

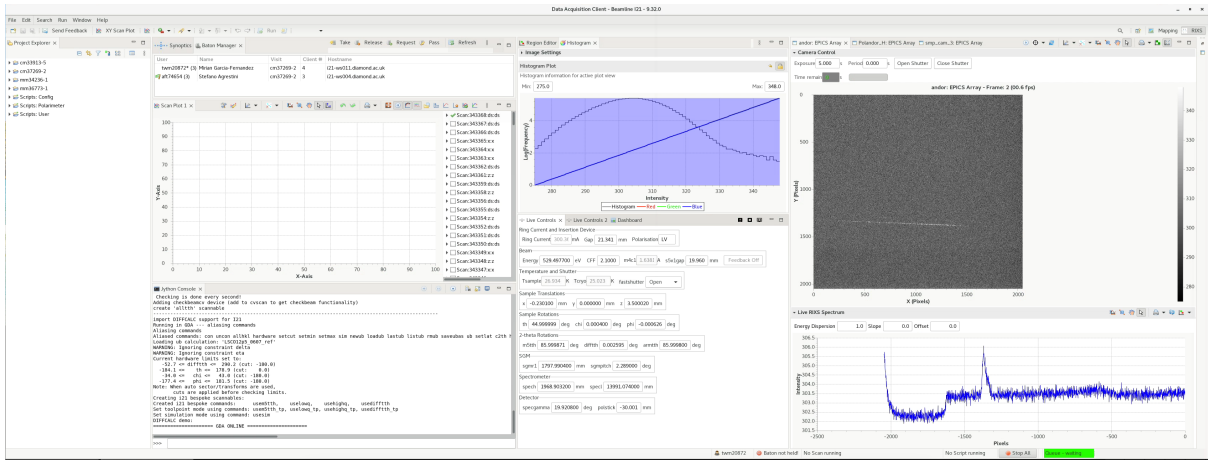
Basic GDA commands

Table of Contents

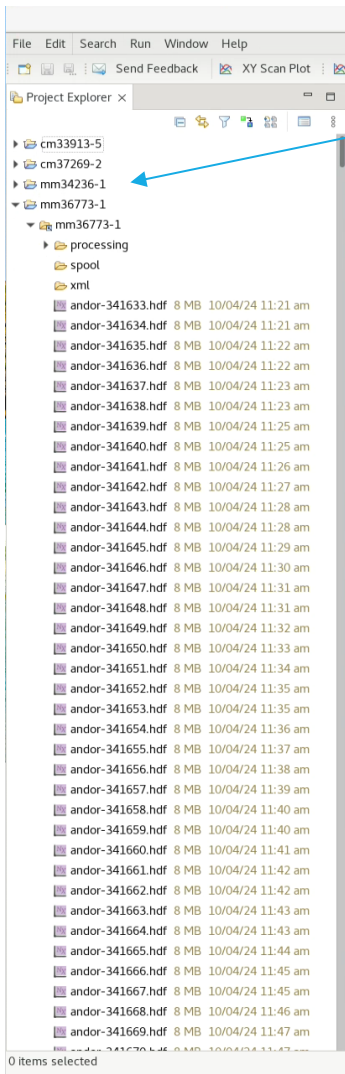
1.- Moving motors	2
2.- Opening and closing beamline valves	3
3.- Fast shutter modes	4
4.- Changing energy and polarization	5
5.- Changing size of exit slits.....	6
6.- Collecting an image with andor detector	6
7.- On the fly XAS scanning.....	6
8.- Scanning.....	7
Absolute scans.....	7
Relative scans	7
theta / 2theta scans	8
9.- Checkbeam.....	8
10.- Collecting I ₀ for normalization of data	8
12.- Scan processing feature	8
13.- Saving sample and carbon tape positions	9
Sample positions.....	9
Carbon tape positions.....	9
Clear positions.....	10
Delete a specific position.....	10

1.- Moving motors

The layout of the I21 GDA looks as follows:

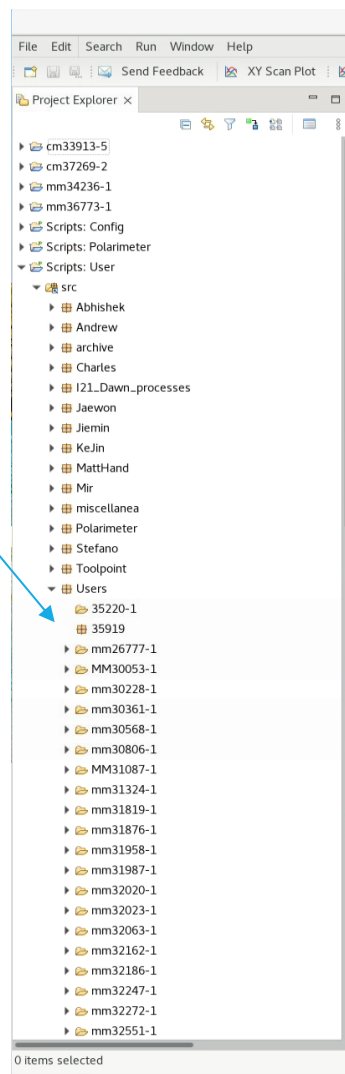


Data folder:

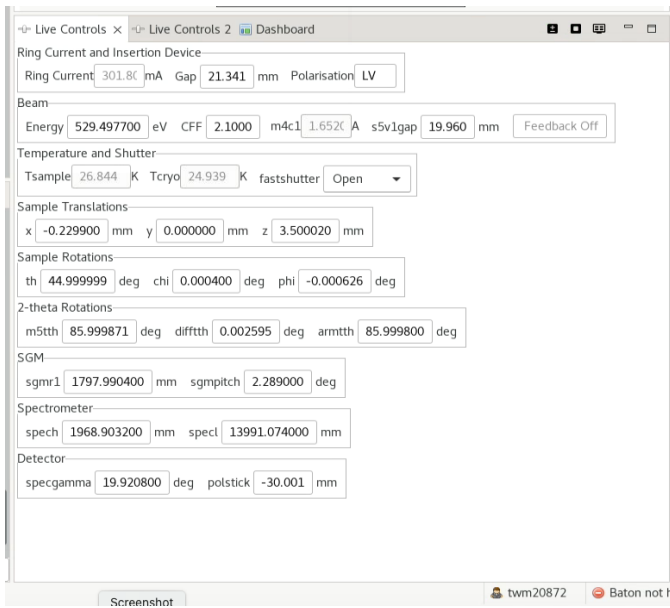


Your folder will have your experiment number.

Scripts folder:



The **Live Controls** panel in GDA shows all the motors you will need to move with their corresponding names



There are two ways to move a motor. The first one is to move to an absolute position:

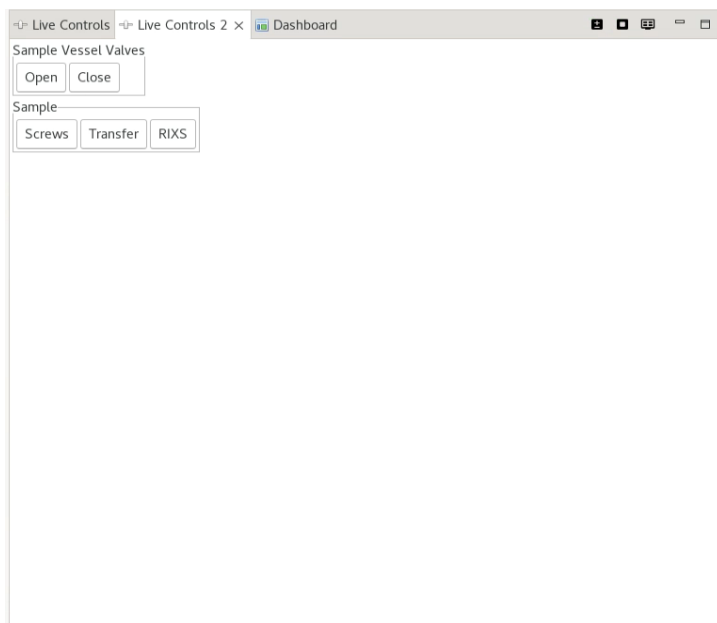
>> pos x 2

The second one is to increase the motor position by a relative amount, either positive or negative

>> inc x 0.2

2.- Opening and closing beamline valves

There is an automatized mode for opening and closing the beamline valves that can be accessed in GDA in the **Live controls 2**:



3.- Fast shutter modes

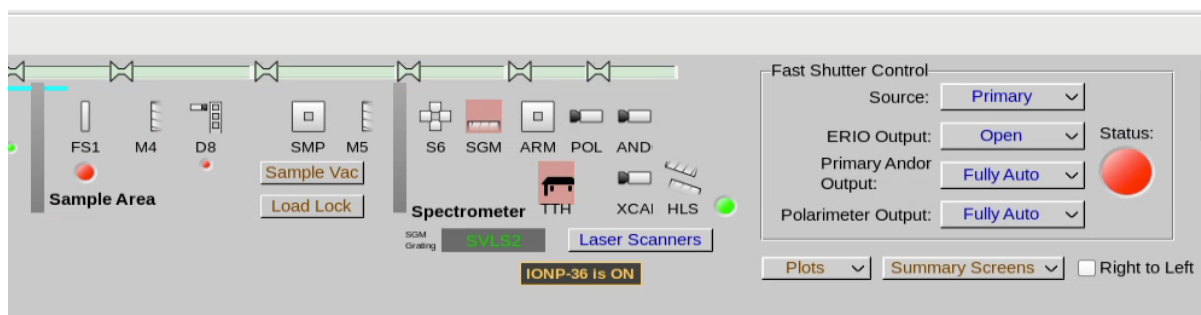
There is a fast shutter at the beamline located before the sample vessel. This fast shutter prevents the sample to be continuously exposed to x-rays while not collecting data. There are three mode of operations of the fast shutter.

Primary mode:

This can be selected by typing in GDA

>> primary()

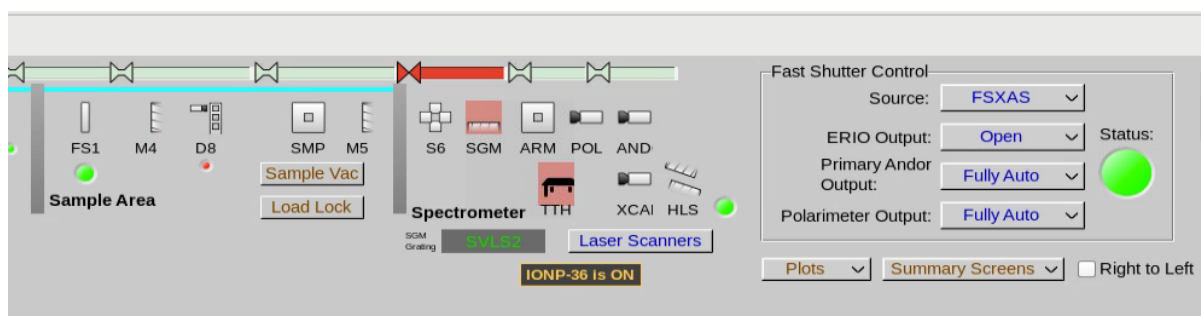
In this mode the fast shutter is synchronized with the CCD detector and it will open and close to collect the RIXS images.



FSXAS mode:

In this mode the fast shutter is opened while the valve after the sample and before the spectrometer arm is closed to prevent exposing the CCD detector. This mode should be used when performing XAS or diffraction measurements. It can be chosen by typing in GDA

>> fsxas()



Erio mode:

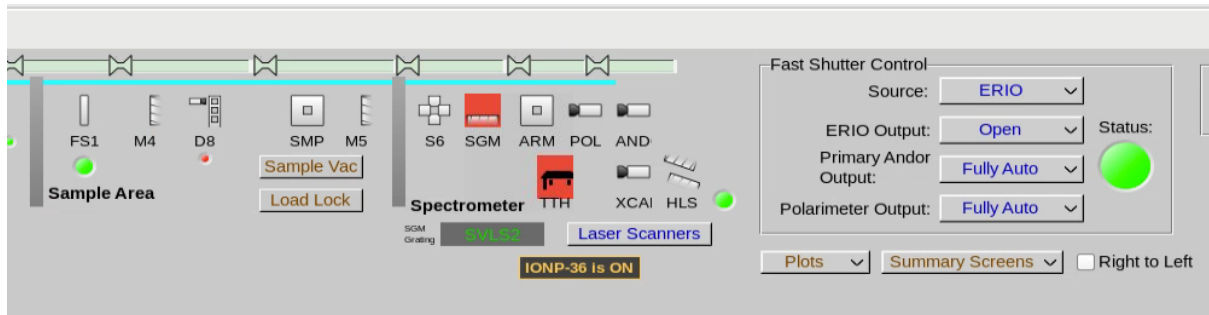
In this mode the fast shutter is controlled by EPICS and not by the CCD detector. It can be chosen by typing in GDA

>> erio()

In this mode we can chose whether the fast shutter is open or close by typing in GDA

>> fastshutter('Open')

>> fastshutter('Close')



4.- Changing energy and polarization

The command **energy** moves both the monochromator and the ID gap to the correct position for the demanded energy.

>> pos energy 531

To change the **polarization** one must use the following command.

For **Linear Vertical** polarization:

>> go(531, LV)

For **Linear Horizontal** polarization:

>> go(531, LH)

For **Circular Right** polarization:

>> go(531, CR)

For **Circular Left** polarization:

>> go(531, CL)

For the **intermediate energy range**, i.e. between 2000 and 3000 we need to use the third harmonic, hence the command will be as follows

For **Linear Vertical** polarization:

>> go(2531, LV3)

For **Linear Horizontal** polarization:

>> go(2531, LH3)

5.- Changing size of exit slits

The motor to move the exit slits is **s5v1gap** and the units are **microns**. This motor has an alias that is **eslit**. So to set the opening of the exit slits to 20 microns for instance:

```
>> pos eslit 20
```

6.- Collecting an image with andor detector

There are two ways to collect an image with the Andor detector:

```
>> get_rixs(sample_no_images, exposure_time)
```

```
>> get_rixs(1, 60)
```

```
>> acquireRIXS sample_no_images andor exposure_time
```

```
>> acquireRIXS 1 andor 60
```

In both cases, 1 corresponds to the number of images to acquire and 60 is the time of exposure in seconds. These commands will automatically set the fast shutter mode to `primary()`

7.- On the fly XAS scanning

In I21, it has been implemented the feature of scanning the energy while both the ID and the PGM continuously move. This allows for a much faster acquisition of the XAS signal. In case that better statistics is required, one can simply acquire several scans and add them up. To measure the continuous energy scan one should use the following command:

```
>> repeat_xas(start_energy, end_energy, number_of_scans)
```

```
>> repeat_xas(910, 960, 5)
```

This will launch a total of 5 continuous energy scans from 910 eV to 960 eV. This command will automatically set the fast shutter mode to `fsxas()`

To change Femto Gain if the diode signal or drain current is too small or is saturating, the value must be one of (1e3, 1e4, 1e5, 1e6, 1e7, 1e8, 1e9) for "Low Noise" mode. To do that, choose the adequate value and select it by typing into GDA:

```
>> draincurrent_i.setGain(1e8)
```

```
>> diff1_i.setGain(1e9)
```

```
>> fy2_i.setGain(1e9)
```

8.- Scanning

There are different modes in which we can scan a motor with GDA.

Absolute scans

The first one is to scan between two specific positions setting the step size for the scanning and the counter. For instance, to scan energy between 630 eV and 660 eV in steps of 250 meV using the photodiode:

```
>> scan energy 630 660 0.25 diff1
```

If we would want to repeat the same scan but using the Andor detector counting 20 seconds per point:

```
>> scan energy 630 660 0.25 andor 20
```

We could also use two or more counters at the same time. For instance, if we are collecting XAS and we want to measure both the signal in the photodiode and the drain current. We just need to add the other counter to the scan.

```
>> scan energy 630 660 0.25 diff1 draincurrent
```

One must remember that at the end of this scan the motor remains in the position of the last point.

When using these counters we only use the preset gain in the femto boxes. To use in autogain, integration mode we need to use different names for the counters and in that case we need to add the counting time.

```
>> scan energy 630 660 0.25 diff1_i 1
```

```
>> scan energy 630 660 0.25 diff1_i draincurrent_i 1
```

Relative scans

We can also measure scans relative to the current position of the motor. The advantage of this type of scan is that at the end of the scan, the motor will come back to the position it had before the scan.

For instance, if we were at $E = 643$ and we would like to repeat the previous scan between 630 and 660 in steps of 25 meV but in relative mode:

```
>> rscan energy -13 17 0.25 diff1_i 1
```

Analogously with the Andor detector:

```
>> rscan energy 630 660 0.25 andor 20
```

We can also launch relative scans setting the number of points of the scan instead of the step (spec style). For the same scan as the previous with 100 points:

```
>> dscan energy 630 660 100 diff1_i 1
```

Analogously with the Andor detector:

```
>> dscan energy 630 660 100 andor 20
```

theta / 2theta scans

Sometimes, especially for diffraction measurements is useful to scan two motors at the same time as for instance one does in a theta / 2 theta scan where the step of the two theta motor is double of the step of the theta motor.

```
>> rscan th -4 4 0.2 diffth -8 0.4 diff1_i 1
```

If we include an ending point for the second motor, then the scan will run like a mesh scan.

9.- Checkbeam

The checkbeam function will pause the scan during the refills and if the beam is lost. We just need to add the function at the end of the scan.

```
>> rscan energy 630 660 0.25 andor 20 checkbeam
```

The on the fly XAS scan already includes the checkbeam feature.

10.- Collecting I₀ for normalization of data

The I₀ is measured as the drain current in the refocussing mirror 4 before the sample. The counter for this is m4c1 and we just need to add to the scan this counter with the same counting time as the CCD exposure in case we want to collect the I₀.

```
>> rscan energy 630 660 0.25 andor 20 checkbeam m4c1 20
```

12.- Scan processing feature

Often it could be useful to use the scan processing feature included in GDA. To switch it on one must type:

```
>> scan_processing_on()
```

This feature fits the scan that is collected, and it gives back the peak position, maximum intensity position and COM. Then one can move to these positions by simply typing goto:

```
>> goto peak
```

```
>> goto COM
```

The scan processing slows the data collection, so once having this feature is no longer helpful, like when collecting RIXS data, it is best to switch it off.

```
>> scan_processing_off()
```


13.- Saving sample and carbon tape positions

There are build-in functions for users to capture both sample and carbon tape positions within a GDA session and use them to move motors within the session.

Sample positions are stored in a dictionary called '*sample_pos*'. Multiple sample positions can be captured into this dictionary with any given name. If a given name already exists its value will be updated, otherwise a new item will be added.

Carbon tape positions are stored in another dictionary called '*carbon_tape_pos*'. Multiple carbon tape positions can be captured into this dictionary with any given name. If a given name already exists its value will be updated, otherwise a new item will be added.

Sample positions

To capture a sample position from current motor positions, please use function

```
>>save_sample_positions("given_name")
```

or command

```
>> save_sample_positions"given_name"
```

To move to a named sample position, please use function

```
>> move_to_sample_positions("given_name")
```

or command

```
>> move_to_sample_positions"given_name"
```

The default name in defining function is '**sample**'.

Carbon tape positions

To capture a carbon tape position from current motor positions, please use function

```
>> save_carbon_tape_positions("given_name")
```

or command

```
>> save_carbon_tape_positions "given_name"
```

To move to a named carbon tape position, please use function

```
>>move_to_carbon_tape_positions("given_name")
```

or command

```
>> move_to_carbon_tape_positions "given_name"
```

The default name in defining function is '**carbon**'.

Clear positions

```
>> sample_pos.clear()
>> carbon_tape_pos.clear()
```

Delete a specific position position

```
>> del sample_pos["specific_position_name"]
>> del carbon_tape_pos["specific_position_name"]
```