Hardware Triggered Scanning: Malcolm

Philip Taylor, Emma Arandjelovic

Observatory Sciences Limited

Software Stack: Reminder

Data Analysis WorkbeNch

- Analysis and visualization

Generic Data Acquisition

- Experiment setup and supervision

Malcolm

- Scan configuration

Experimental Physics & Industrial Control System

- Low level control of hardware

What is Malcolm?

- Generic and extensible framework for scanning
- Middle layer between GDA and control system
- Implemented in python™
- Creates a s/w map of the h/w layer
- https://pymalcolm.readthedocs.io
- Web GUI called MalcolmJS
- https://malcolmjs.readthedocs.io

Malcolm Process

Each Malcolm instance is a Python process managed by procServ

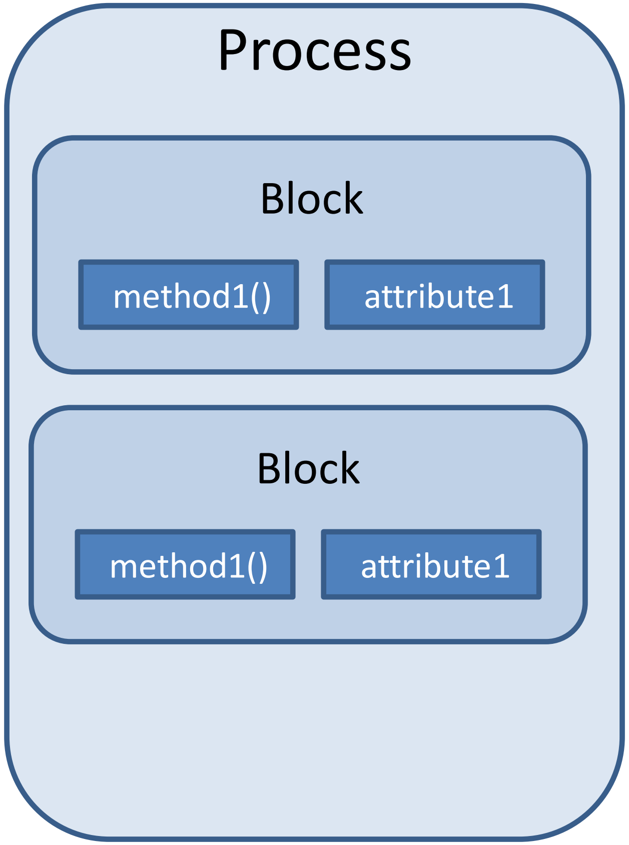
On beamlines this runs on ixx-control

```
[p47user@localhost ]$ ioc-list
BL47P-EA-IOC-01: pid = 2628, telnet port = 7002
BL47P-ML-MALC-01: pid = 2663, telnet port = 7003
```

To reboot:

```
ioc-connect BL4xP-ML-MALC-01
Ctrl-X
```

Malcolm Concepts



- A block is a user-centred view

  Examples:
  - Motion controller
  - PandA
  - Detector
  - ‘Hello World’ program

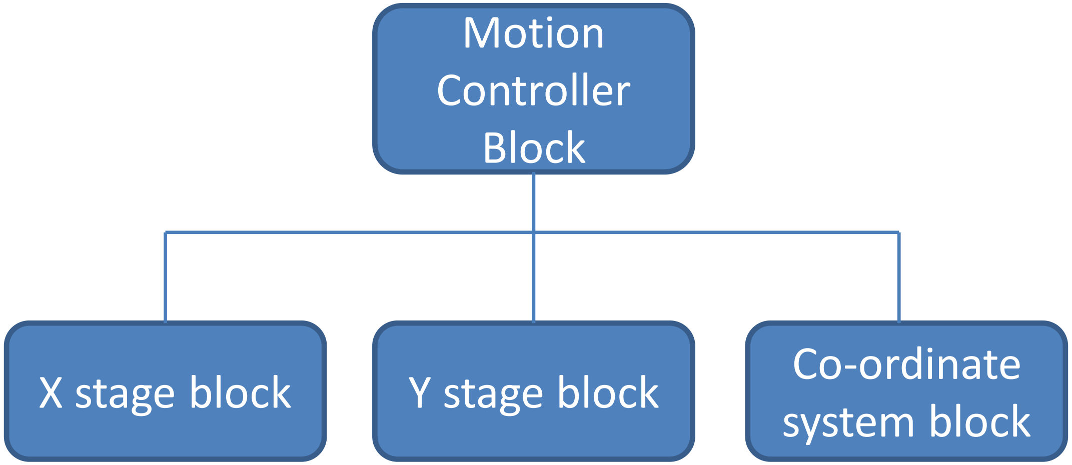
- Blocks contain:
  - Methods (actions)
  - Attributes (data)

- A process hosts multiple blocks

- Parent blocks contain child blocks connected together in a design

Malcolm Concepts

- Each block has a unqiue name called an MRI (Malcolm Resource Identifier) e.g. BL47P-ML-SCAN-01
- A device block is a higher level block for synchronizing a number of child blocks

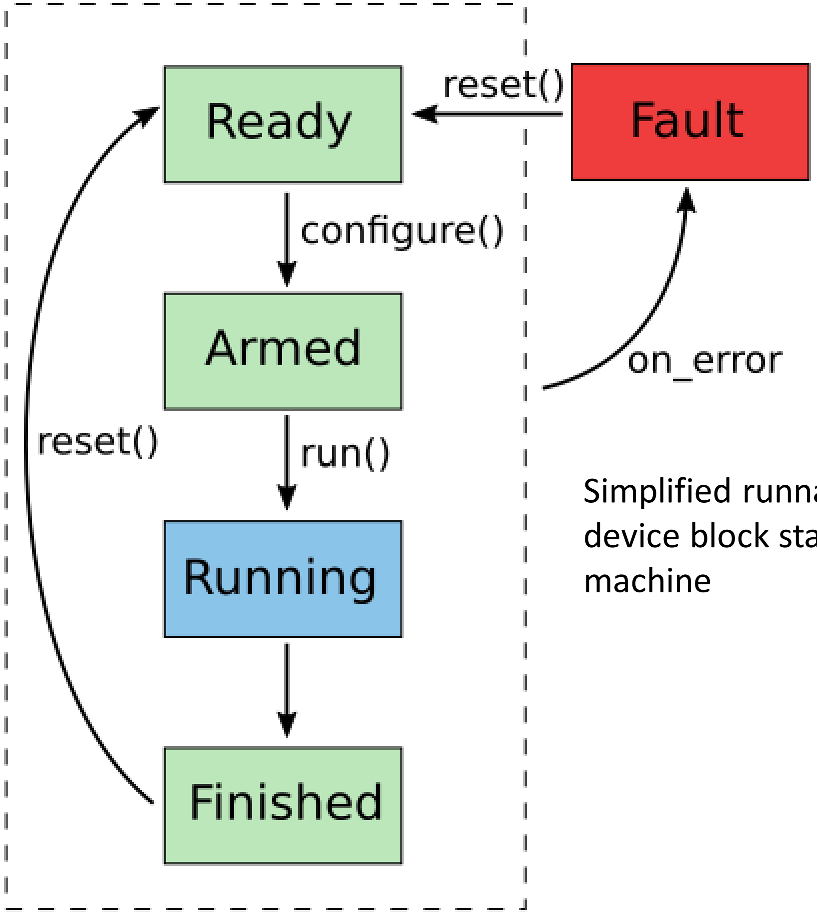


Malcolm Concepts

- A *runnable* device block adds state machine functionality with two main methods:

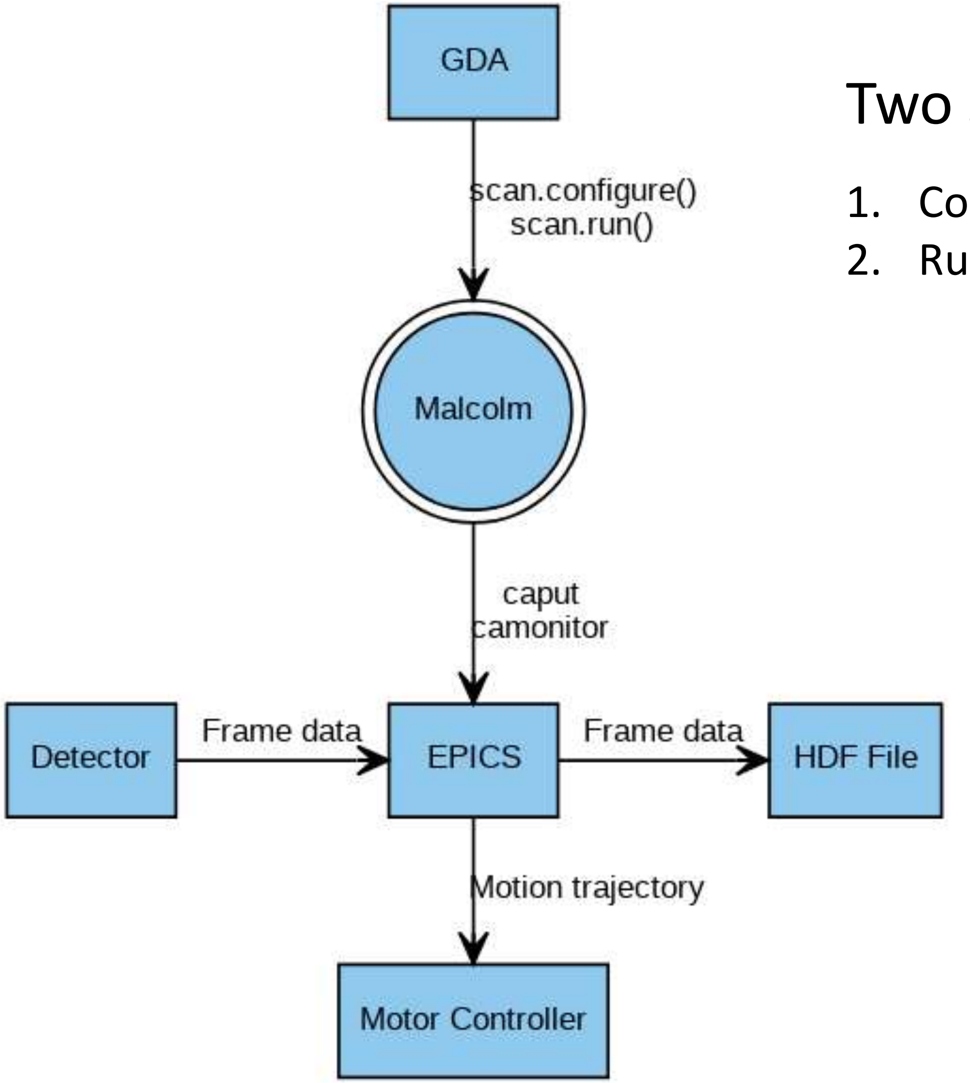
1. **configure(params)**
   – setup all child blocks

2. **run()**
   – start all children and supervise their progress



Simplified runnable device block state machine

Control Flow



Two step process:

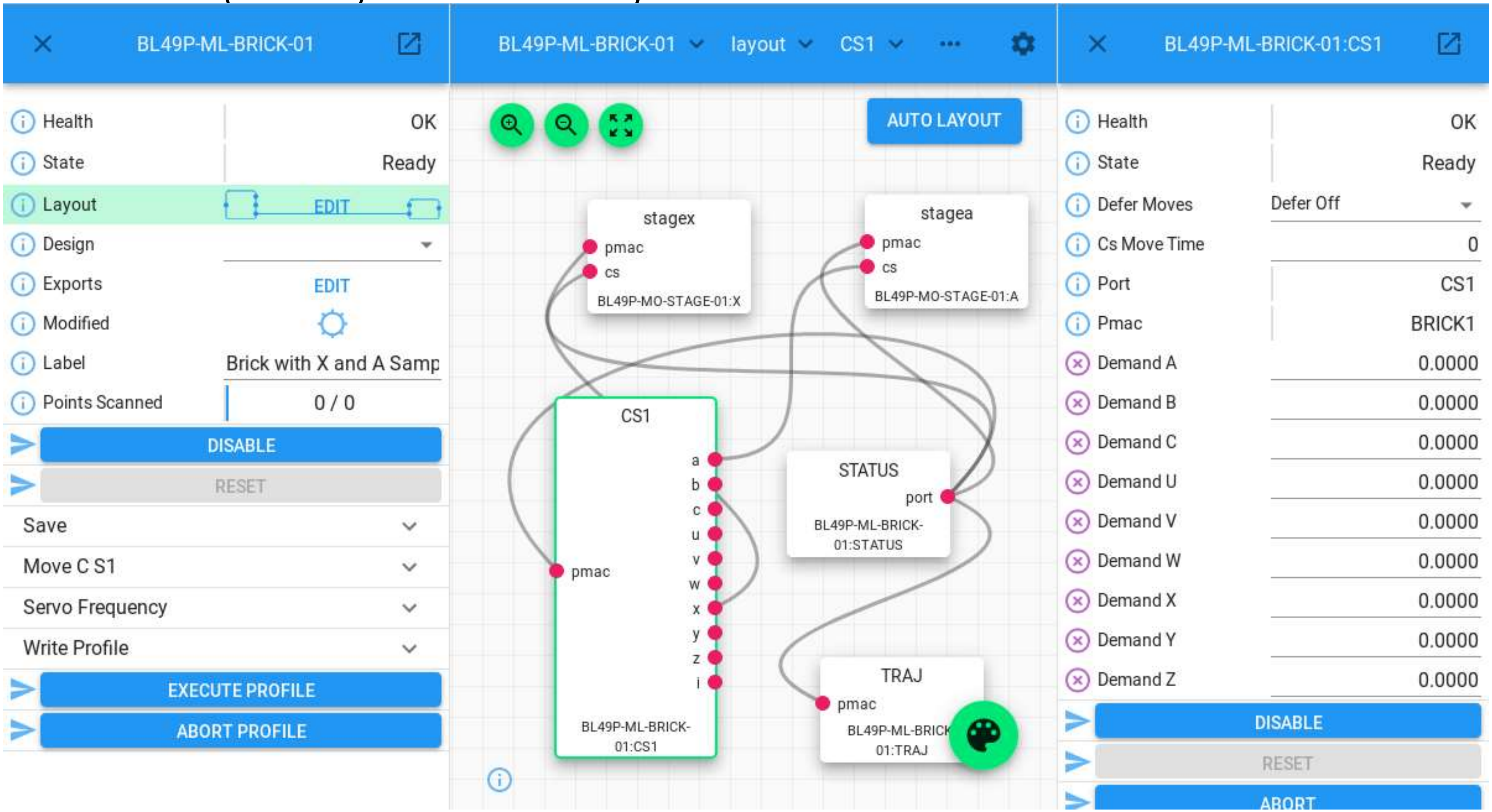
1. Configure the scan
2. Run the scan

Web GUI: Overview

Left: parent block (in focus)

Central panel: design layout / attribute view

Right: inspector



BL49P-ML-BRICK-01

Health	OK
State	Ready
Layout	EDIT
Design	
Exports	EDIT
Modified	
Label	Brick with X and A Samp
Points Scanned	0 / 0

DISABLE

RESET

Save

Move C S1

Servo Frequency

Write Profile

EXECUTE PROFILE

ABORT PROFILE

BL49P-ML-BRICK-01 layout CS1

AUTO LAYOUT

BL49P-ML-BRICK-01:CS1

Health	OK
State	Ready
Defer Moves	Defer Off
Cs Move Time	0
Port	CS1
Pmac	BRICK1
Demand A	0.0000
Demand B	0.0000
Demand C	0.0000
Demand U	0.0000
Demand V	0.0000
Demand W	0.0000
Demand X	0.0000
Demand Y	0.0000
Demand Z	0.0000

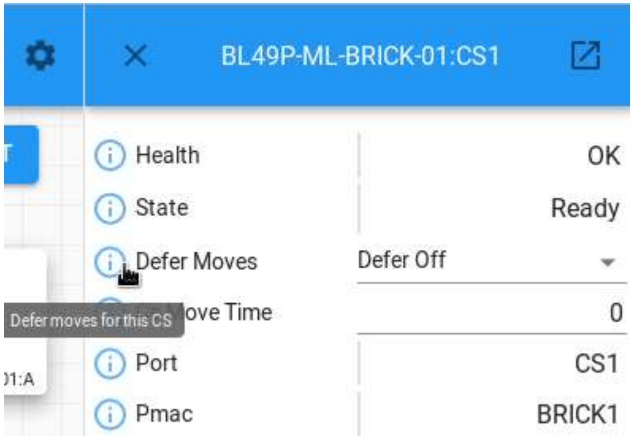
DISABLE

RESET

ABORT

Web GUI: Block View

- Attributes displayed depend on the type of block
- Tool-tip text provide descriptions
- Any problems connecting to the h/w are displayed in the tool-tip

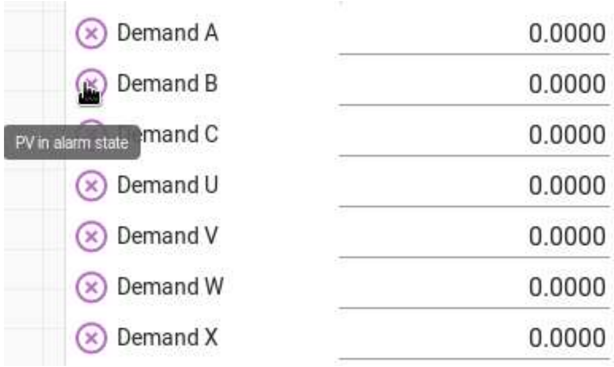


BL49P-ML-BRICK-01:CS1

Health	OK
State	Ready
Defer Moves	Defer Off
ove Time	0
Port	CS1
Pmac	BRICK1

Defer moves for this CS

01:A

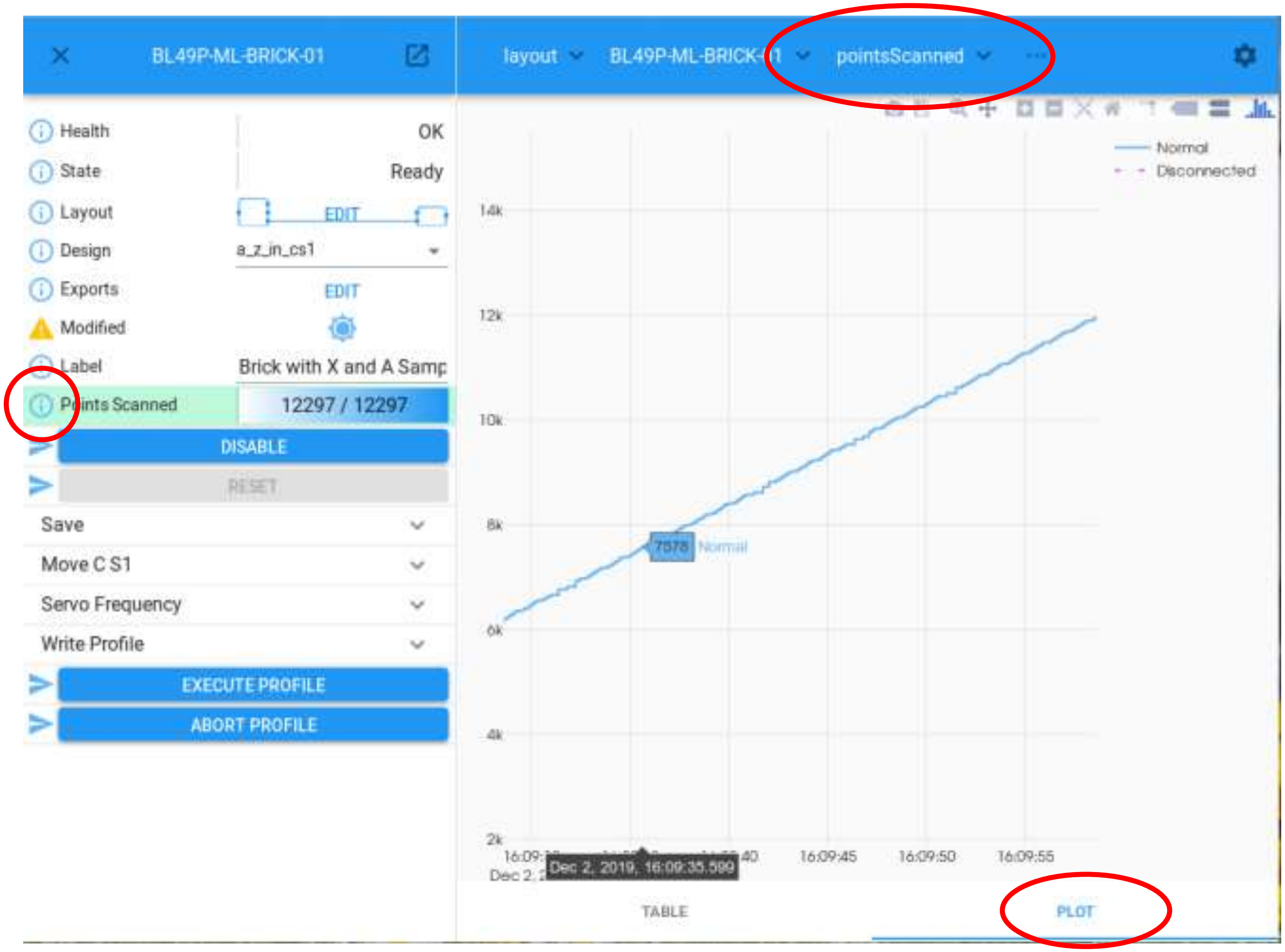


Demand A	0.0000
Demand B	0.0000
mand C	0.0000
Demand U	0.0000
Demand V	0.0000
Demand W	0.0000
Demand X	0.0000

PV in alarm state

Web GUI: Attribute View

Displays table or plot of historical data values

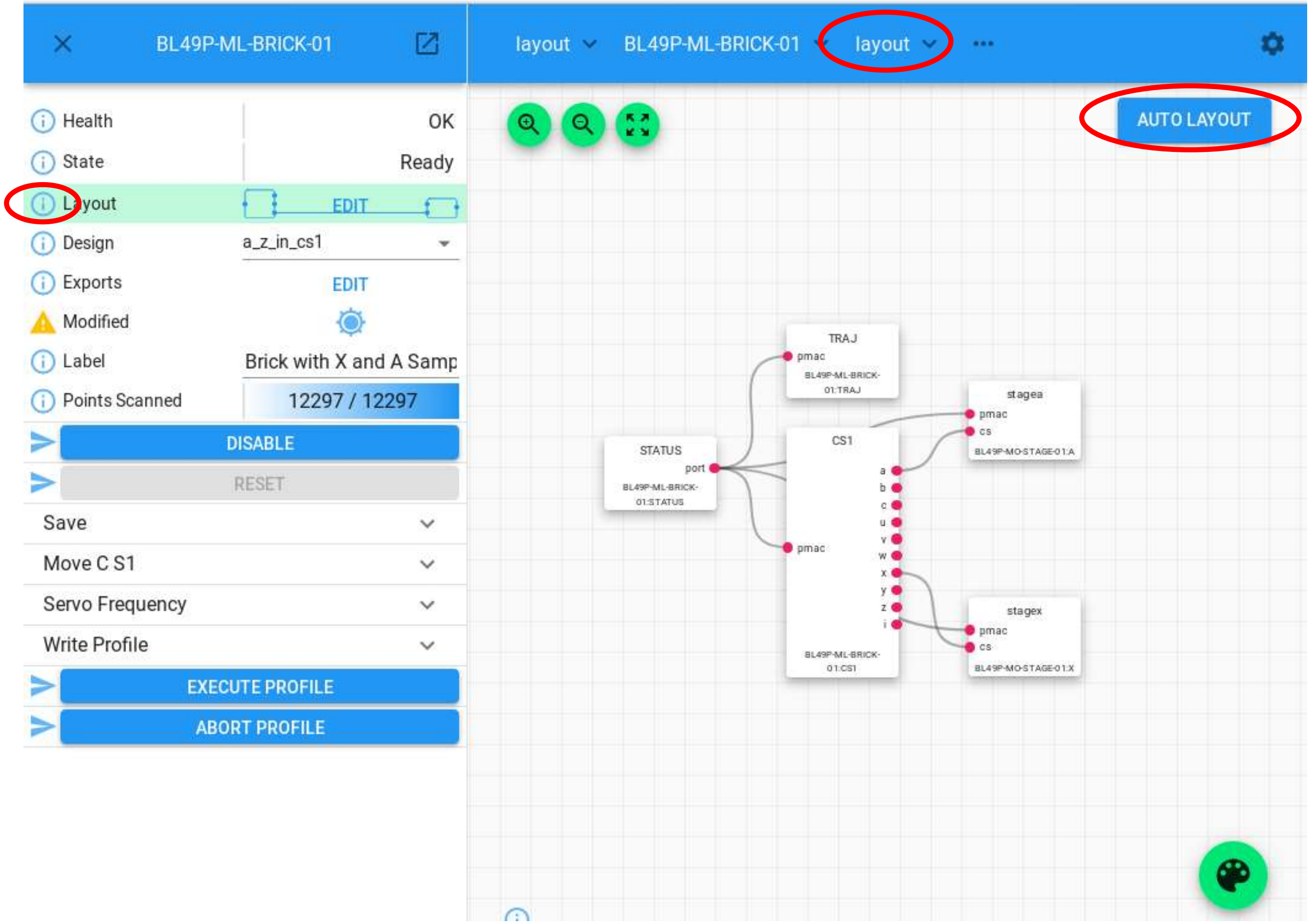


BL49P-ML-BRICK-01

Health	OK
State	Ready
Layout	EDIT
Design	a_z_in_cs1
Exports	EDIT
Modified	
Label	Brick with X and A Samp
Points Scanned	12297 / 12297

Web GUI: Layout View

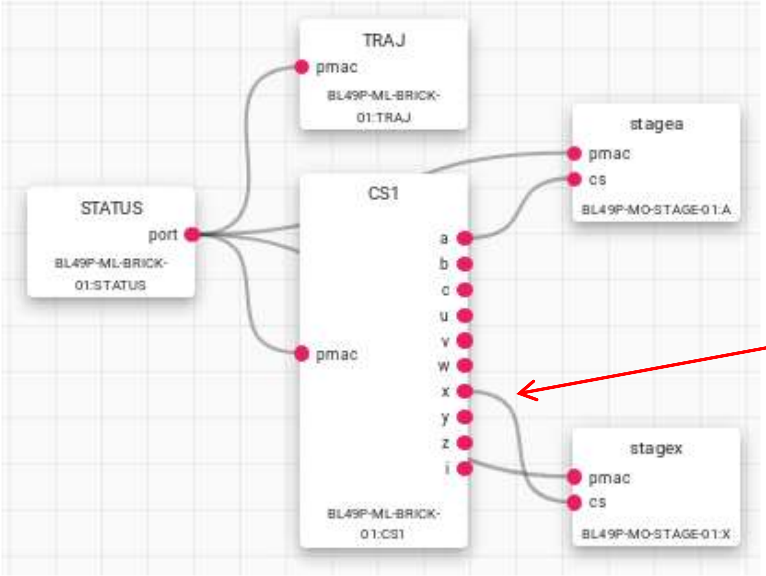
Graphical view of selected root block, with automatic layout feature



BL49P-ML-BRICK-01

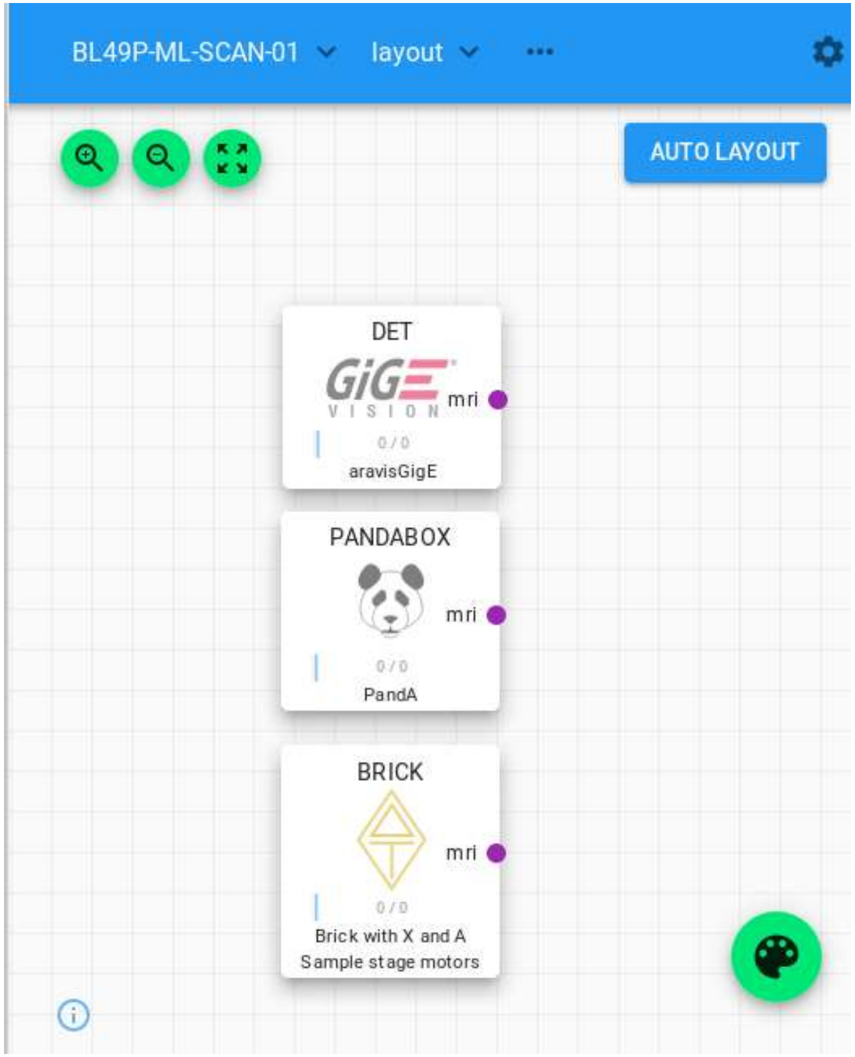
Health	OK
State	Ready
Layout	EDIT
Design	a_z_in_cs1
Exports	EDIT
Modified	
Label	Brick with X and A Samp
Points Scanned	12297 / 12297

Web GUI: Layout View



‘Wire-up’ components visually

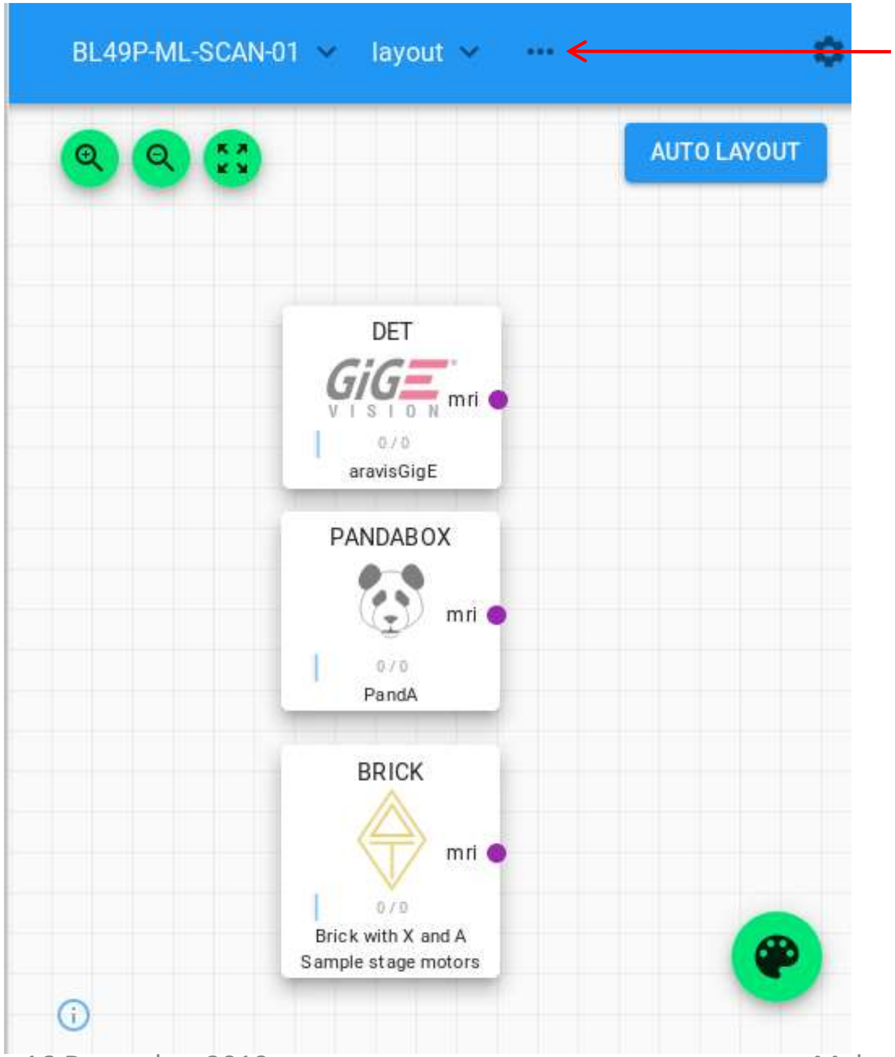
Web GUI: Layout View



Auto Layout – first step for complex layouts

Palette – drag and drop new items

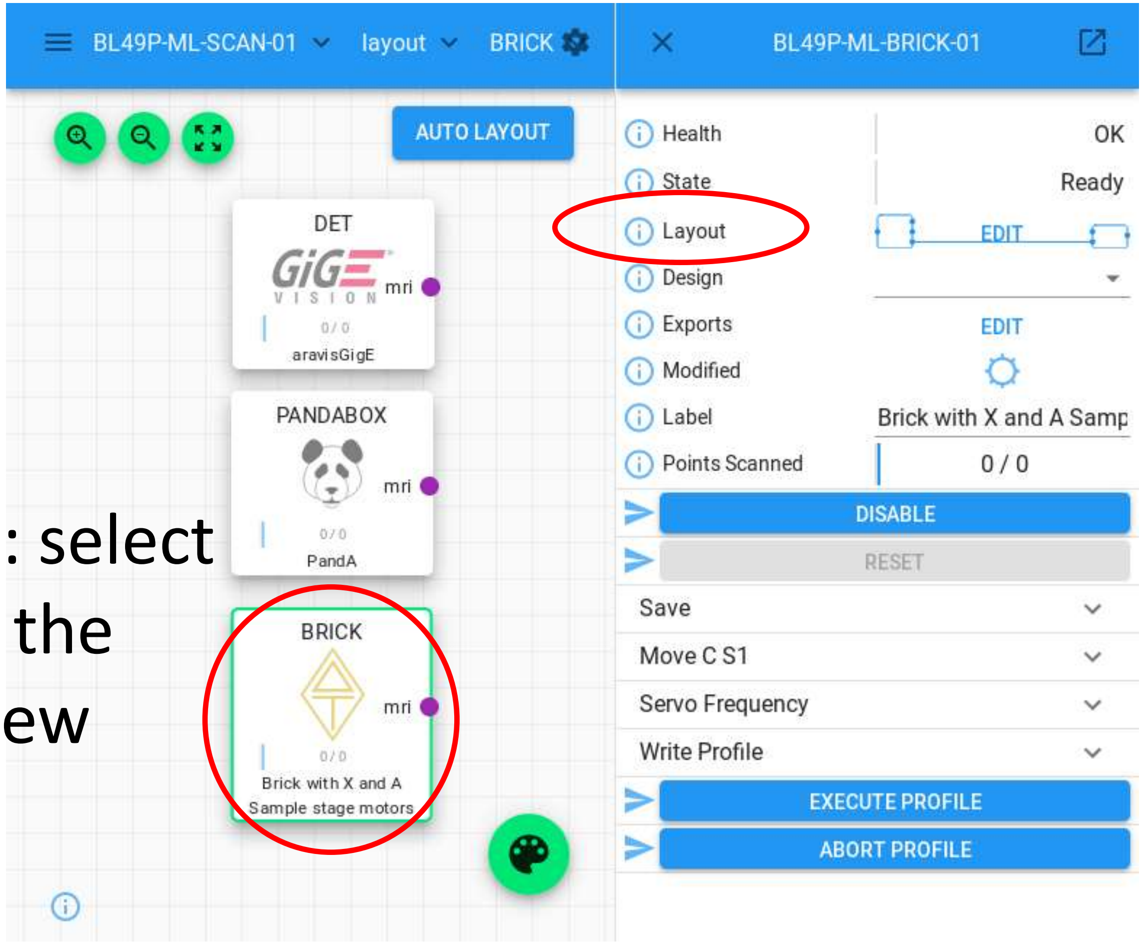
Web GUI: Navigation



Breadcrumb trail in central panel

Starting with the selected root block, you can drill down into the design

Web GUI: Navigation



Health	OK
State	Ready
Layout	EDIT
Design	
Exports	EDIT
Modified	
Label	Brick with X and A Samp
Points Scanned	0 / 0

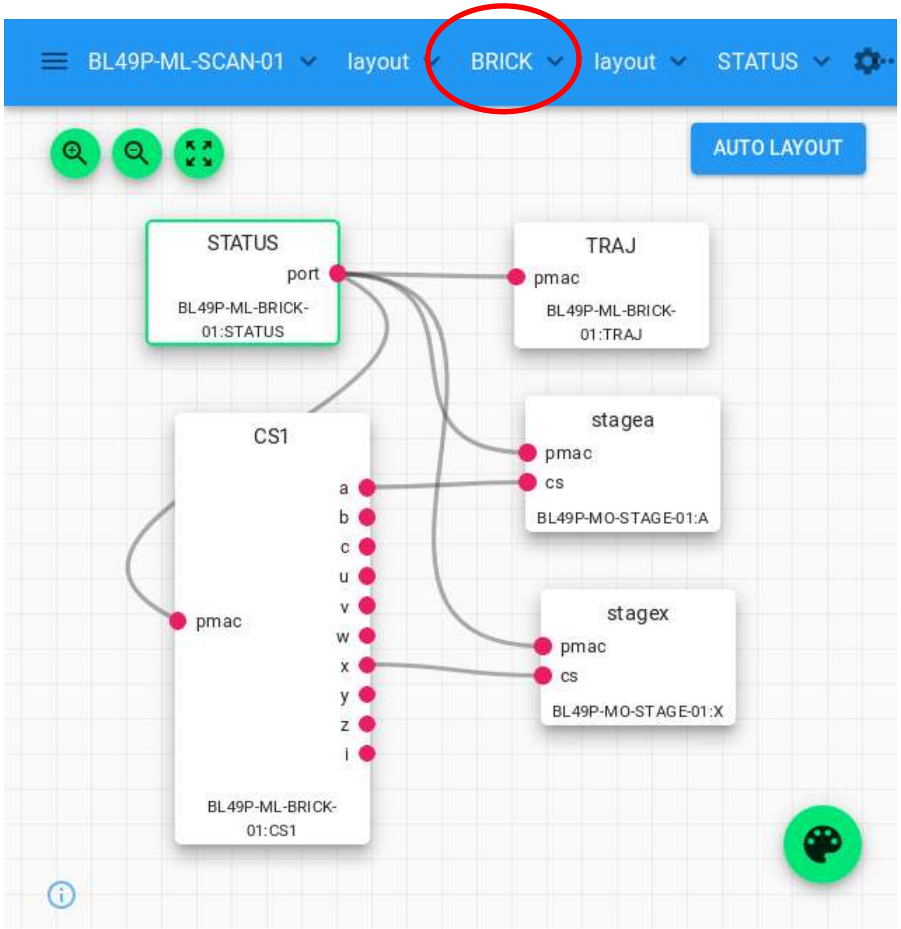
Example: select *BRICK* in the layout view

Then choose *Layout* in the right hand inspector

Web GUI: Navigation

Now focused on the BRICK block:

1. Controller
2. Trajectory
3. Co-ordinate system
4. Axes
5. Status

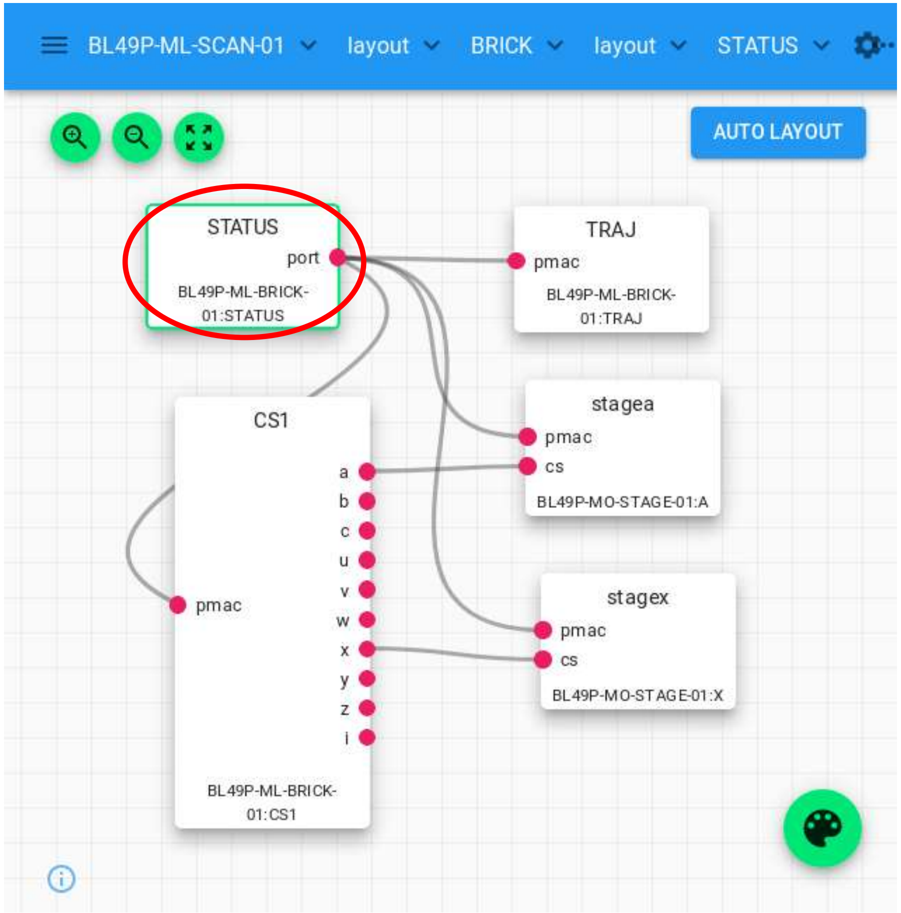


Web GUI: Navigation

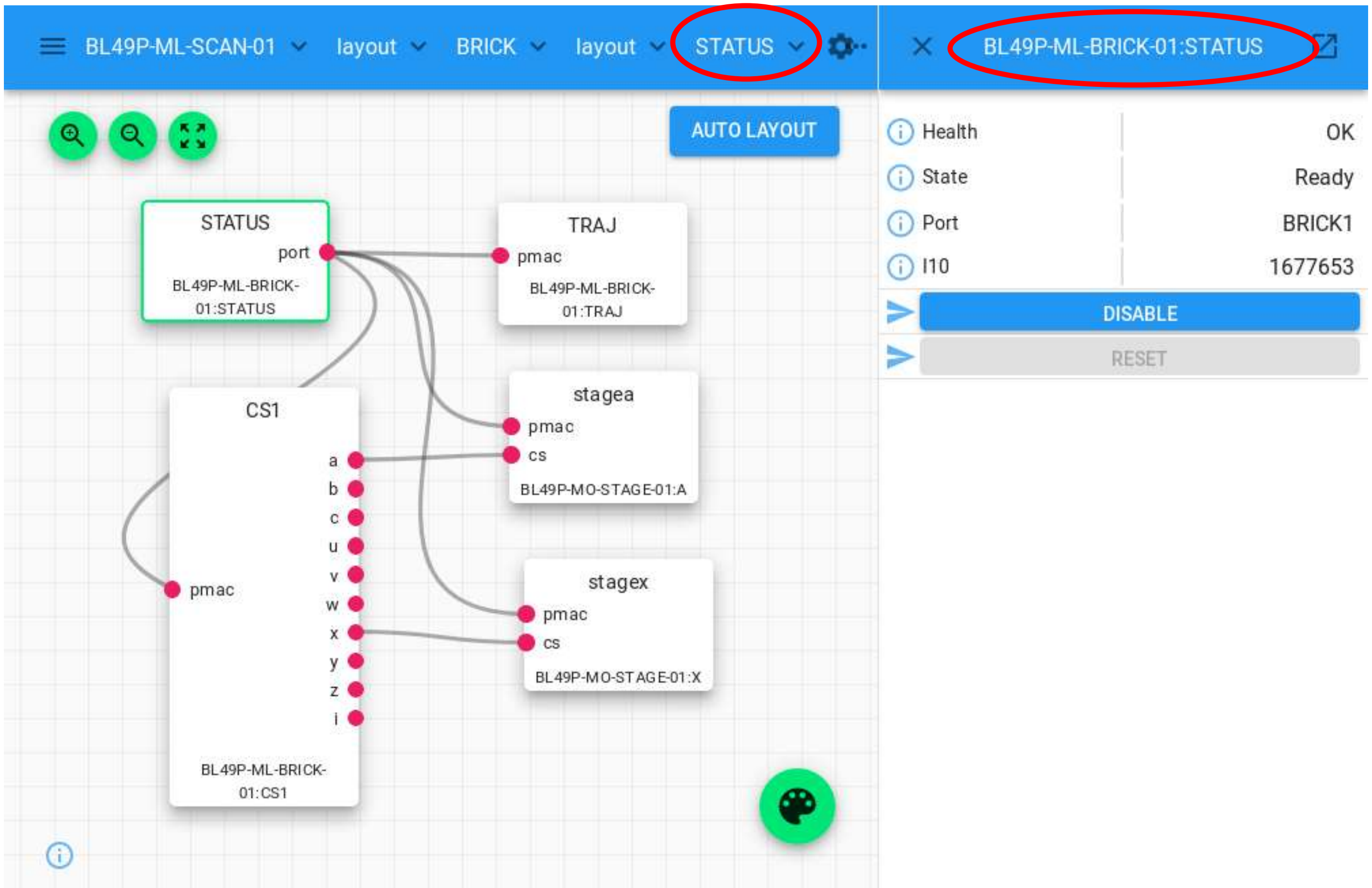
Now focused on the BRICK block:

1. Controller
2. Trajectory
3. Co-ordinate system
4. Axes
5. Status

Select STATUS



Web GUI: Navigation



Health	OK
State	Ready
Port	BRICK1
I10	1677653

Exercise: Web GUI

- Navigate to the PANDA block
- Try to toggle on/off the light controlled by TTLOUT2

- For more info on the web GUI: https://malcolmjs.readthedocs.io

Designs

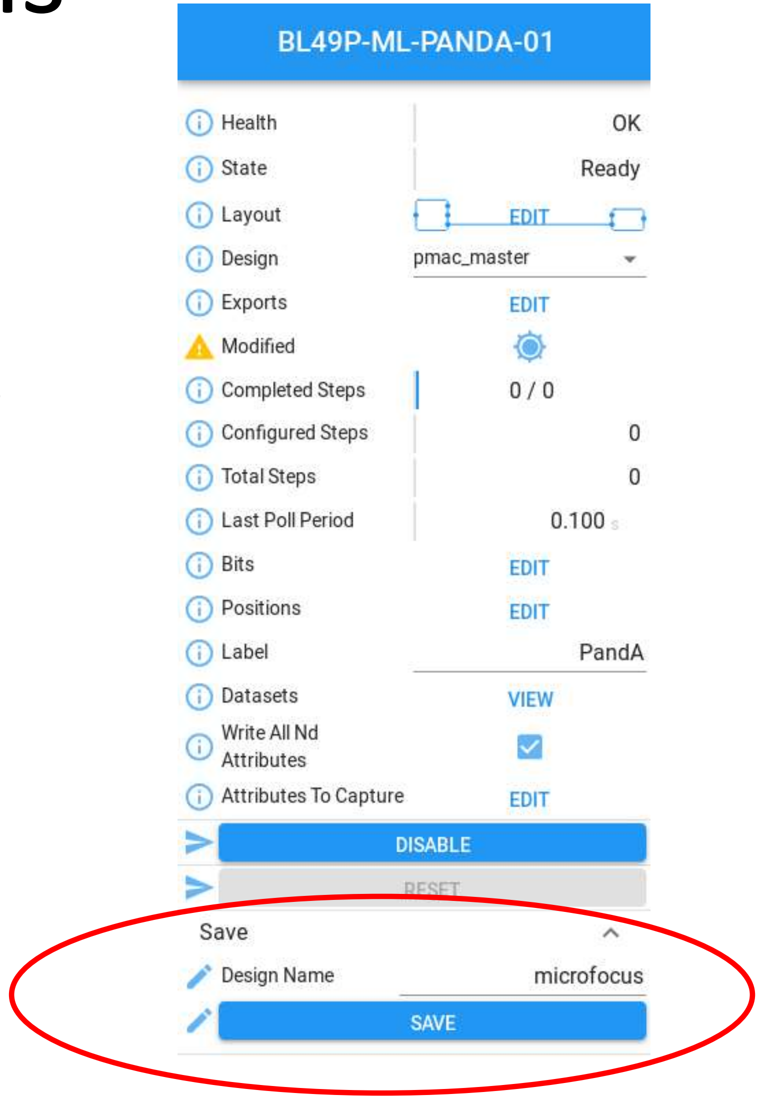
Different scan requirements can be implemented using multiple designs

Example: a new experiment requires a different PandA configuration

Step 1: Open the PANDA block in the layout view

Step 2: Modify the design as required

Step 3: Choose a name and save



BL49P-ML-PANDA-01

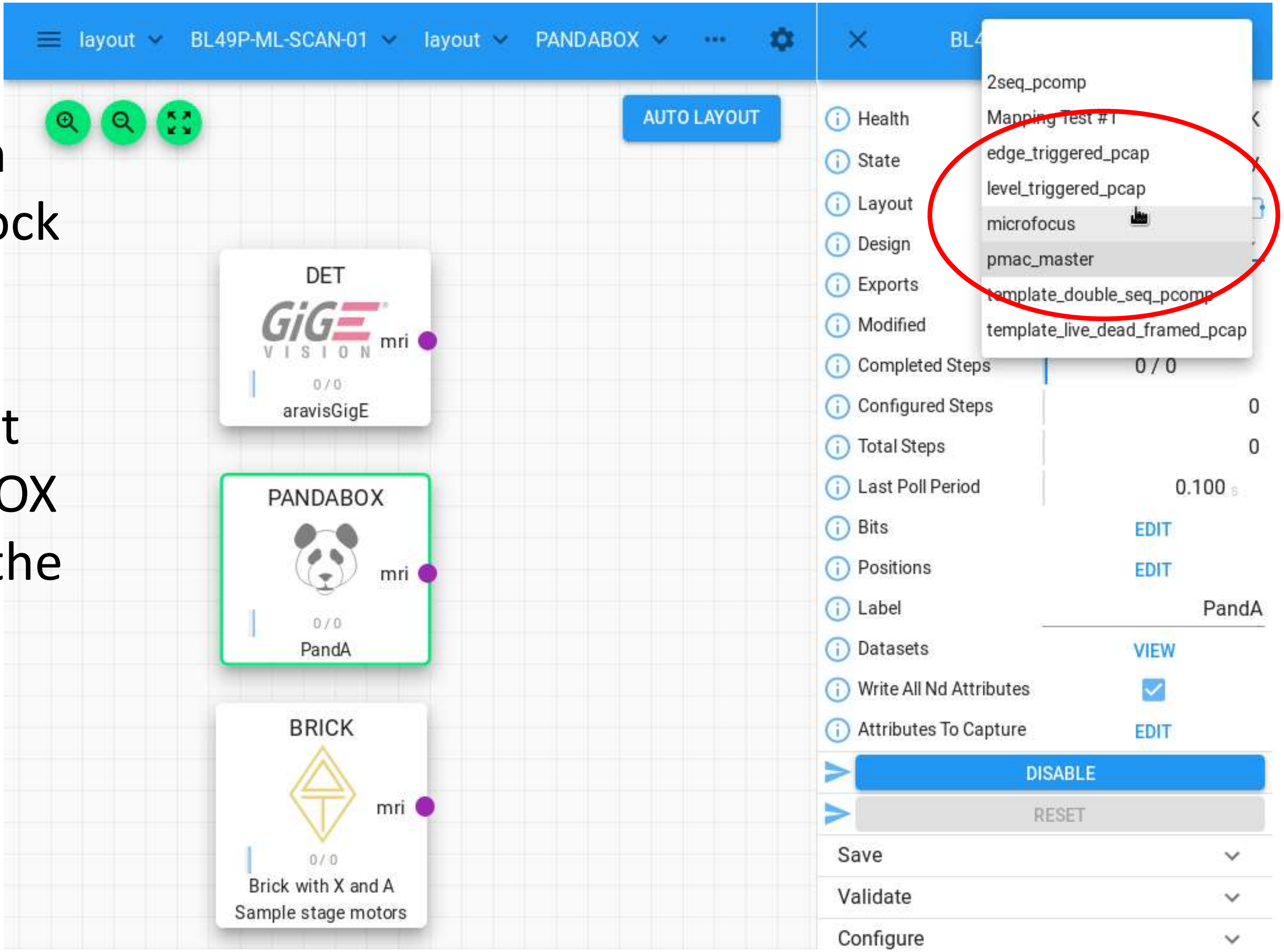
Health: OK
State: Ready
Layout: EDIT
Design: pmac_master
Exports: EDIT
Modified
Completed Steps: 0 / 0
Configured Steps: 0
Total Steps: 0
Last Poll Period: 0.100 s
Bits: EDIT
Positions: EDIT
Label: PandA
Datasets: VIEW
Write All Nd Attributes: ☑
Attributes To Capture: EDIT

Save
Design Name: microfocus

Designs

Step 4: Open the SCAN block layout

Step 5: Select the PANDABOX and choose the new design



Completed Steps 0 / 0

Configured Steps 0

Total Steps 0

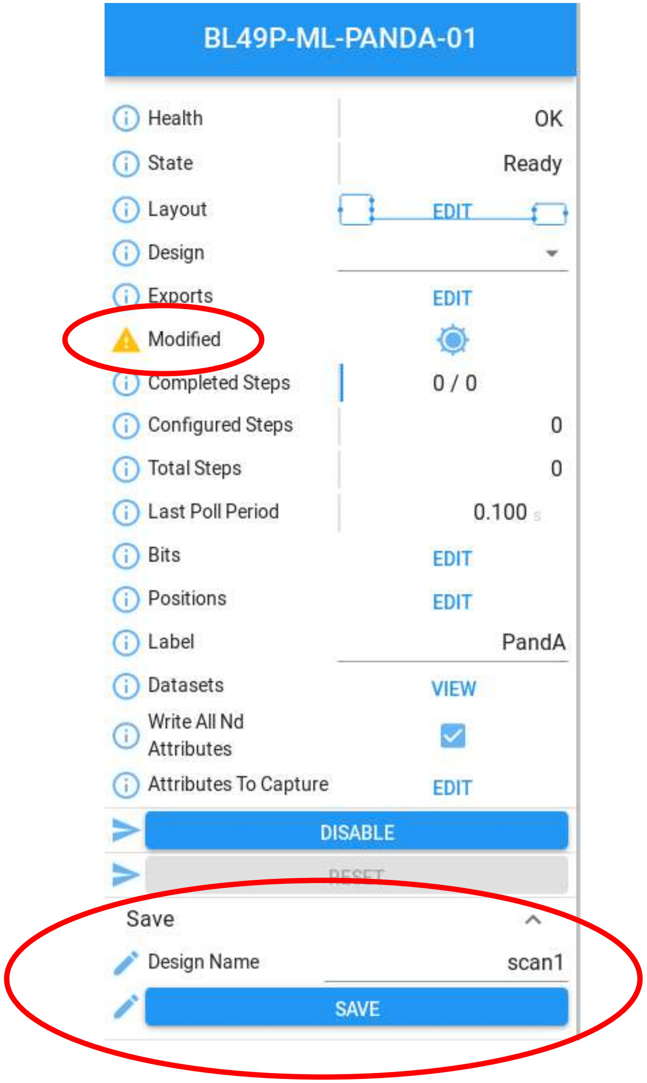
Last Poll Period 0.100 s

Label PandA

Write All Nd Attributes ☑

Designs

Step 6: Using the left hand panel, save this SCAN configuration as a new design



BL49P-ML-PANDA-01

Health: OK

State: Ready

Layout: EDIT

Design:

Exports: EDIT

Modified

Completed Steps: 0 / 0

Configured Steps: 0

Total Steps: 0

Last Poll Period: 0.100 s

Bits: EDIT

Positions: EDIT

Label: PandA

Datasets: VIEW

Write All Nd Attributes: ☑

Attributes To Capture: EDIT

DISABLE

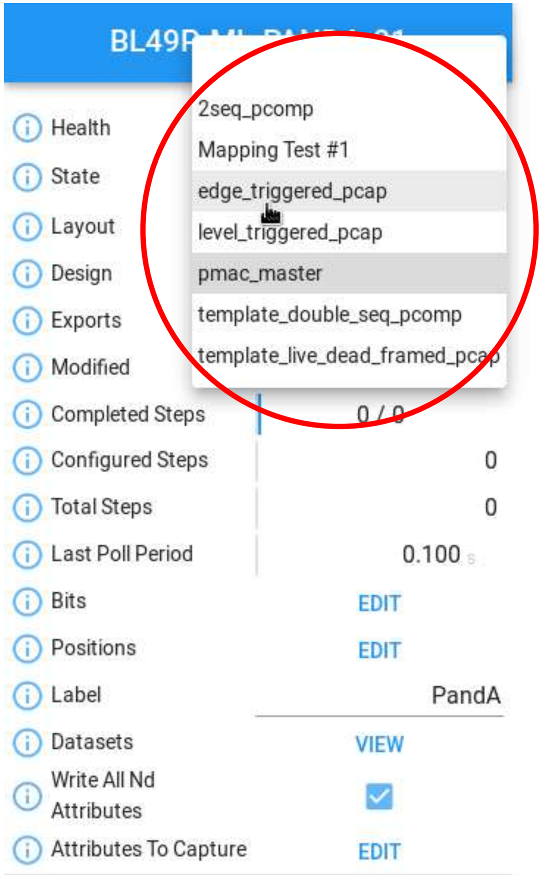
RESET

Save

Design Name: scan1

SAVE

Designs



BL49P

- 2seq_pcomp
- Mapping Test #1
- edge_triggered_pcap
- level_triggered_pcap
- pmac_master
- template_double_seq_pcomp
- template_live_dead_framed_pcap

Health	
State	
Layout	
Design	
Exports	
Modified	
Completed Steps	0 / 0
Configured Steps	0
Total Steps	0
Last Poll Period	0.100 s
Bits	EDIT
Positions	EDIT
Label	PandA
Datasets	VIEW
Write All Nd Attributes	☑
Attributes To Capture	EDIT

To load a design in the Web GUI:

1. Navigate to the root block
2. Select the design from the drop down list in the left hand panel

Malcolm Configuration

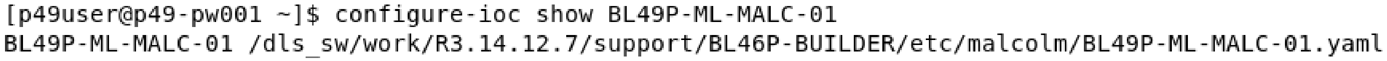
- Blocks are reusable components that require configuration

  Examples:

  - PV prefix (for EPICS devices), IP address (for PandA)
  - Initial design
  - Runtime config directories

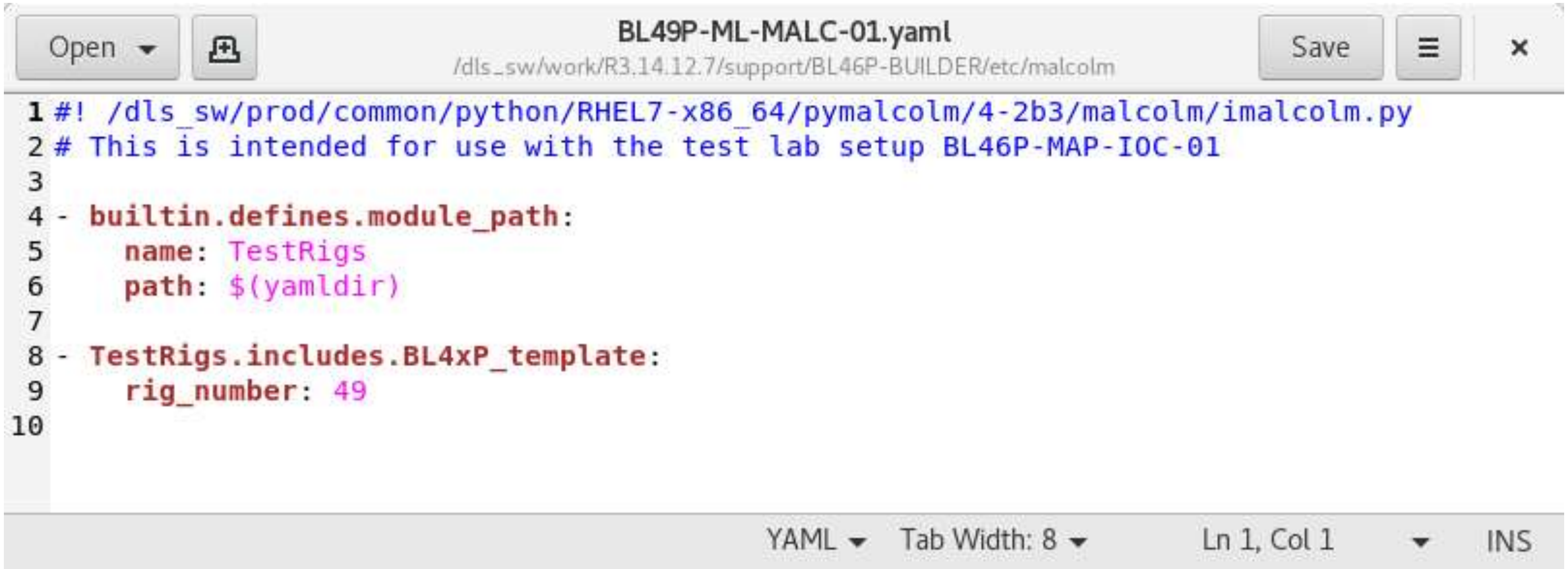
- Configuration specified in YAML

  (‘**Y**AML **A**in’t **M**arkup **L**anguage’!)



```
[p49user@p49-pw001 ~]$ configure-ioc show BL49P-ML-MALC-01
BL49P-ML-MALC-01 /dls_sw/work/R3.14.12.7/support/BL46P-BUILDER/etc/malcolm/BL49P-ML-MALC-01.yaml
```

YAML Files

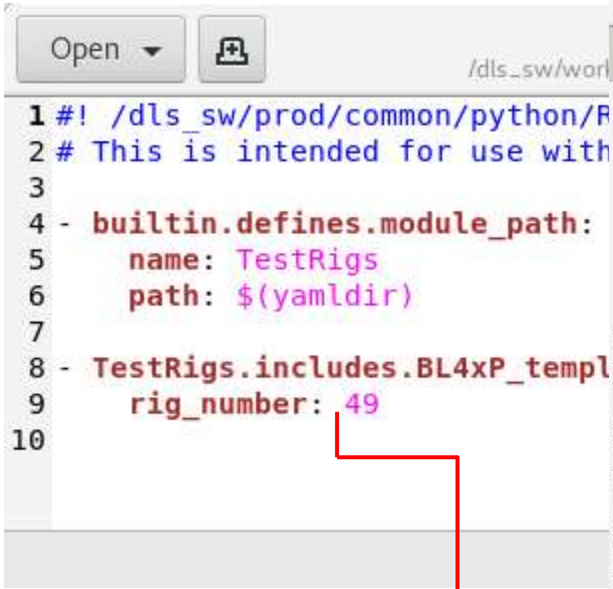


BL49P-ML-MALC-01.yaml
/dls_sw/work/R3.14.12.7/support/BL46P-BUILDER/etc/malcolm

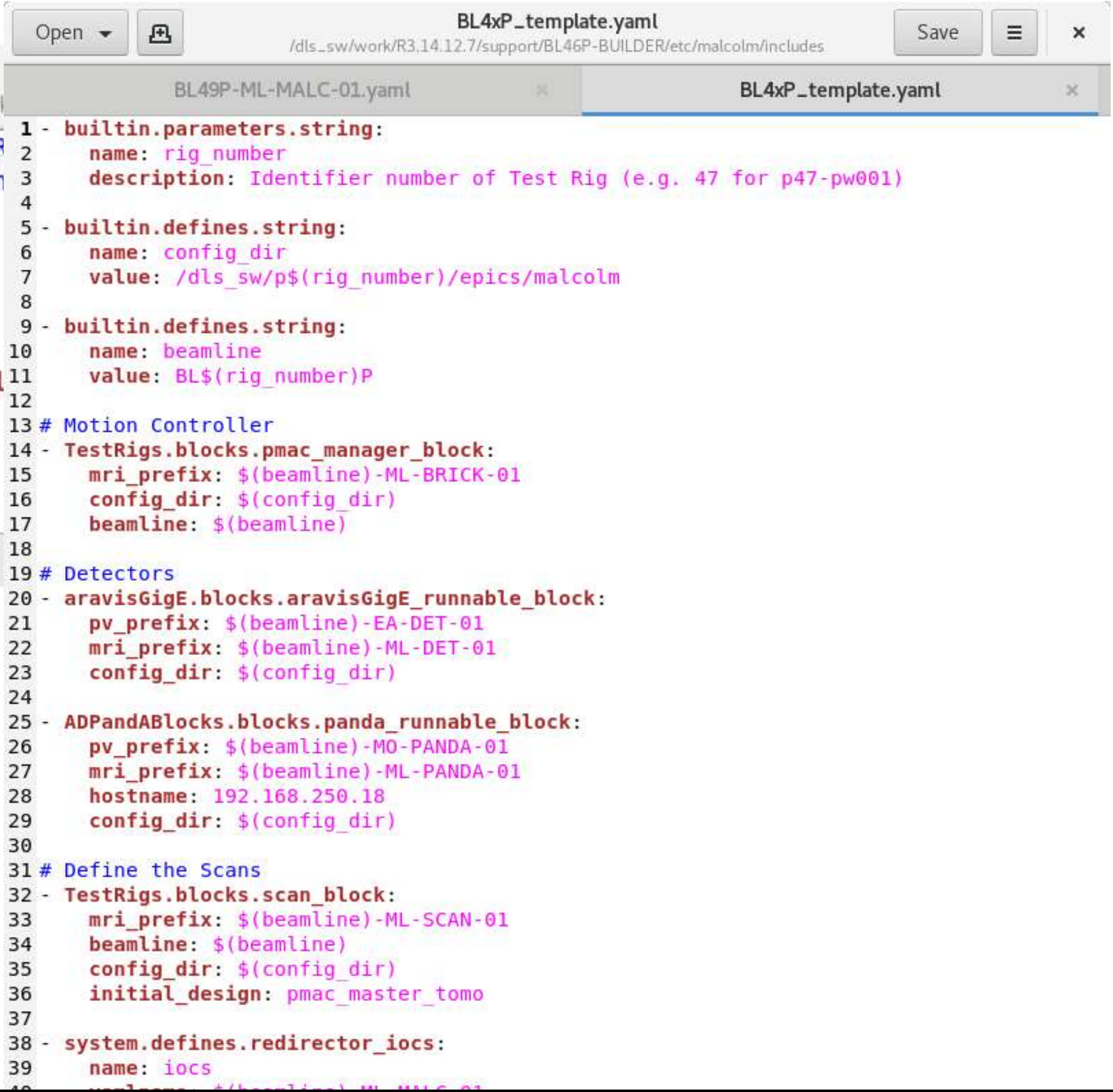
```yaml
#! /dls_sw/prod/common/python/RHEL7-x86_64/pymalcolm/4-2b3/malcolm/imalcolm.py
# This is intended for use with the test lab setup BL46P-MAP-IOC-01

- builtin.defines.module_path:
    name: TestRigs
    path: $(yamldir)

- TestRigs.includes.BL4xP_template:
    rig_number: 49

```

YAML Tab Width: 8 Ln 1, Col 1 INS

YAML Files

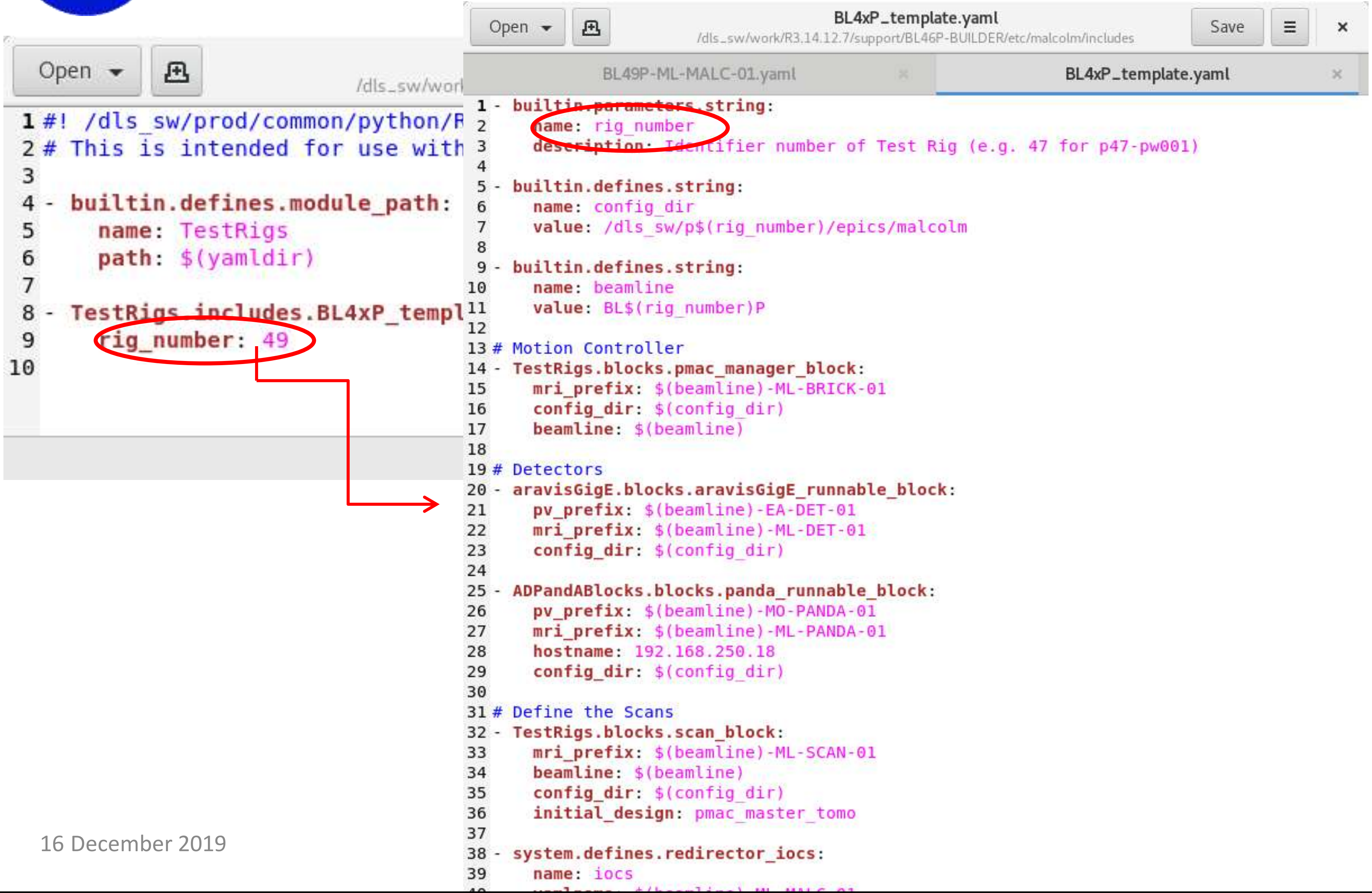


```
#! /dls_sw/prod/common/python/R
# This is intended for use with

- builtin.defines.module_path:
    name: TestRigs
    path: $(yamldir)

- TestRigs.includes.BL4xP_templ
    rig_number: 49

```



```
- builtin.parameters.string:
    name: rig_number
    description: Identifier number of Test Rig (e.g. 47 for p47-pw001)

- builtin.defines.string:
    name: config_dir
    value: /dls_sw/p$(rig_number)/epics/malcolm

- builtin.defines.string:
    name: beamline
    value: BL$(rig_number)P

# Motion Controller
- TestRigs.blocks.pmac_manager_block:
    mri_prefix: $(beamline)-ML-BRICK-01
    config_dir: $(config_dir)
    beamline: $(beamline)

# Detectors
- aravisGigE.blocks.aravisGigE_runnable_block:
    pv_prefix: $(beamline)-EA-DET-01
    mri_prefix: $(beamline)-ML-DET-01
    config_dir: $(config_dir)

- ADPandABlocks.blocks.panda_runnable_block:
    pv_prefix: $(beamline)-MO-PANDA-01
    mri_prefix: $(beamline)-ML-PANDA-01
    hostname: 192.168.250.18
    config_dir: $(config_dir)

# Define the Scans
- TestRigs.blocks.scan_block:
    mri_prefix: $(beamline)-ML-SCAN-01
    beamline: $(beamline)
    config_dir: $(config_dir)
    initial_design: pmac_master_tomo

- system.defines.redirector_iocs:
    name: iocs
```

YAML Files



```
#! /dls_sw/prod/common/python/R
# This is intended for use with

- builtin.defines.module_path:
    name: TestRigs
    path: $(yamldir)

- TestRigs.includes.BL4xP_templ
    rig_number: 49

```

```
- builtin.parameters.string:
    name: rig_number
    description: Identifier number of Test Rig (e.g. 47 for p47-pw001)

- builtin.defines.string:
    name: config_dir
    value: /dls_sw/p$(rig_number)/epics/malcolm

- builtin.defines.string:
    name: beamline
    value: BL$(rig_number)P

# Motion Controller
- TestRigs.blocks.pmac_manager_block:
    mri_prefix: $(beamline)-ML-BRICK-01
    config_dir: $(config_dir)
    beamline: $(beamline)

# Detectors
- aravisGigE.blocks.aravisGigE_runnable_block:
    pv_prefix: $(beamline)-EA-DET-01
    mri_prefix: $(beamline)-ML-DET-01
    config_dir: $(config_dir)

- ADPandABlocks.blocks.panda_runnable_block:
    pv_prefix: $(beamline)-MO-PANDA-01
    mri_prefix: $(beamline)-ML-PANDA-01
    hostname: 192.168.250.18
    config_dir: $(config_dir)

# Define the Scans
- TestRigs.blocks.scan_block:
    mri_prefix: $(beamline)-ML-SCAN-01
    beamline: $(beamline)
    config_dir: $(config_dir)
    initial_design: pmac_master_tomo

- system.defines.redirector_iocs:
    name: iocs
```

Exercise: pvAccess

- Malcolm<->GDA interface uses *pvAccess* (EPICS V4)
- Each block is represented as a V4 process variable (pv) with a number of fields

- Use *pvget* to dump the entire SCAN block structure

```
pvget BL4xP-ML-SCAN-01 -r “”
```

- (If *pvget* not in PATH, run module load pvatools)

pvAccess continued

- -r ‘*request*’ option specifies which fields to return

  pvget BL4xP-ML-SCAN-01 –r health.value

- What information comes back for the *layout* field?

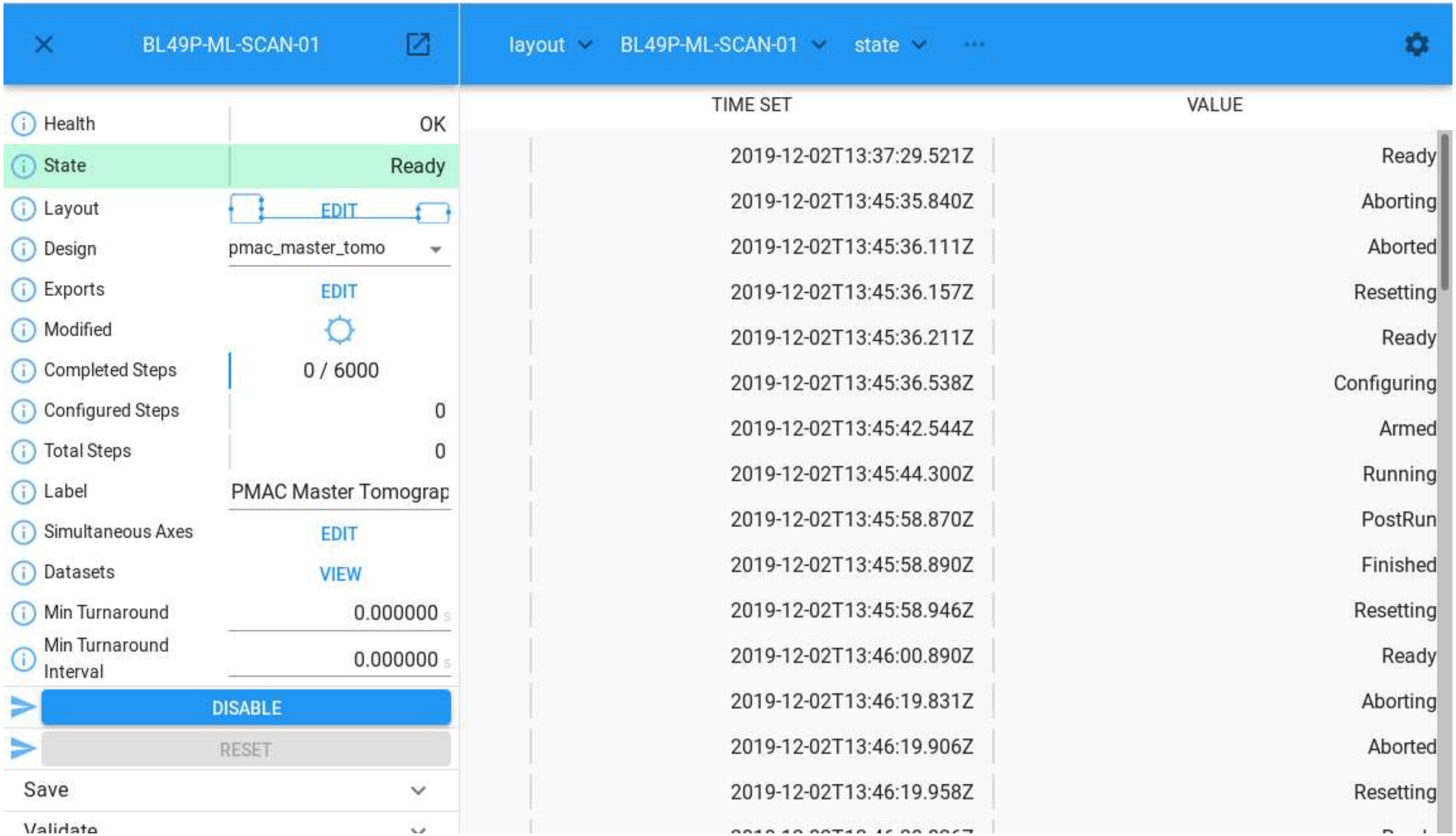
- Other useful SCAN block fields include:
  - totalSteps
  - completedSteps
  - state

Exercise: Scan States

- Start a scan in GDA and monitor the SCAN block in the Malcolm web GUI. What states does it go through?

- The -m (‘*monitor’)* option to pvget monitors the PV for changes. Use this together with the –r option to monitor the *completedSteps* field from the command line as the scan progresses.

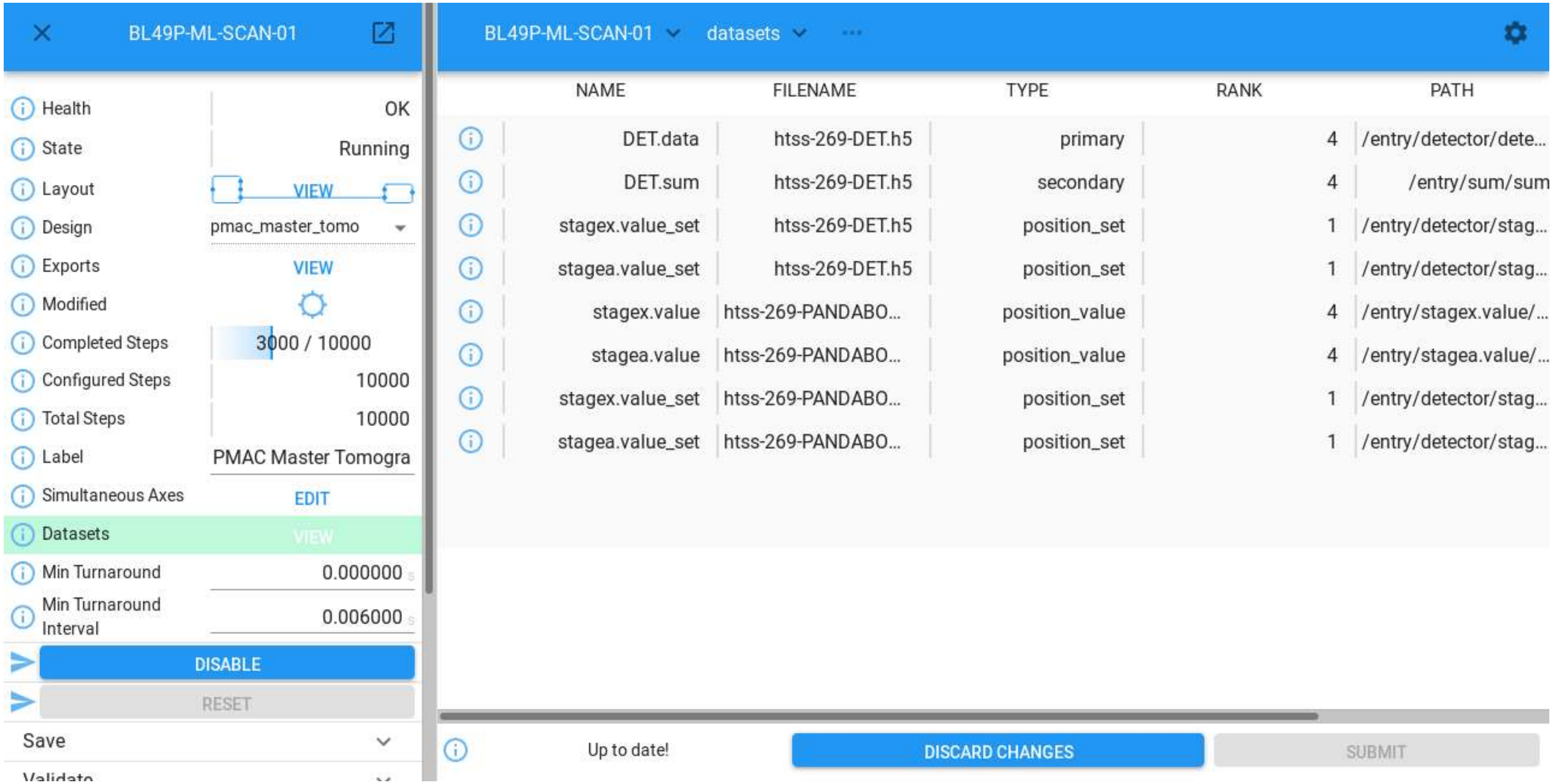
Scan States Screenshot



Health	OK
State	Ready
Layout	EDIT
Design	pmac_master_tomo
Exports	EDIT
Modified	
Completed Steps	0 / 6000
Configured Steps	0
Total Steps	0
Label	PMAC Master Tomograp
Simultaneous Axes	EDIT
Datasets	VIEW
Min Turnaround	0.000000 s
Min Turnaround Interval	0.000000 s

TIME SET	VALUE
2019-12-02T13:37:29.521Z	Ready
2019-12-02T13:45:35.840Z	Aborting
2019-12-02T13:45:36.111Z	Aborted
2019-12-02T13:45:36.157Z	Resetting
2019-12-02T13:45:36.211Z	Ready
2019-12-02T13:45:36.538Z	Configuring
2019-12-02T13:45:42.544Z	Armed
2019-12-02T13:45:44.300Z	Running
2019-12-02T13:45:58.870Z	PostRun
2019-12-02T13:45:58.890Z	Finished
2019-12-02T13:45:58.946Z	Resetting
2019-12-02T13:46:00.890Z	Ready
2019-12-02T13:46:19.831Z	Aborting
2019-12-02T13:46:19.906Z	Aborted
2019-12-02T13:46:19.958Z	Resetting

Dataset table



Health	OK
State	Running
Layout	VIEW
Design	pmac_master_tomo
Exports	VIEW
Modified	
Completed Steps	3000 / 10000
Configured Steps	10000
Total Steps	10000
Label	PMAC Master Tomogra
Simultaneous Axes	EDIT
Datasets	VIEW
Min Turnaround	0.000000 s
Min Turnaround Interval	0.006000 s

NAME	FILENAME	TYPE	RANK	PATH
DET.data	htss-269-DET.h5	primary	4	/entry/detector/dete...
DET.sum	htss-269-DET.h5	secondary	4	/entry/sum/sum
stagex.value_set	htss-269-DET.h5	position_set	1	/entry/detector/stag...
stagea.value_set	htss-269-DET.h5	position_set	1	/entry/detector/stag...
stagex.value	htss-269-PANDABO...	position_value	4	/entry/stagex.value/...
stagea.value	htss-269-PANDABO...	position_value	4	/entry/stagea.value/...
stagex.value_set	htss-269-PANDABO...	position_set	1	/entry/detector/stag...
stagea.value_set	htss-269-PANDABO...	position_set	1	/entry/detector/stag...

Up to date!