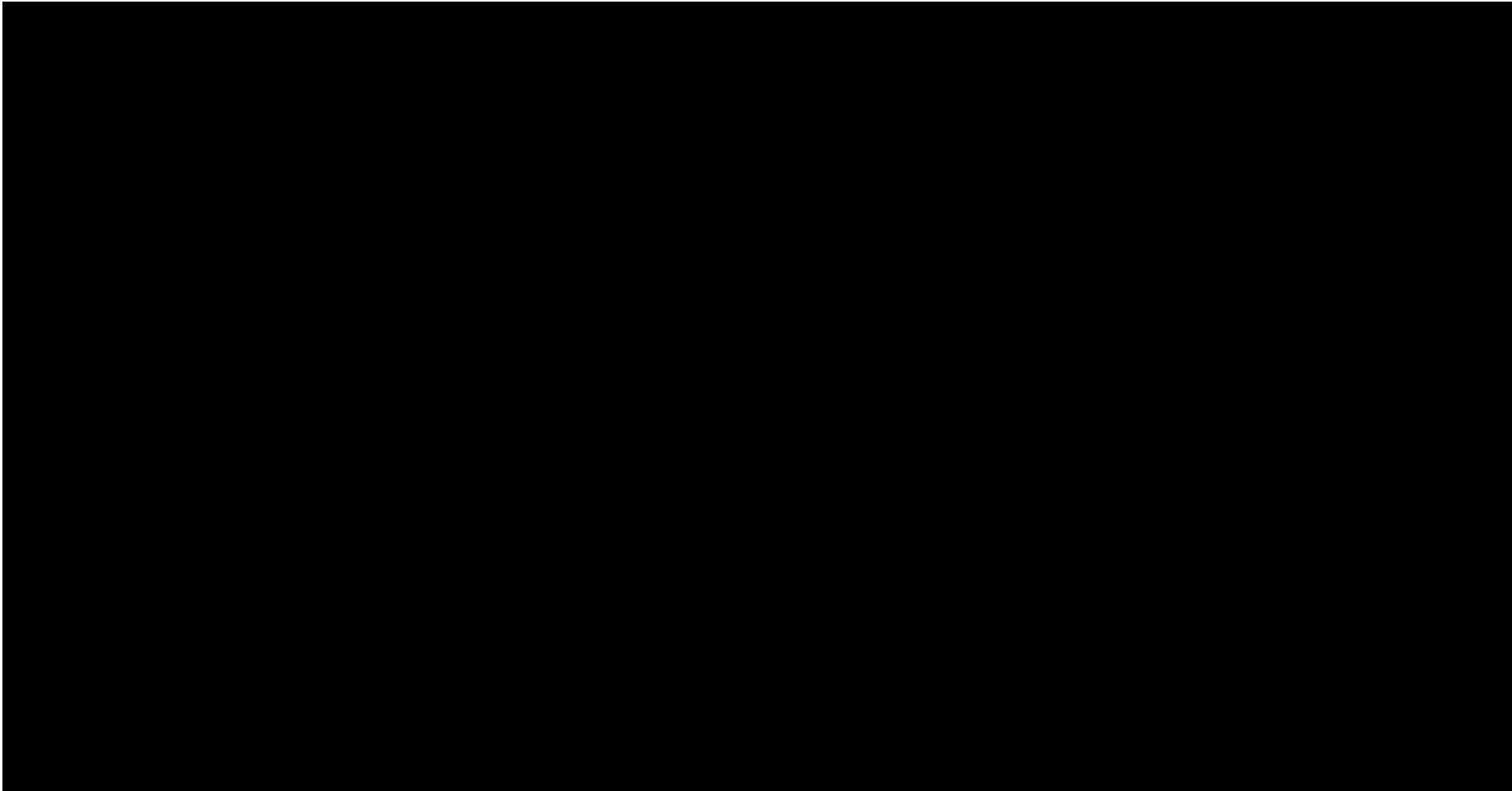


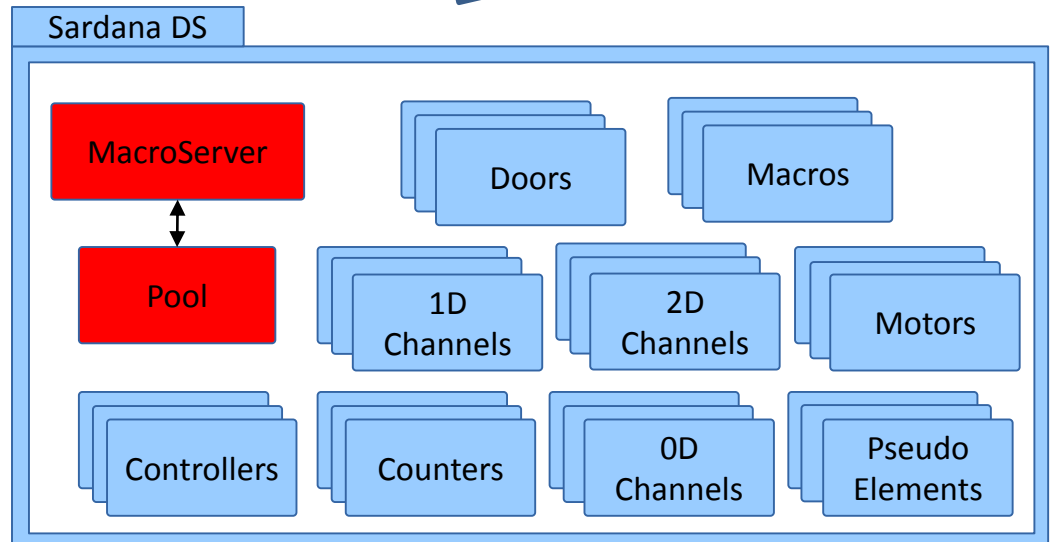
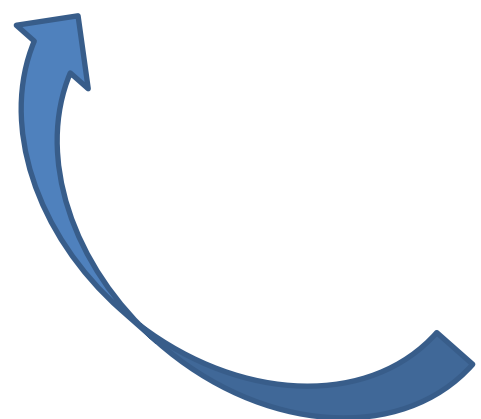
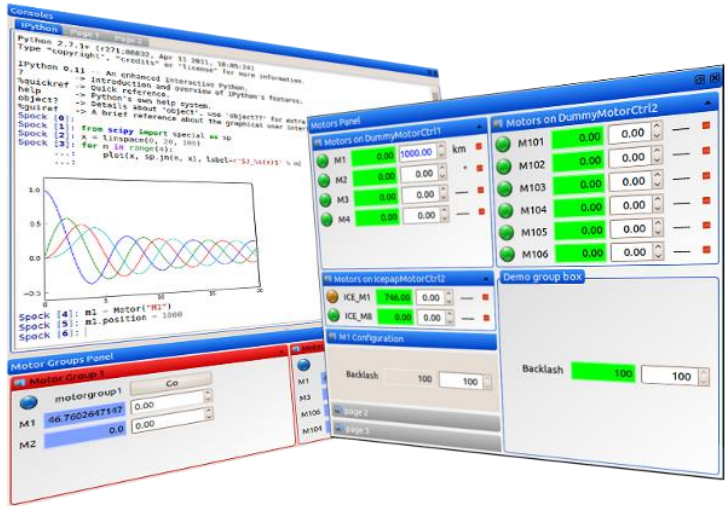
Install Sardana in 3 min



- Sardana is redistributed as the official Debian package (thanks to Frédéric Picca from the Soleil Synchrotron!)
`$> apt-get install python-sardana`
- Sardana is hosted on PyPI
`$> pip install taurus`
`$> pip install sardana`
- Sardana code is hosted on the Sourceforge platform
`$> git clone`
`git://git.code.sf.net/p/sardana/sardana.git`
`python setup.py install`

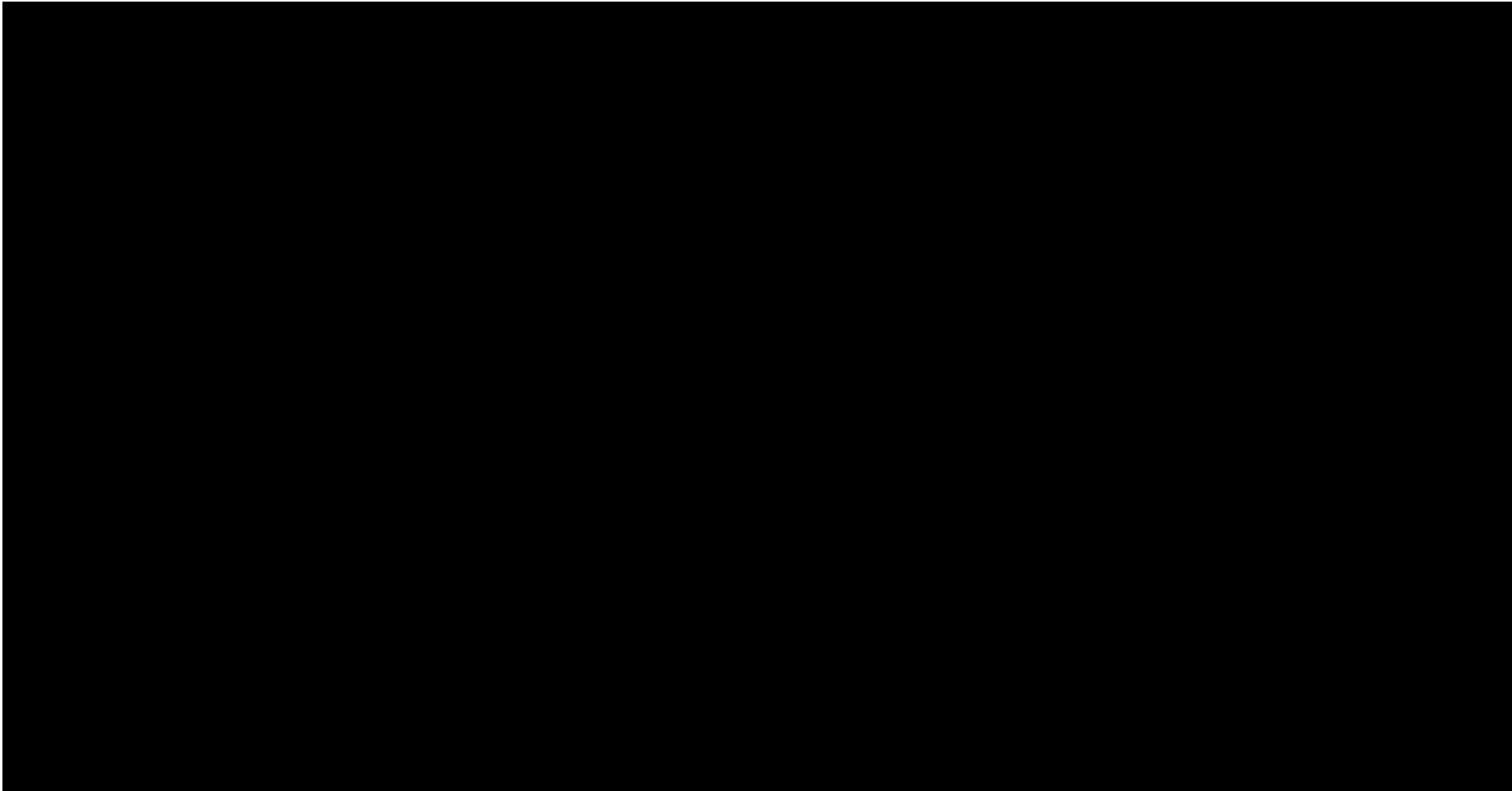
More details on http://www.sardana-controls.org/en/latest/users/getting_started/installing.html



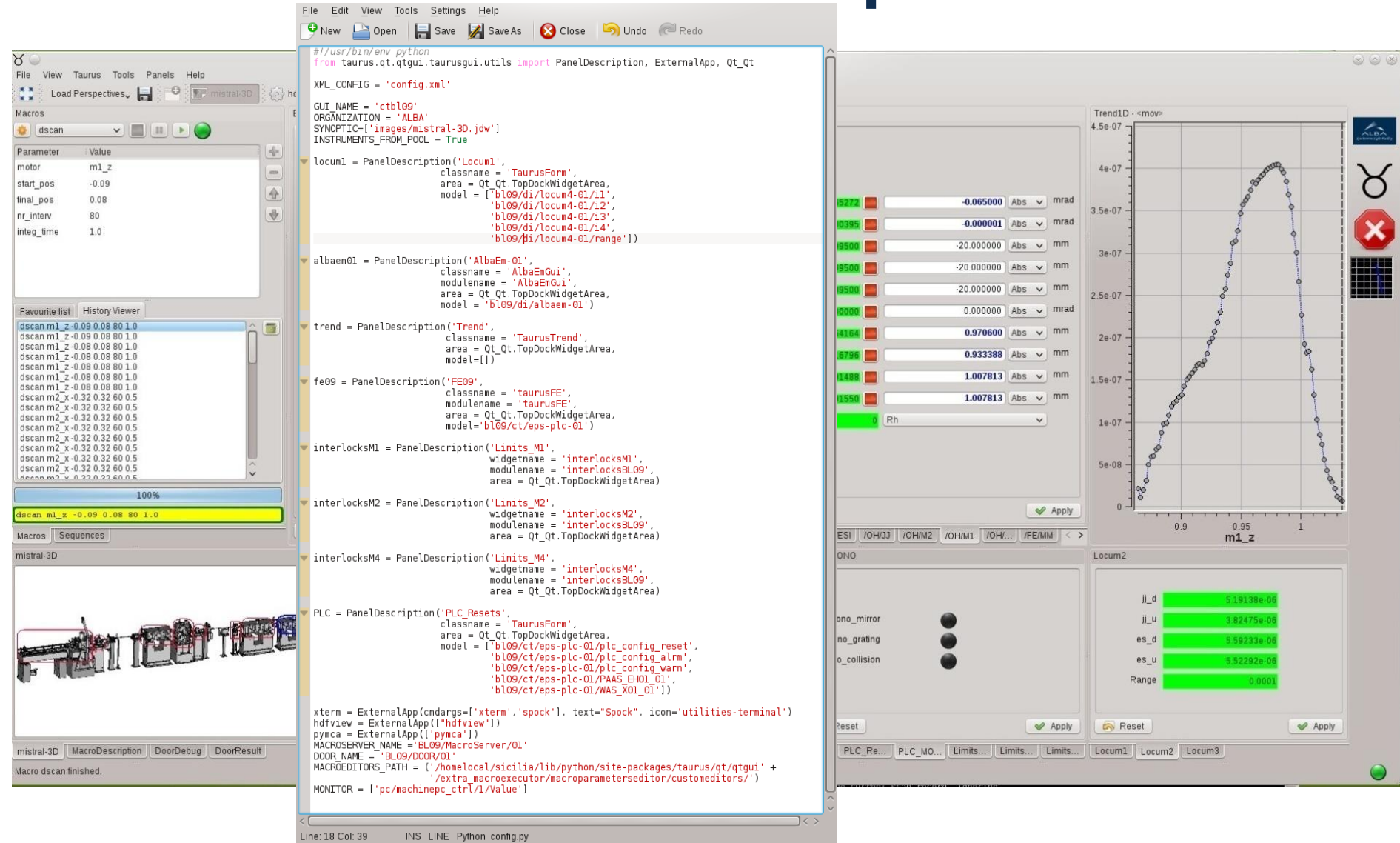




Taurus demo



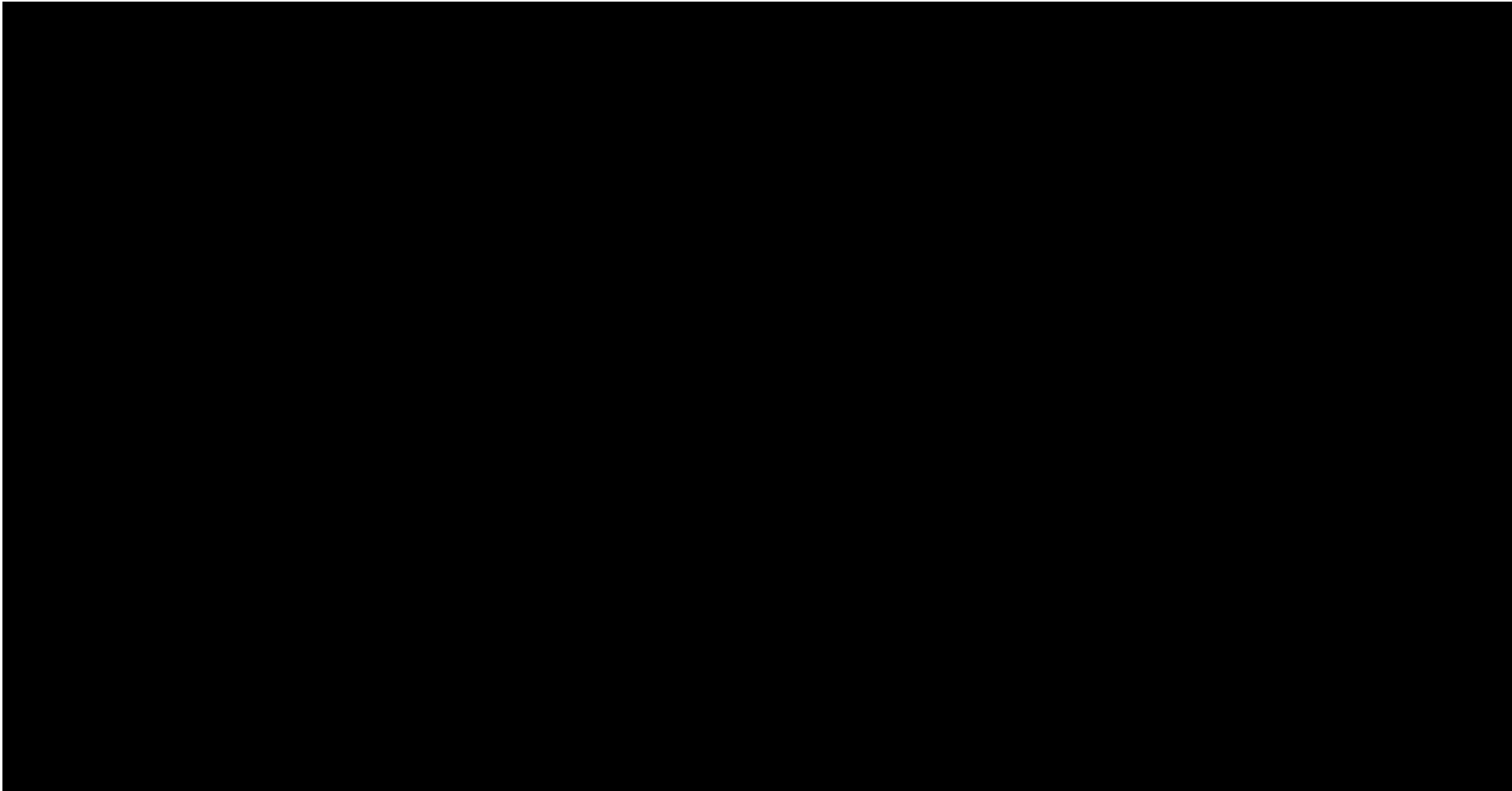
Taurus GUI example: CTBL09

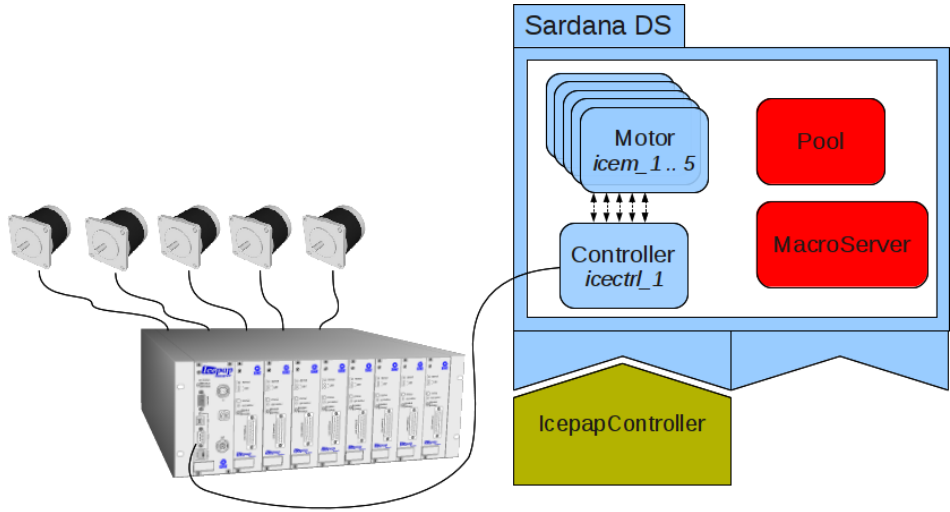


The screenshot displays the Taurus GUI interface for the CTBL09 experiment. It is divided into several main sections:

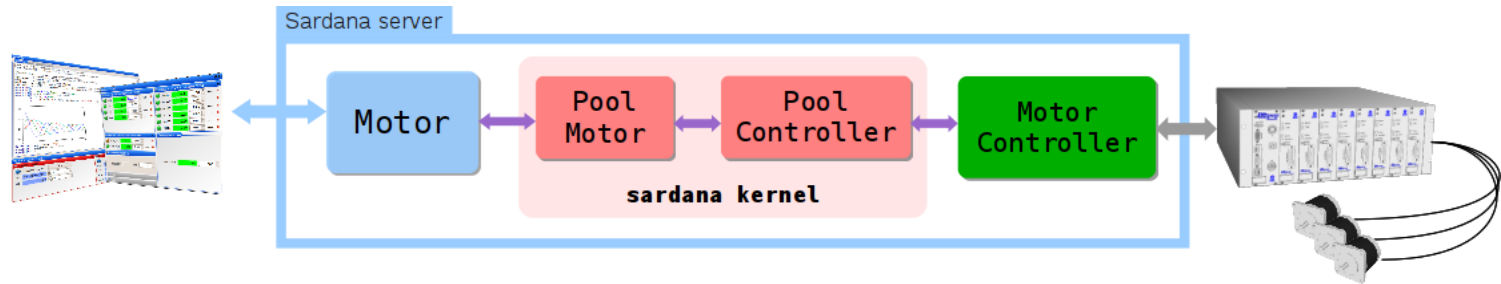
- Configuration Editor (Left):** Shows a Python script defining the GUI components. Key elements include:
 - `locum1`: A `PanelDescription` for the main control area, using the `TaurusForm` class.
 - `albaem01`: A `PanelDescription` for the AlbaEmGui module.
 - `trend`: A `PanelDescription` for the trend display.
 - `fe09`: A `PanelDescription` for the FE09 detector.
 - `interlocksM1`, `interlocksM2`, and `interlocksM4`: `PanelDescription` objects for different interlock systems.
 - `PLC`: A `PanelDescription` for PLC resets.
 - `xterm`, `hdfview`, and `pymca`: `ExternalApp` objects for external utilities.
- Parameter Table (Top Left):** A table with columns 'Parameter' and 'Value'.

| | |
|------------|-------|
| motor | m1_z |
| start_pos | -0.09 |
| final_pos | 0.08 |
| nr_interv | 80 |
| integ_time | 1.0 |
- Scan Parameters List (Middle Left):** A list of scan parameters, including `dscan m1_z` and `dscan m2_x` with their respective values and integration times.
- 3D Model (Bottom Left):** A 3D visualization of the CTBL09 detector components.
- Control Panel (Right):** A graphical interface for controlling the experiment. It features:
 - A **Trend1D** graph showing a peak at approximately `m1_z = 0.95`.
 - A list of parameters with sliders and units (e.g., `0.5722` mrad, `0.0395` mrad, `0.9500` mm).
 - Buttons for `Apply` and `Reset`.
 - A **Locum2** section showing status for `ono_mirror`, `no_grating`, and `o_collision`.



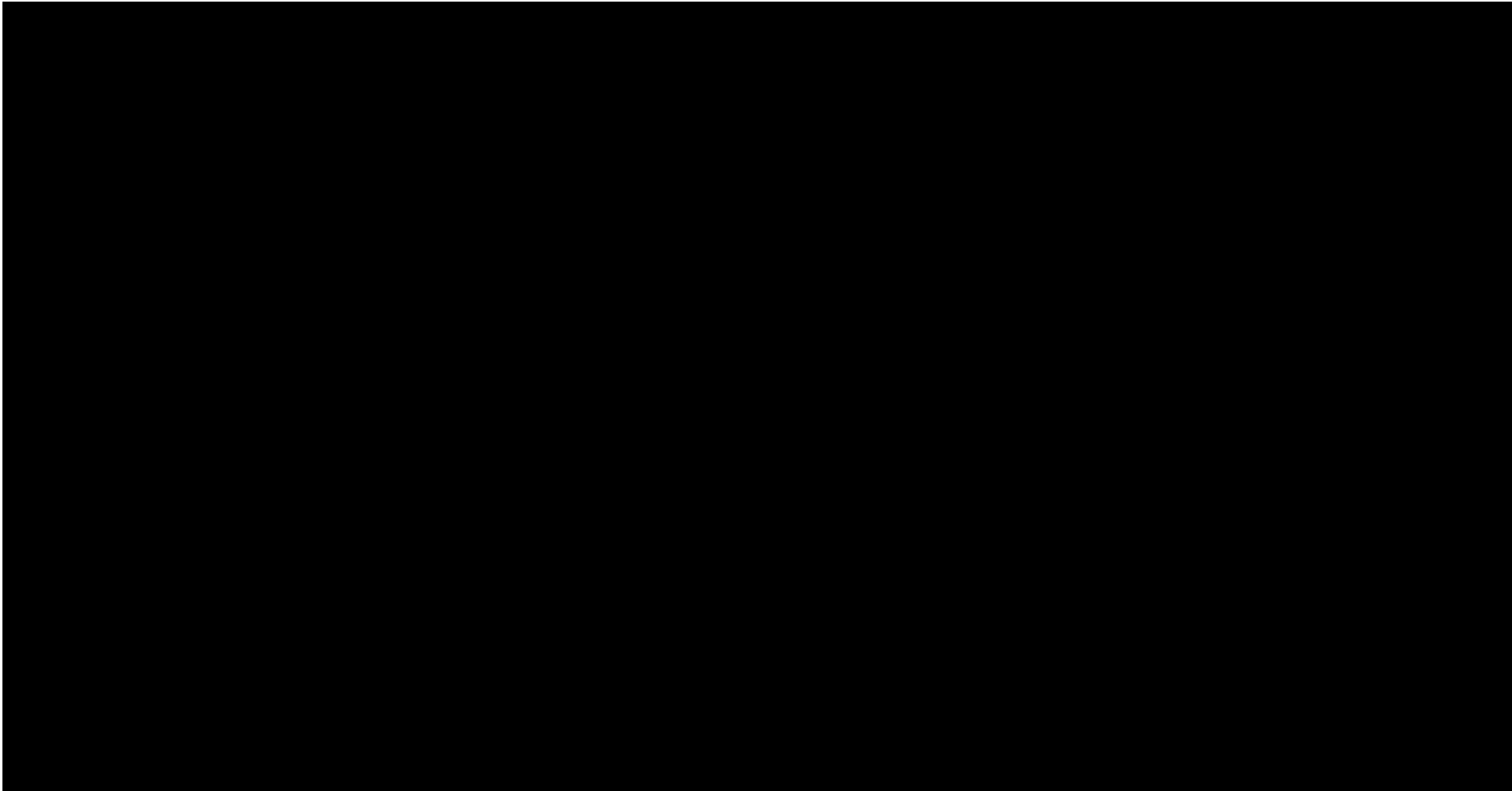


Motor Device
 States: ON, MOVING, ALARM,
 FAULT
 Commands: Stop, Abort
 Attributes: Position, Offset,
 Velocity, Acceleration...



http://www.sardana-controls.org/en/latest/devel/howto_controller

<http://sourceforge.net/p/sardana/wiki/Howto-CreateControllers>



Sardana Beta
 A python based control system
 Brought to you by: anmilan, opascual, tere29, tiagocoutinho, zreszela

Summary | Files | Reviews | Support | Tickets | Wiki | Mailing Lists | Git ▾

Browse Commits
 Fork
 Forks 2

Branches
 master
 sep0

Tree [519fe9] master /python/

RO HTTP Read Only access git clone git://gi

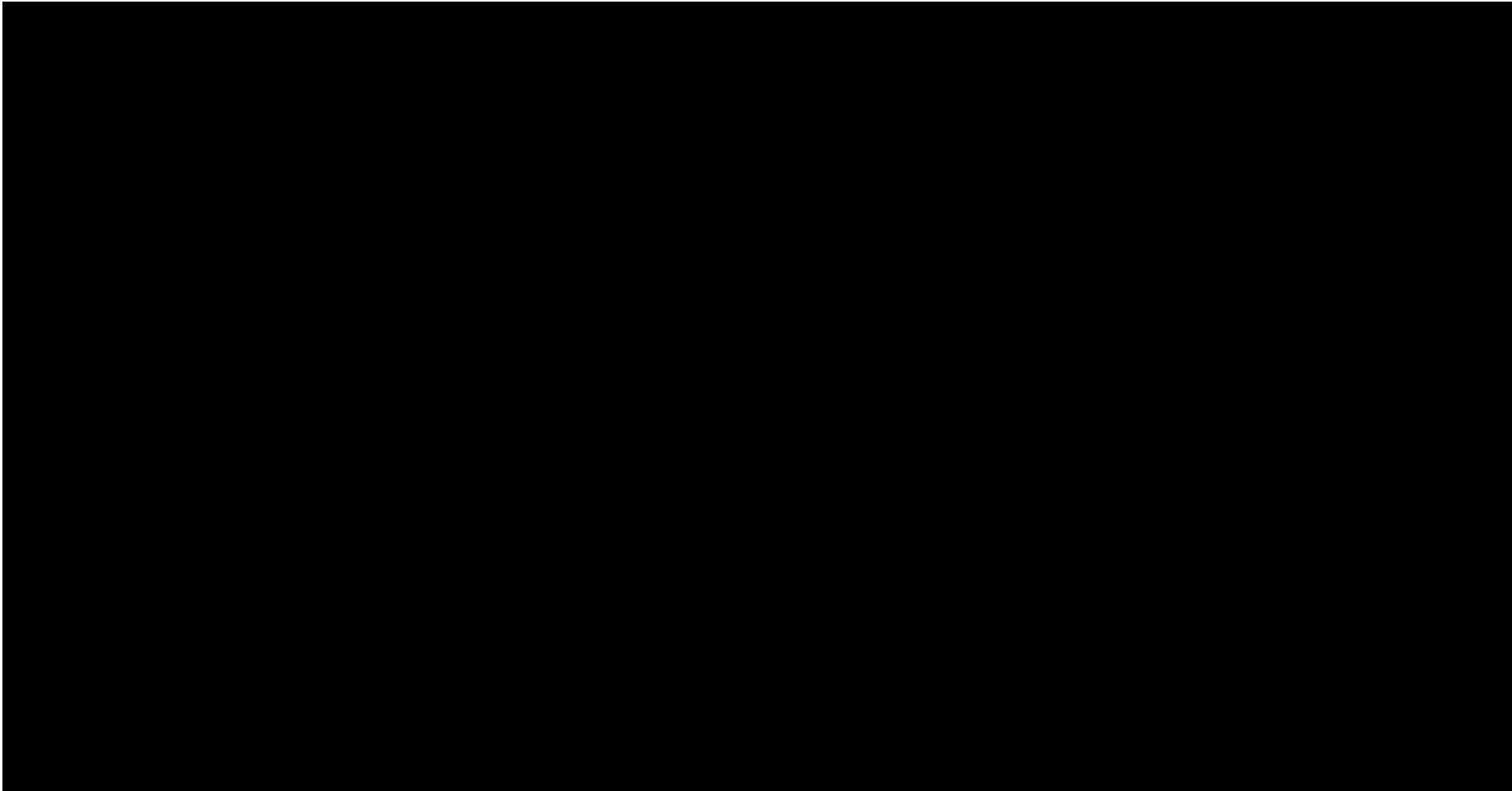
| File | DATE | Author |
|---------------|------------|--------------------|
| communication | 2012-07-30 | tiagocoutinho |
| countertimer | 2015-04-09 | Teresa |
| ioregister | 2014-12-10 | Teresa |
| motor | 2015-04-08 | Teresa |
| oned | 2014-12-22 | Teresa |
| pseudocounter | 2014-05-27 | Sergi Blanch-Torné |
| pseudomotor | 2015-04-10 | Jairo Moldes |
| twod | 2015-04-30 | Teresa |
| zerod | 2014-12-10 | Teresa |

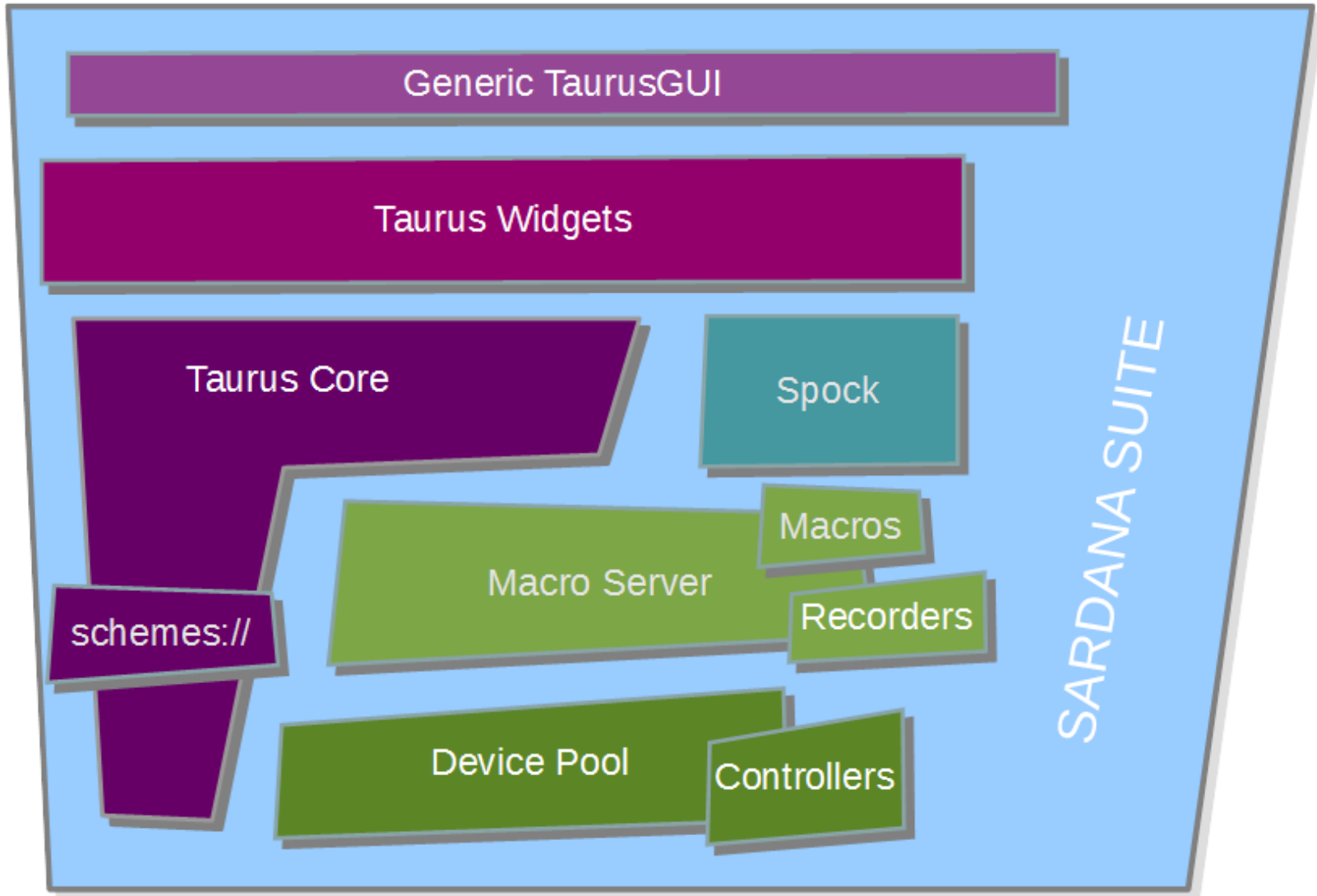
| File | Date | Author | Commit |
|---------------------|------------|----------------------------|---|
| ALBA_BL04_MSPD | 2015-04-10 | Marc Rosanes | [4beeee] Add macro for continuous madscan |
| ALBA_BL09 | 2014-08-25 | marc rosanes | [f100f0] Syntax error in last line of BL09_macros.py |
| ALBA_BL13_XALOC | 2014-08-25 | Teresa | [d8e918] Typo bug |
| ALBA_BL22_CLAESS | 2015-01-12 | Roberto J. Homs Puron | [38ae3d] Include qExafs macro |
| ALBA_BL24_CIRCE | 2012-11-07 | anmilan | [7ff2f4] Added macros for BL24 CIRCE. |
| ALBA_BL29 | 2014-09-28 | Sicilia Developer Controls | [5d10ed] - BL29keithley.py: keitley names once again cha... |
| ALBA_GENERAL | 2015-04-08 | Roberto J. Homs Puron | [45c754] Include a timeout in the photon shutter macros |
| ALBA_LIMA | 2014-12-19 | Daniel Roldán Ballesteros | [242707] Fix syntax |
| ALBA_LTP | 2012-07-30 | tiagocoutinho | [d89d17] restructure |
| ALBA_MACH | 2012-07-30 | tiagocoutinho | [d89d17] restructure |
| DESY_P01 | unknown | | |
| DESY_P03 | 2014-09-24 | Teresa | [14coe7] P03 macros |
| DESY_P06 | 2014-11-25 | Teresa | [d41f45] Adding maia macros |
| DESY_general | 5 days ago | Teresa | [ea054b] New version |
| albaemmacros.py | 2013-08-19 | zreszela | [0ee825] migration to python pool sardana (changing impo... |
| ioepap.py | 2014-02-24 | droidan | [230e73] Implement intelligent to reset loepap: |
| niposition.py | 2012-07-30 | tiagocoutinho | [d89d17] restructure |
| sardana-macros-code | 2013-08-25 | droidanb | [889142] |
| starter.py | 2012-10-28 | gjover | [188396] Starter macro added to start/stop device servers. |

```
git clone git://git.code.sf.net/p/sardana/controllers.git sardana-controllers.git
git clone git://git.code.sf.net/p/sardana/macros.git sardana-macros.git
```



Let's use it...





Synchronization, generic continuous scans, soft triggering, buffering

Improved log capabilities and log filters

Configuration editor

Continuous deployment. Native distribution packages. Improve installation of dependencies

Simple to use

Simple to debug

Simple to configure

Simple to install

THANK YOU

ALBA Studios 😊

