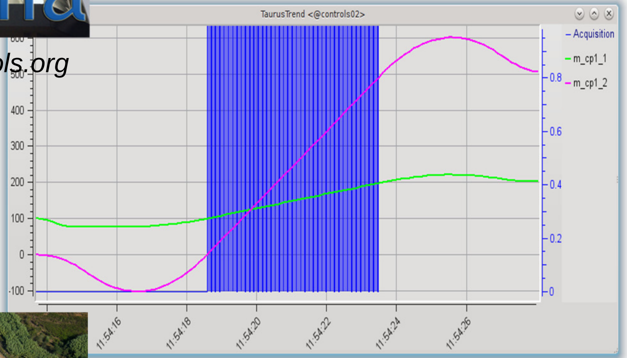
Sardana Based **Continuous Scans** at **ALBA** - Current Status

Z. Reszela, F. Becheri, G. Cuní, C. M. Falcón Torres, D. Fernández-Carreiras, R. Homs-Puron, J.Moldes, C. Pascual-Izarra, R. Pastor Ortiz, D. Roldan, M. Rosanes Siscart

ICALEPCS2017

www.sardana-controls.org



www.albasynchrotron.es

Contents

Introduction to Sardana

Generic Scan Framework Project

Continuous Scan Demo

Continuous Scans at ALBA

Contents

Introduction to Sardana

Generic Scan Framework Project

Continuous Scan Demo

Continuous Scans at ALBA

Sardana – Scientific SCADA Suite

- Sardana is an **open source**, **Python** based, **scientific SCADA suite** applicable in large spectrum of installations such as particle accelerators, experimental stations or small labs.
  It is highly modular and easily extendable with plugins.

- Sardana was initially an internal Alba project ...but after its successful use in other synchrotrons it became a **community driven project**

- Its architecture is based on the **client-server** model with **Tango** as the middleware

www.sardana-controls.org

www.taurus-scada.org

www.albasynchrotron.es

www.tango-controls.org

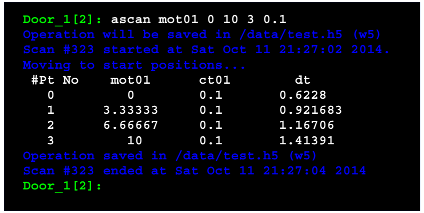
Taurus – Provides User Interfaces

www.taurus-scada.org

- Taurus is a framework for creating GUI and CLI to interact with control systems or other data sources [*]

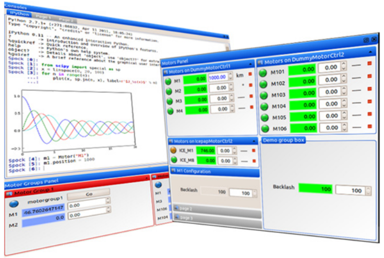
- Spock – IPython based Sardana CLI which syntax mimics SPEC commands, provides total control over the system: executes procedures, interacts with the elements, …

- Taurus based widgets and complete GUIs e.g. macro executor, motor, experiment configuration, scan plots, …



```
Door_1[2]: ascan mot01 0 10 3 0.1
Operation will be saved in /data/test.h5 (w5)
Scan #323 started at Sat Oct 11 21:27:02 2014.
Moving to start positions...
 #Pt No    mot01       ct01         dt
   0         0         0.1        0.6228
   1      3.33333      0.1        0.921683
   2      6.66667      0.1        1.16706
   3        10         0.1        1.41391
Operation saved in /data/test.h5 (w5)
Scan #323 ended at Sat Oct 11 21:27:04 2014
Door_1[2]:
```

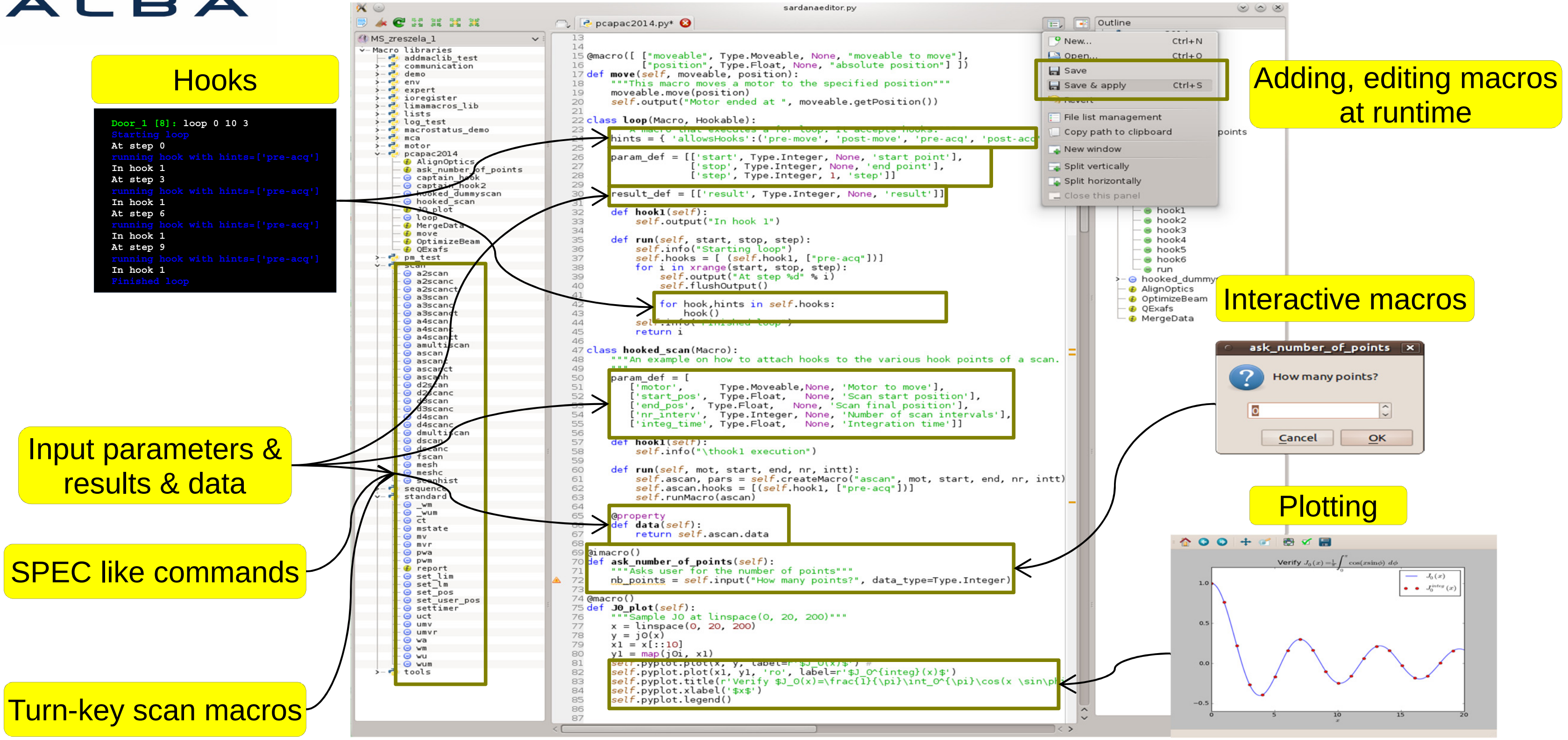
Scan execution using Spock.



Taurus based widgets interacting with Sardana

[*] TUBPL02: C. Pascual-Izarra et al. “Taurus For Big & Small: From Particle Accelerators to Desktop Labs

Macroserver - Runs User Procedures



Hooks

```
Door_1 [8]: loop 0 10 3
Starting loop
At step 0
running hook with hints=['pre-acq']
In hook 1
At step 3
running hook with hints=['pre-acq']
In hook 1
At step 6
running hook with hints=['pre-acq']
In hook 1
At step 9
running hook with hints=['pre-acq']
In hook 1
Finished loop
```

Input parameters & results & data

SPEC like commands

Turn-key scan macros

Adding, editing macros at runtime

Interactive macros

Plotting

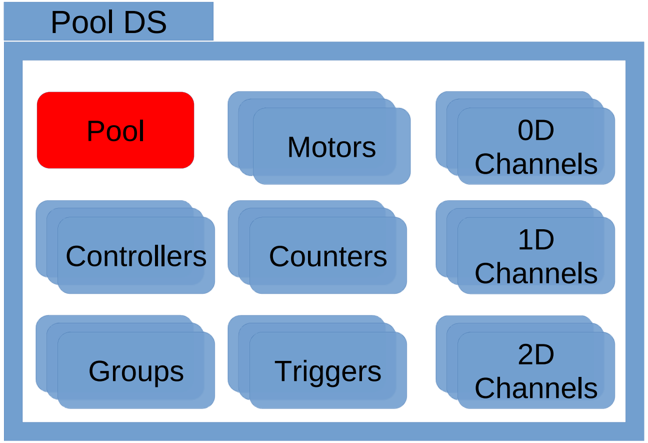
```
@macro([ ["moveable", Type.Moveable, None, "moveable to move"],
         ["position", Type.Float, None, "absolute position"] ])
def move(self, moveable, position):
    """This macro moves a motor to the specified position"""
    moveable.move(position)
    self.output("Motor ended at ", moveable.getPosition())

class loop(Macro, Hookable):
    """A macro that executes a for loop. It accepts hooks.
    hints = { 'allowsHooks':('pre-move', 'post-move', 'pre-acq', 'post-acq'

    param_def = [['start', Type.Integer, None, 'start point'],
                 ['stop', Type.Integer, None, 'end point'],
                 ['step', Type.Integer, 1, 'step']]

    result_def = [['result', Type.Integer, None, 'result']]

    def hook1(self):
        self.output("In hook 1")

    def run(self, start, stop, step):
        self.info("Starting loop")
        self.hooks = [ (self.hook1, ["pre-acq"])]
        for i in xrange(start, stop, step):
            self.output("At step %d" % i)
            self.flushOutput()

            for hook,hints in self.hooks:
                hook()
        self.info("Finished loop")
        return i

class hooked_scan(Macro):
    """An example on how to attach hooks to the various hook points of a scan.
    """
    param_def = [
       ['motor',      Type.Moveable,None, 'Motor to move'],
       ['start_pos',  Type.Float,   None, 'Scan start position'],
       ['end_pos',   Type.Float,   None, 'Scan final position'],
       ['nr_interv',  Type.Integer, None, 'Number of scan intervals'],
       ['integ_time', Type.Float,   None, 'Integration time']]

    def hook1(self):
        self.info("\thook1 execution")

    def run(self, mot, start, end, nr, intt):
        self.ascan, pars = self.createMacro("ascan", mot, start, end, nr, intt)
        self.ascan.hooks = [(self.hook1, ["pre-acq"])]
        self.runMacro(ascan)

    @property
    def data(self):
        return self.ascan.data

@imacro()
def ask_number_of_points(self):
    """Asks user for the number of points"""
    nb_points = self.input("How many points?", data_type=Type.Integer)

@macro()
def J0_plot(self):
    """Sample J0 at linspace(0, 20, 200)"""
    x = linspace(0, 20, 200)
    y = j0(x)
    x1 = x[::10]
    y1 = map(j0i, x1)
    self.pyplot.plot(x, y, label=r'$J_0(x)$') #
    self.pyplot.plot(x1, y1, 'ro', label=r'$J_0^{integ}(x)$')
    self.pyplot.title(r'Verify $J_0(x)=\frac{1}{\pi}\int_0^{\pi}\cos(x \sin\ph
    self.pyplot.xlabel('$x$')
    self.pyplot.legend()
```

Verify $J_0(x)=\frac{1}{\pi}\int_0^{\pi}\cos(x\sin\phi)\ d\phi$
$J_0(x)$
$J_0^{integ}(x)$
1.0
0.5
0.0
−0.5
0
5
10
15
20
x

Macro editor with exemplary macros demonstrating advanced macro programming features.

Device Pool - Unifies Hardware Access

- All the equipment is interfaced via Pool and its plug-in controller classes (Python)
- Generic elements' interfaces allow to build high level layers on top of them e.g. MeasurementGroup, virtual/pseudo elements, generic GUIs, …



Pool DS

Pool

Motors

0D Channels

Controllers

Counters

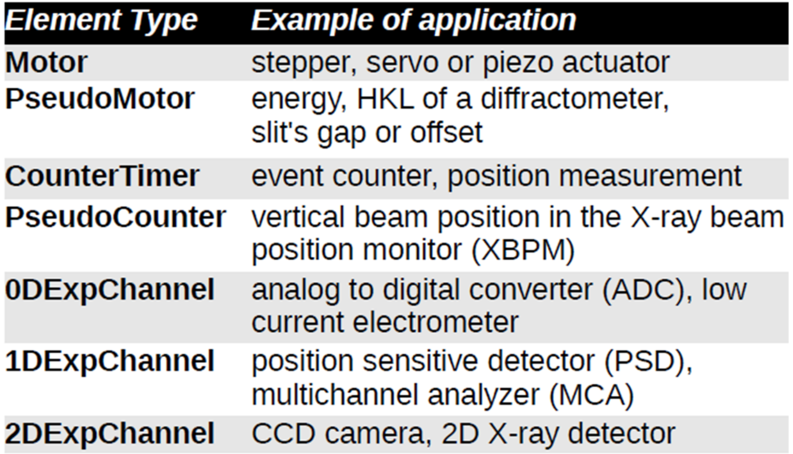
1D Channels

Groups

Triggers

2D Channels

Pool Device Server and its elements



Element Type	*Example of application*
Motor	stepper, servo or piezo actuator
PseudoMotor	energy, HKL of a diffractometer, slit's gap or offset
CounterTimer	event counter, position measurement
PseudoCounter	vertical beam position in the X-ray beam position monitor (XBPM)
0DExpChannel	analog to digital converter (ADC), low current electrometer
1DExpChannel	position sensitive detector (PSD), multichannel analyzer (MCA)
2DExpChannel	CCD camera, 2D X-ray detector

Pool element types and their examples

Contents

Introduction to Sardana

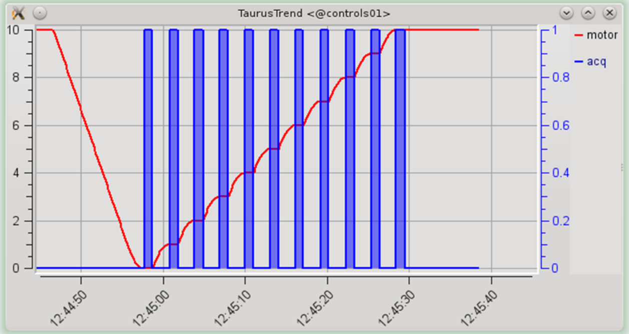
Generic Scan Framework Project

Continuous Scan Demo

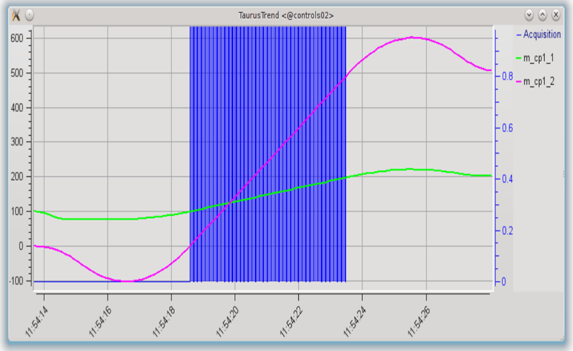
Continuous Scans at ALBA

Step Scan vs. Continuous Scan

- Continuous scan reduces the natural time overheads of a step scan:
  - the ones related to the motion e.g. acceleration, deceleration, instability
  - software state transitions: motion - acquisition and vice versa
- Continuous scans requires high precision synchronization:
  - Software synchronization - the acquisition state transition overheads can not be reduced.
  - Hardware synchronization - the fastest one and the most desirable



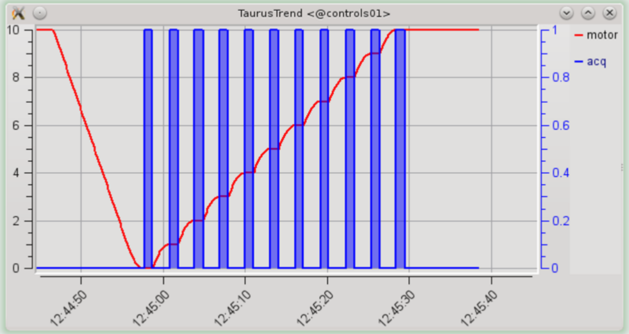
Motion & acquisition during the step scan.



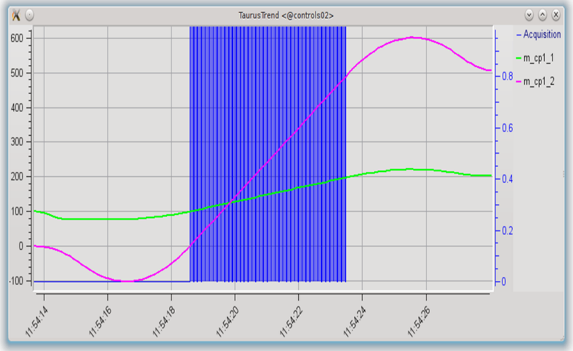
Motion & acquisition during the continuous scan

Step Scan vs. Continuous Scan

- In order to achieve the data acquisition while moveables are at constant speed the effective motion range needs to be extended on both sides:
  - exposing the scan on the risk of hitting moveables' limits
  - adding an extra time for handling these displacements
- The acquired data corresponds to time and position intervals in contrast to the step scan where the position is unchanged during the acquisition.



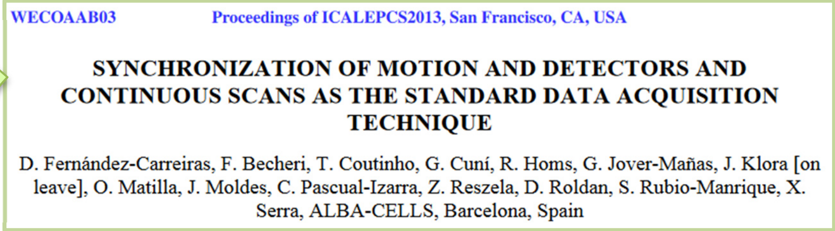
Motion & acquisition during the step scan.



Motion & acquisition during the continuous scan

Generic Continuous Scan in Sardana

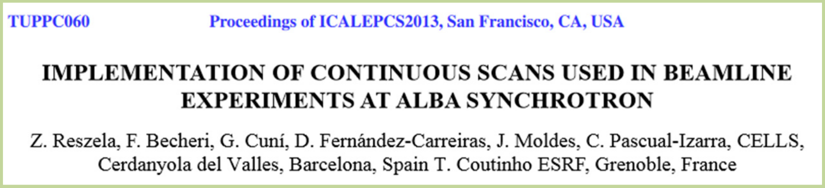
2013



WECOAAB03 Proceedings of ICALEPCS2013, San Francisco, CA, USA

SYNCHRONIZATION OF MOTION AND DETECTORS AND CONTINUOUS SCANS AS THE STANDARD DATA ACQUISITION TECHNIQUE

D. Fernández-Carreiras, F. Becheri, T. Coutinho, G. Cuní, R. Homs, G. Jover-Mañas, J. Klora [on leave], O. Matilla, J. Moldes, C. Pascual-Izarra, Z. Reszela, D. Roldan, S. Rubio-Manrique, X. Serra, ALBA-CELLS, Barcelona, Spain

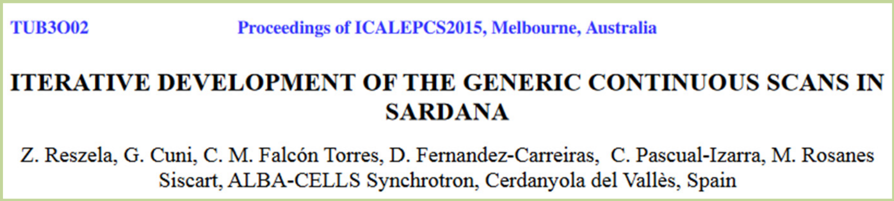


TUPPC060 Proceedings of ICALEPCS2013, San Francisco, CA, USA

IMPLEMENTATION OF CONTINUOUS SCANS USED IN BEAMLINE EXPERIMENTS AT ALBA SYNCHROTRON

Z. Reszela, F. Becheri, G. Cuní, D. Fernández-Carreiras, J. Moldes, C. Pascual-Izarra, CELLS, Cerdanyola del Valles, Barcelona, Spain T. Coutinho ESRF, Grenoble, France

2015



TUB3O02 Proceedings of ICALEPCS2015, Melbourne, Australia

ITERATIVE DEVELOPMENT OF THE GENERIC CONTINUOUS SCANS IN SARDANA

Z. Reszela, G. Cuni, C. M. Falcón Torres, D. Fernandez-Carreiras, C. Pascual-Izarra, M. Rosanes Siscart, ALBA-CELLS Synchrotron, Cerdanyola del Vallès, Spain

2017

Continuous Scans in Sardana version 2.3.2
Upgrade of the Alba Setups

Contents

Introduction to Sardana

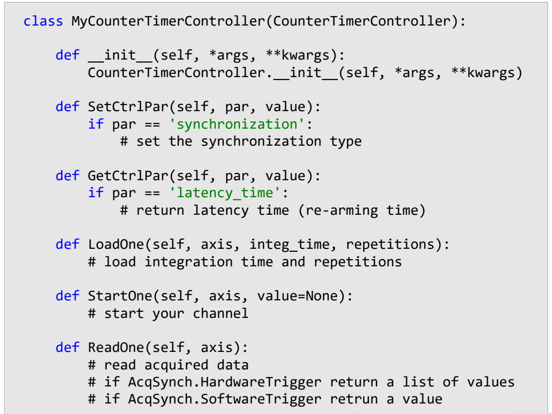
Generic Scan Framework Project

Continuous Scan Demo

Alba Continuous Scans

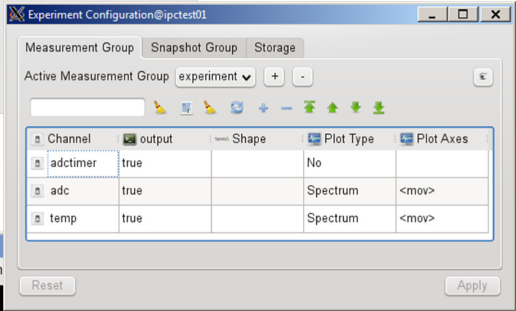
How to scan with Sardana?

- Integrate hardware:
  - reuse an already existing controller code from the third-party repository [*]
  - develop a new controller code
- Configure the experiment (GUI/CLI):
  - select the channels, configure saving, enable online plots, …
- Execute turn-key scanning macros (GUI/CLI):
  - step, continuous or time scans



```
class MyCounterTimerController(CounterTimerController):

    def __init__(self, *args, **kwargs):
        CounterTimerController.__init__(self, *args, **kwargs)

    def SetCtrlPar(self, par, value):
        if par == 'synchronization':
            # set the synchronization type

    def GetCtrlPar(self, par, value):
        if par == 'latency_time':
            # return latency time (re-arming time)

    def LoadOne(self, axis, integ_time, repetitions):
        # load integration time and repetitions

    def StartOne(self, axis, value=None):
        # start your channel

    def ReadOne(self, axis):
        # read acquired data
        # if AcqSynch.HardwareTrigger return a list of values
        # if AcqSynch.SoftwareTrigger retrun a value

    def StateOne(self, axis):
        # read state

    def AbortOne(self, axis):
        # abort your channel
```

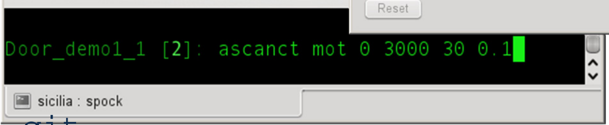
Skeleton of the counter/timer controller



Active Measurement Group experiment

Channel	output	Shape	Plot Type	Plot Axes
adctimer	true		No	
adc	true		Spectrum	<mov>
temp	true		Spectrum	<mov>

Experiment configuration widget



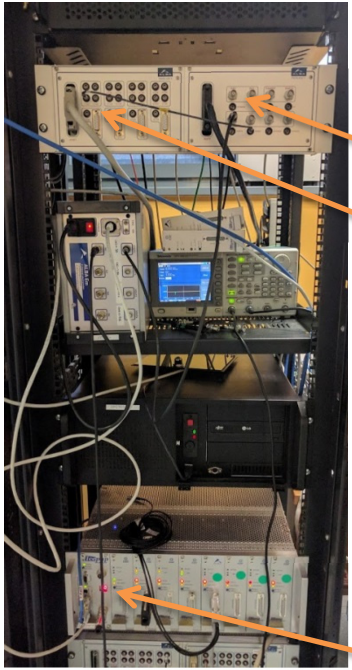
```
Door_demo1_1 [2]: ascanct mot 0 3000 30 0.1
```

Continuous scan macro interface: motor, start and end positions, nr. of intervals, integration time

[*] https://git.code.sf.net/p/sardana/controllers.git

Demo setup

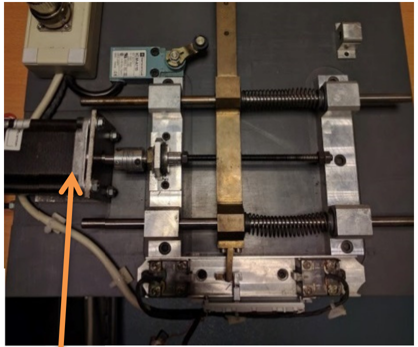
acknowledgements to: D. Roldan, R. Homs and J. Avila



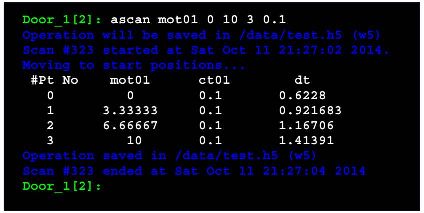
TANGO - temp attribute

ADC channel - adc

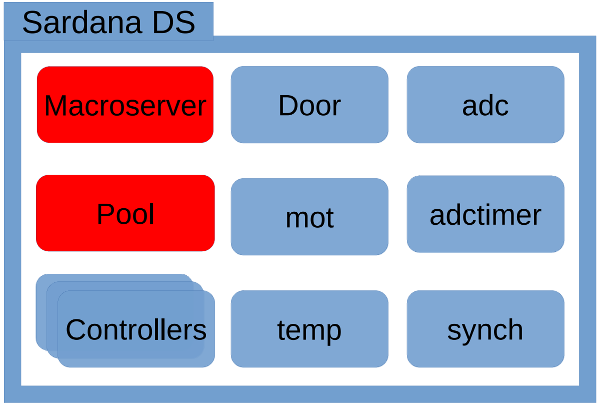
Counter/timer channel - synch



Stepper motor - mot



```
Door_1[2]: ascan mot01 0 10 3 0.1
Operation will be saved in /data/test.h5 (w5)
Scan #323 started at Sat Oct 11 21:27:02 2014.
Moving to start positions...
 #Pt No    mot01      ct01        dt
   0         0        0.1       0.6228
   1      3.33333     0.1       0.921683
   2      6.66667     0.1       1.16706
   3        10        0.1       1.41391
Operation saved in /data/test.h5 (w5)
Scan #323 ended at Sat Oct 11 21:27:04 2014
Door_1[2]:
```

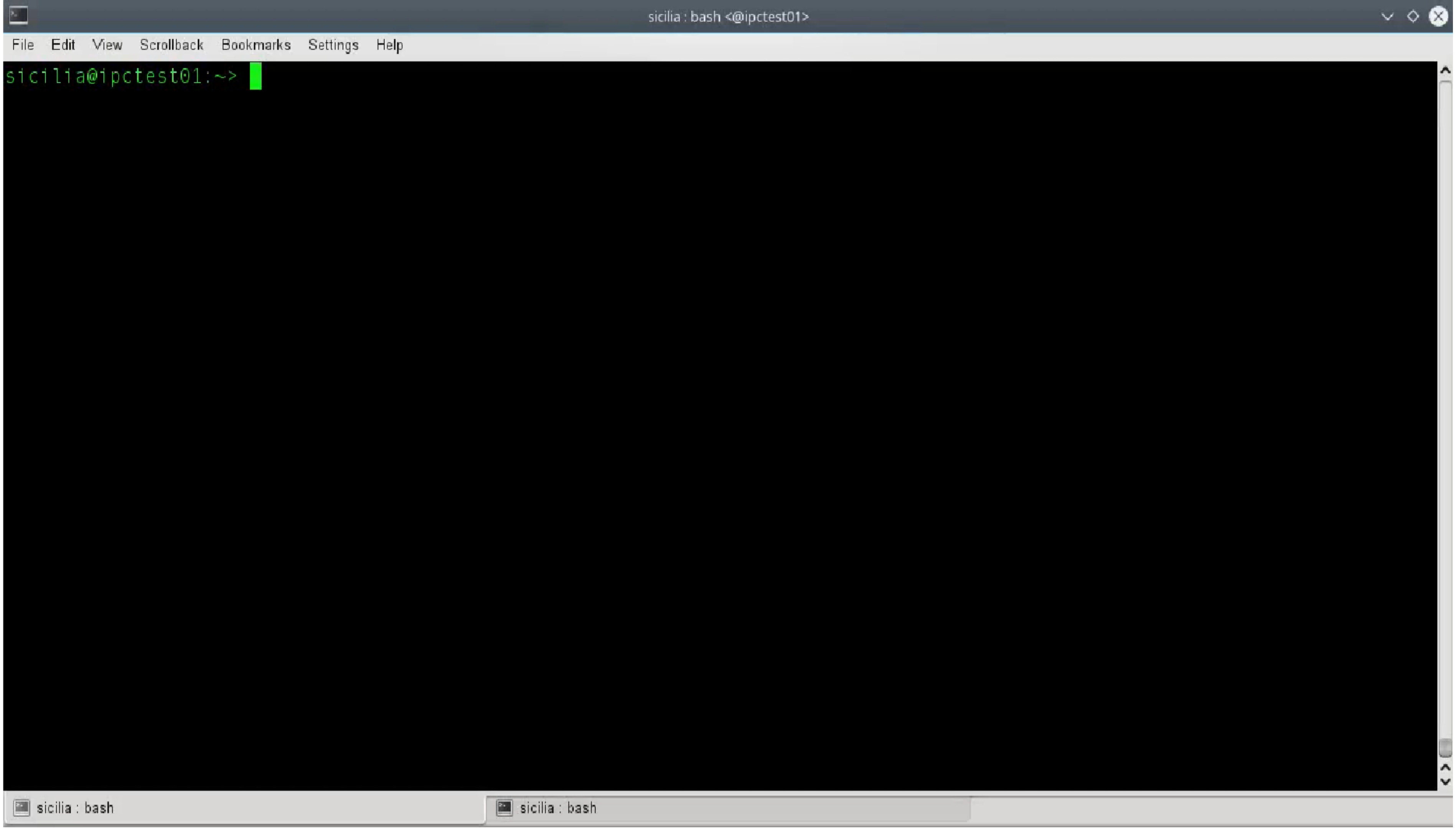


Sardana DS

Macroserver | Door | adc

Pool | mot | adctimer

Controllers | temp | synch



```
sicilia@ipctest01:~>
```

Contents

Introduction to Sardana

Generic Scan Framework Project

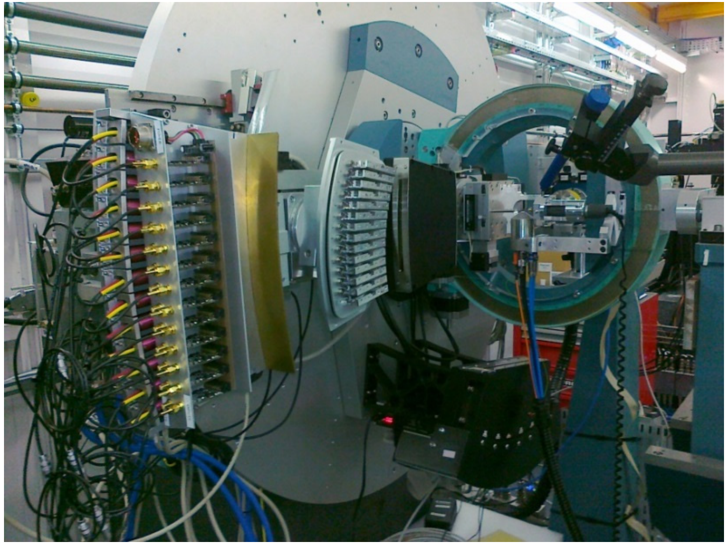
Continuous Scan Demo

Continuous Scans at ALBA

Powder Diffraction at MSPD beamline

by D. Roldan; acknowledgements to the beamline staff: F. Fauth, O. Vallcorba and C. Popescu

- Scan of the diffractometer circle (stepper driven by IcePAP)
- Acquisition with the MAD26 detector – 14 scintillator channels + electronics chain, photons counted with the NI6602 counters synchronized by hardware in the time domain
- Arbitrary number of software synchronized sample environment attributes



MSPD 3-circle diffractometer with MAD26 and Mythen detectors

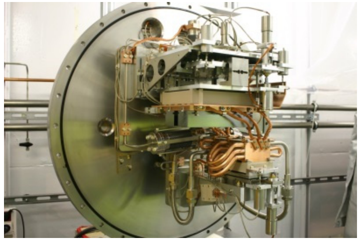
Experiment time reduction
~9h30min (step scan) -> ~**40min** (cont. scan)

OC range: 100 degrees; Integration Time: 0.025s; Intervals: 10000

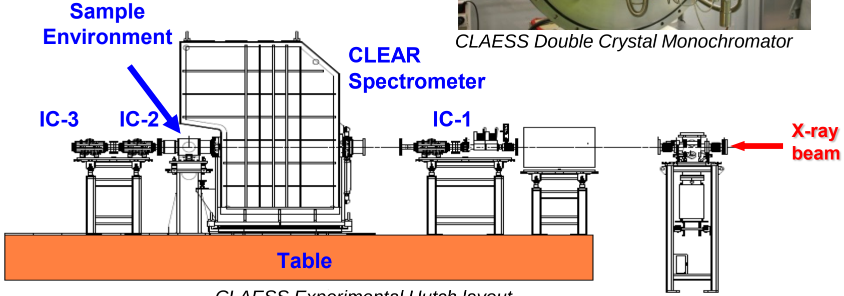
Absorption and Emission Spectroscopy at CLAESS

by R. Homs; acknowledgements to the beamline scientists: L. Simonelli, C. Marini, W. Olszewski, N. Ramanan, K. Klementiev [on leave]

- Scan of the beam energy using monochromator (servo DC driven by TurboPmac2)
- Acquisition with ionization chambers (AlbaEm + external ADC or VTF), position sensitive (Mythen) and fluorescence detectors (Xspress3 or AmptekPX5) – up to 40 channels
- Hardware synchronization done by NI6602 (time domain) and TurboPmac2 (position domain)



CLAESS Double Crystal Monochromator



CLAESS Experimental Hutch layout

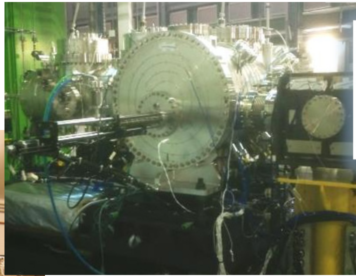
Experiment time reduction:
1h (step scan) -> **3min** (cont. scan)

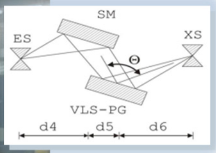
Energy range: 1keV (8969keV - 9969keV); Integration Time: 0.0291s; Intervals: 4000

Absorption Spectroscopy and Dichroism at BOREAS

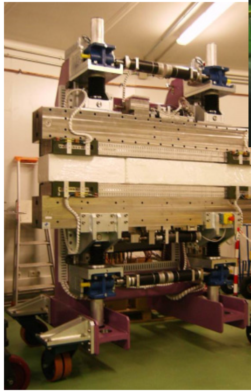
by J. Moldes; acknowledgements to the laboratory scientists: M. Valvidares, P. Gargiani, J. Herrero, H. Babu

- Scan of the beam energy using monochromator (PGM) (stepper driven by IcePAP)
- ID energy must closely follow the monochromator energy change – no synchronization with the PGM
- Photoelectric effect electron yield is measured by AlbaEm + external ADC;
- Hardware synchronization done by NI6602 (time domain)

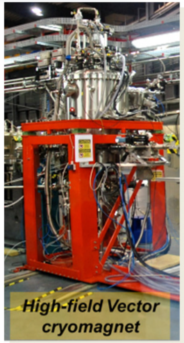




Plain Graiting Monochromator of BOREAS



Insertion Device of BOREAS



High-field Vector cryomagnet

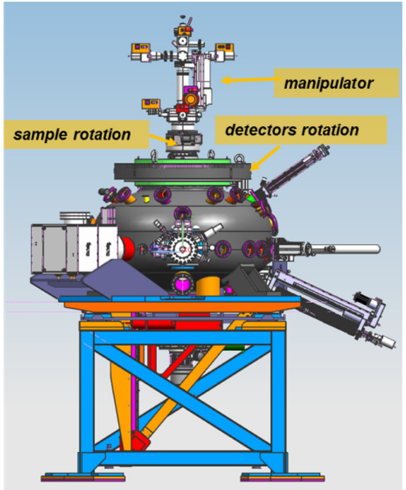
Experiment time reduction:
1h25min (step scan) -> **2min** (cont. scan)

Energy range: 65 eV (755eV – 820eV) Integration time: 124ms Intervals: 4000

Resonant Scattering and Reflectivity at BOREAS

by J. Moldes; acknowledgements to the laboratory scientists: M. Valvidares, P. Gargiani, J. Herrero, H. Babu

- Simultaneous scan of 3 axes: sample (theta), detector (2theta) and HTS Magnet – steppers driven by IcePAP
- Reflectivity detection using photodiode (Keithley6517 + external ADC)
- Currently sychronized by software in the position domain



MARES end station reflectometer

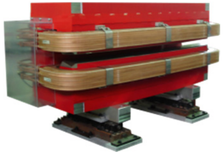
Experiment time reduction:
17min (step scan) -> **7min** (cont. scan)

Specular range: 47°; Integration time: 0.2 s ; Intervals" 470

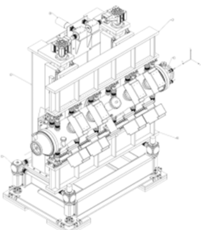
Magnetic Field Map Measurement

by R. Pastor, F. Becheri, R. Homs; acknowledgements to the laboratory scientists: V. Massana, J. Campmany, J. Marcos

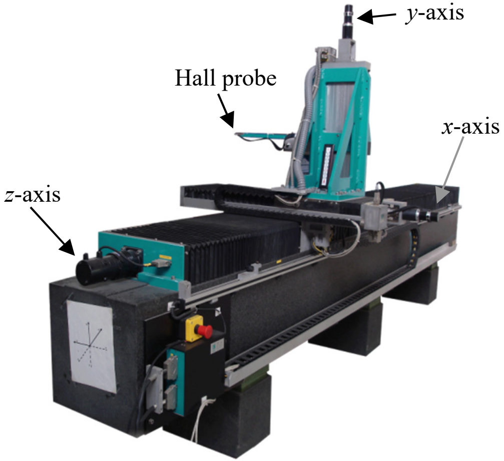
- Scan of the Z-axis (servo DC driven by PowerPmac) of the 3D robotic arm
- Acquisition with three Hall probe sensors. Their output voltage measured with Keithleys 2001 is used to determine the induction magnetic field.
- Hardware synchronization done by PowerPmac in the position domain.
- Software synchronized environmental temperatures.



Bending Magnet



Insertion Device Design



Magnetic Field Map workbench

Experiment time reduction:
~7h30min (step scan) -> **~4min** (cont. scan)

Z range: 2.7 m; Integration Time: 0.06s; Intervals: 2700

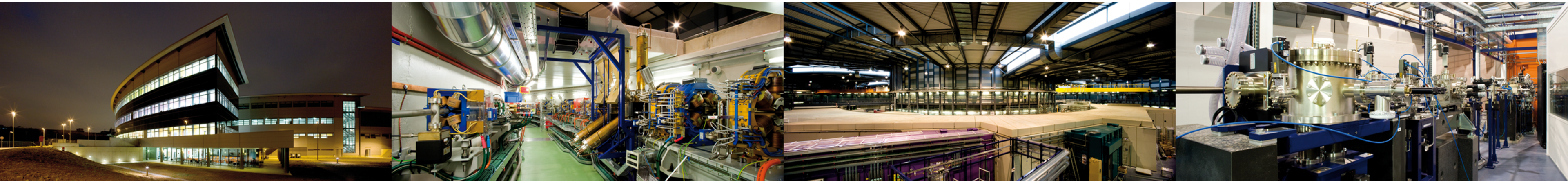
Conclusions

- Sardana provides user friendly and transparent continuous scans which allows to:
  - Scan any moveable(s) (also virtual/pseudo)
  - Acquire 0D and 1D experimental channels
  - Synchronize by software and hardware
- Basic requirements of the Alba beamlines are currently fulfilled
- Experiment time reduction by factor of ~20 (hardware synchronization)
- We plan to deploy it for other scanning setups at Alba
- DESY, MAXIV and SOLARIS are planning to evaluate it for their needs
- Further enhancements of the framework will continue:
  - Support to the 2D detectors
  - Generic interface to the nonlinear trajectories

Acknowledgements

- ALBA beamline scientists: François Fauth, Manuel Valvidares, Laura Simonelli, Valenti Massana, Oriol Vallcorba, Pierluigi Gargiani, Carlo Marini, Javier Herrero, Josep Campmany, Catalin Popescu, Jordi Marcos, Wojciech Olszewski, Hari Babu and Nitya Ramanan
- ALBA Electronics Section: Jose Avila, Xavier Serra, Roberto Petrocelli and Julio Lidon (on leave)
- Teresa Nuñez from DESY, Tiago Coutinho and Alejandro Homs from ESRF and Nicolas Leclercq from Soleil

ALBA



ICALEPCS2017

Thank you for your attention!