UCL Helium Line

Running instructions for the UCL Helium Line
Functional Description of the PyMS application

By **Dr Gary Twinn**

London Geochronology Centre

Contents

List of Figures

List of Tables

Helium Line Sample Change

During the sample change procedures detailed below care must be taken that air at atmospheric pressure does not enter Volume A or the Hot Getter volume, both of which will be permanently damaged should this happen. When carrying out sample changes it is imperative that Valves 8,10,11 and 13 are closed on the PyMS Control Panel (closed valves are coloured RED), this should happen automatically as the unload batch is the last process PyMS runs. Valves 1,2,3 and 4 (Helium Tank pipettes) should never be manually opened, and it is even more important during sample changes not to admit air into the pipettes. Do not open these valves. The manual Swagelok valve described below, and pictured in (Figure 1), is located between the box housing the laser and valve 10. Do not confuse it for the depletion tank manual valves under no circumstances open either of these two valves.

Sample loading

If samples are currently loaded into the sample chamber, follow the sample unload procedure first.

When loading samples there is potential for leaks into the laser port. The procedure below must be followed carefully to ensure that the vacuum pumps do not see a pressure higher than they are designed to cope with. Atmospheric pressure will damage the turbo pumps.

1. Check that the manual Swagelok valve (Figure 1) is closed. This valve needs to be hand tight, but not over tightened. Exerting too much force will distort the valve seal, making it increasingly difficult to obtain a vacuum tight seal
2. Using the PyMS Control Panel, check Valves 10 (Laser port), 11 (Hot Getter) and 13 (Turbo Pump) are closed (red). Check that valve 8 (between volumes A and B) is closed. This is extremely important damage to the mass analyser will occur if this valve is left open.
3. Gloves are not necessary for this procedure, but take care not to touch any parts which will be inside the vacuum system this includes the inner edge of the copper gasket; if you need to hold the gasket then carefully hold it on the outer rim only, or wear gloves. All parts placed in the vacuum system (i.e. samples and planchette) must be clean.
4. If the viewport window was replaced after the previous samples were unloaded, remove it (and the gasket) and place the viewport in a safe place, preferably on top of the gasket, without damaging the knife edge on the lower surface, or getting it dirty. Do not remove the bolts they are numbered and in specific positions.
5. A 4 mm bolt is kept in a petri dish next to the laser port. If samples have been unloaded previously it may already be screwed into the planchette. If not, screw this bolt, handling only the cap head end, into one of the threaded holes in the planchette, and carefully lift the planchette into the sample chamber. Align the planchette so that hole number 1 is in the top right of the chamber as you are viewing it (Figure 2). Carefully

remove the bolt. Place a new copper gasket (do not reuse old gaskets!) onto the recess in the sample chamber port. Make sure there are no bits of plastic/hairs/etc on the gasket these will cause a leak.

Figure 1: Manual Swagelok Valve A for rough pumping the laser port.
Anti-clockwise to close, clockwise to open. It is located between the laser housing (left of the picture) and valve 10 (pneumatic tubing towards valve 10 is on the right).

6. Replace the viewport, and do the bolts up by hand until all are evenly tight. The bolts are numbered - do them up in numerical order. By doing the bolts up by hand, the viewport should remain level if it is at an angle a leak becomes more likely.
7. The bolts can now be tightened with either a 13 mm spanner or the hand wrench, in numerical order. Tighten each bolt a small amount each time (a few degrees). It can take many iterations of the 8 bolts until the bolts are sufficiently tight. Do not overtighten when the bolt is difficult to tighten any further, it is tight enough. During this procedure, ensure the spanner or wrench does not come into contact with the window of the viewport.
8. Lower the laser/microscope assembly while watching the PyMS screen, the planchet should come into sharp focus on the GX camera as well as the Dynolite.

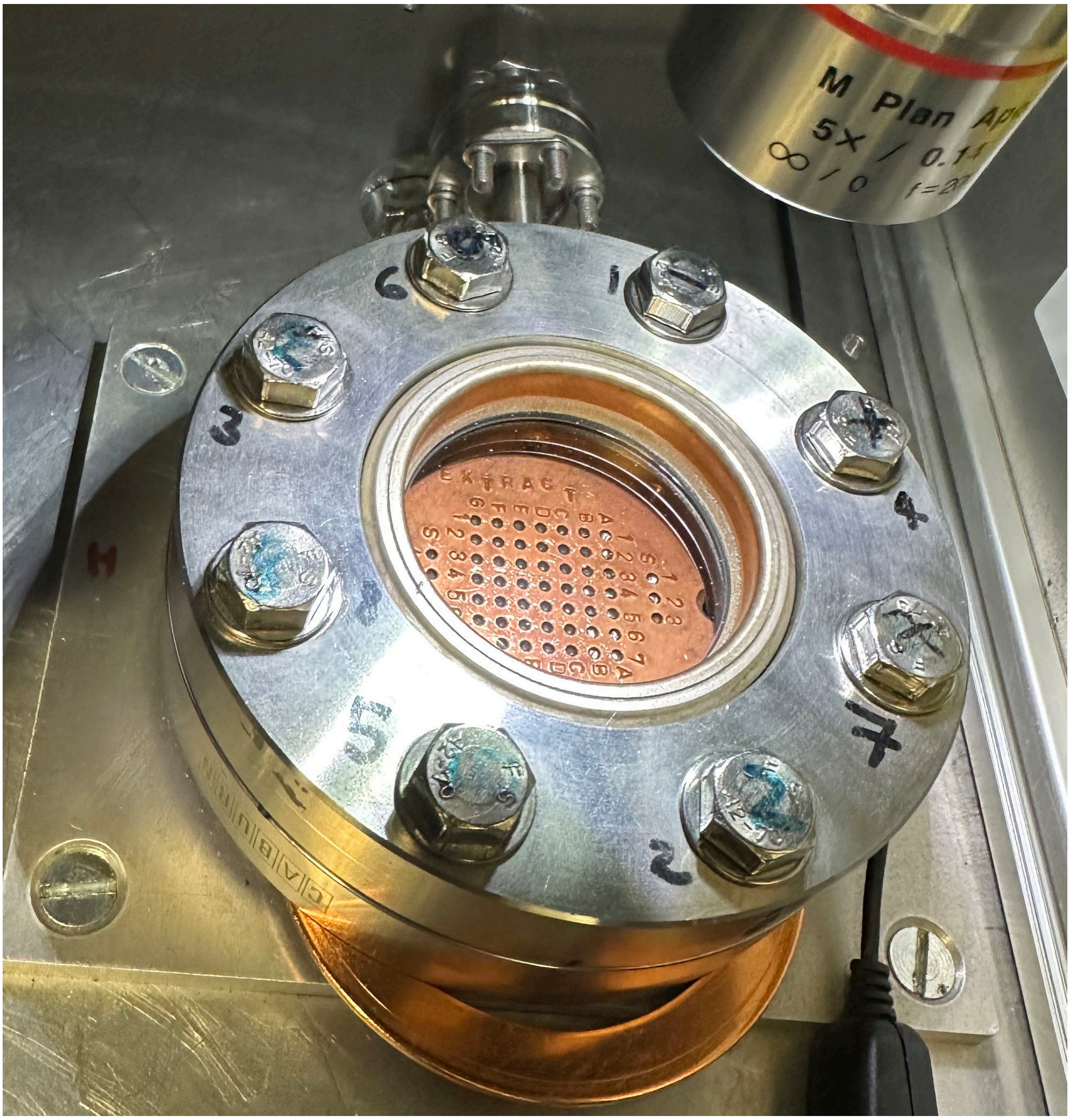


Figure 2: Planchet in Laser Chamber
Bolts are numbered order they should be tightened

9. The system in now ready to pump:
    a. Open the manual valve A in Figure 1, it will click as it opens!
    b. The pressure on the tank will increase and then rapidly drop to 10^{-3} mbar over the space of around 15 minutes.
    c. Close valve A (Figure 1)
    d. Open valve 13 and then valve 10 on the PyMS application. The Line pressure valve should increase to 10^{-5} mb but then decrease to 10^{-7} mbar over an hour.
    e. Over 24hrs the pressure should drop to the 10^{-8} mbar range if there are no leaks

Sample Unloading

1. Using the PyMS Control Panel, ensure Valves 8 (Volume A-B), 10 (Laser port), 11 (Hot Getter) and 13 (Turbo Pump) are closed (red). Damage to the mass analyser will occur if air at atmospheric pressure reaches volume A.
2. Raise the laser optics, Make sure that there are no cables being stretched or caught as the optics are raised.
3. Using a 13 mm spanner, undo the bolts on the laser port, taking care not to touch the viewport glass. The bolts are numbered, undo them in numerical order and do not remove the bolts from the viewport. When all bolts are undone, carefully lift the viewport, leaving the bolts in the viewport, and put it to one side. Do not place the viewport directly onto a surface, or the knife edge seal on the underside may be damaged. The viewport can be placed on its side, but ensure it will not fall or roll of the table! Remove the copper gasket (the viewport can then be placed on this gasket on the desk).
4. Screw the 4 mm bolt (usually stored in a petri dish in the laser box) into one of the threaded screw holes in the copper planchette (there are only two it will thread into). If you cannot find a bolt, a standard 4mm bolt will do, but clean it in acetone first to ensure there is no grease on the thread!
5. Carefully lift out the planchette with the bolt and place it in the petri dish. The bolt can be left in the planchette to allow easy manipulation without touching the copper.

PyMS Start Up Sequence

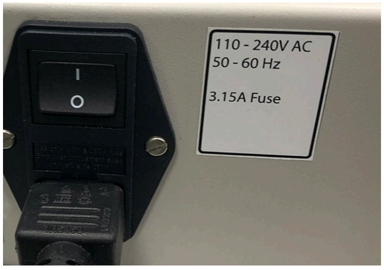
Use this process each time the pc is rebooted (or following a protracted power cut beyond the limits of the UPS system) to ensure the software is running and communicating correctly.

Switch on the Hiden Mass Spectrometer

Switch on the Hiden controller using the power switch at the front of the controller

Switch on the X-Y and Valve Controller

Switch on the X-Y and Valve Controller at the power switch



Start the DynoCapture Software

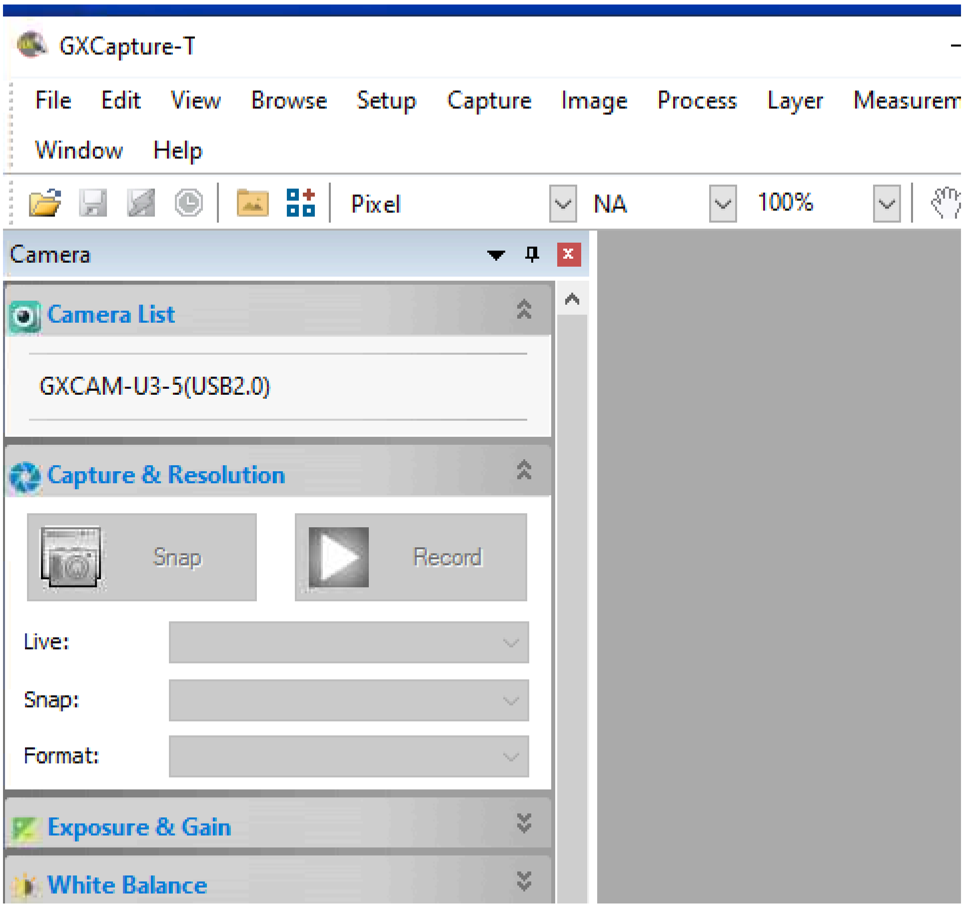
Start the DinoCapture 2 software from the start menu

This will start the software to view the dynolite positioning camera.

Start the GXCapture-T Software

Start the GXCapture-T software from the start menu

Once the software has started select the Camera GXCAM-U3-5(USB2.0) to view images from the microscope camera



Set the **zoom** to **100**%

Start the Mass Spectrometer Measure application

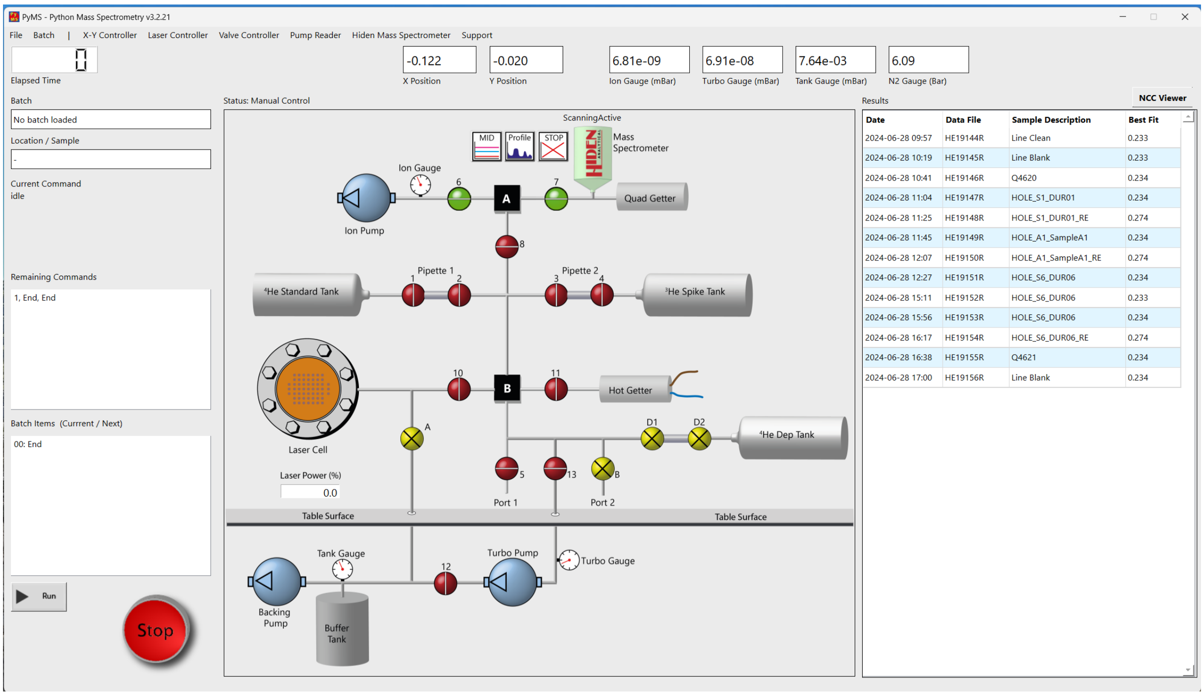
Start the MASsoft 10 software from the start menu

This will start the Mass Spectrometer software and connect it to the mass spectrometer.

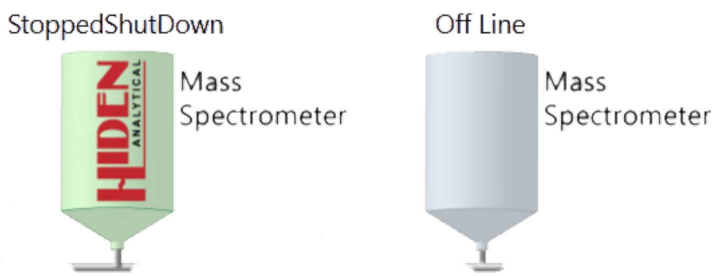
Start the PyMS application

Start the PyMS application from the start menu

The main screen will start and after 2 seconds will display the data from the vacuum pumps.



If the MASsoft 10 is running the Mass spectrometer will be green and the status will be shown above the image. If the MS offline it will appear as grey and the words Off Line will show



PyMS running a helium extraction

Process to run a helium extraction on a set of samples loaded onto a planchet.

Follow the Startup Sequence

Ensure you have followed the "**PyMS Start Up Sequence**" and the MASoft 10, DynoCapture, GXCapture and PyMS applications are running

Populate the planchet

Using the Olympus microscope, load samples into the planchet starting with a Durango in S1, (additional Durangos can be loaded into S2 & S3). Load samples starting at A1 through to G7. Add another Durango in position S4. Each pit in the planchet is hemispherical, try and load each platinum tube in the centre of the location pit in the planchet.

If the Pt tube is to one side, then it will not present the ideal aspect to the incoming laser beam and this will affect heating and the colour shown on the cameras. Note-down each sample on a planchet sheet so the sample identifiers can be entered into the PyMS system once the planchet has been loaded.

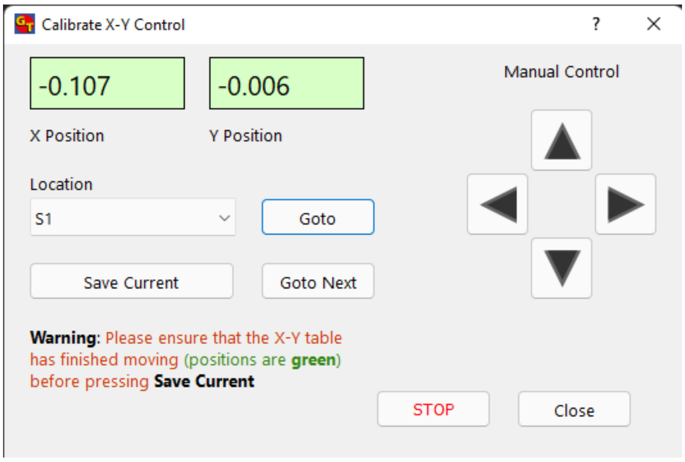
Load the planchet and start to pump down the line.

Follow the "**Sample loading**" process in this document to load the planchet into the laser chamber and pump the chamber down to a hard vacuum (<5 x 10^{-8} mbar). You do not need to wait for the vacuum to be fully pumped to run the next step.

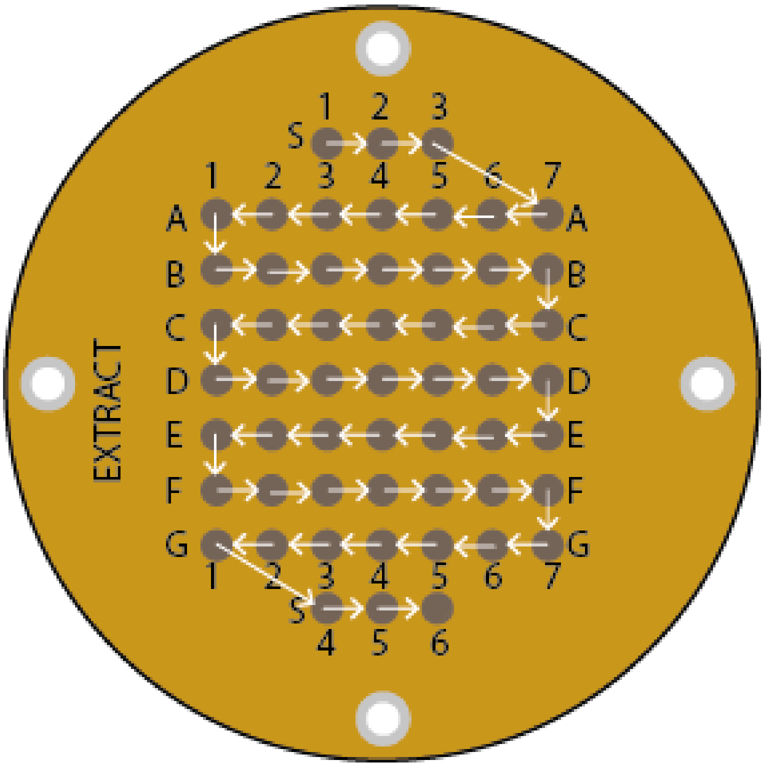
Calibrate the planchet

Open the menu item “X-Y Controller → Calibrate X-Y Stage”

The X-Y Calibrate screen will open.



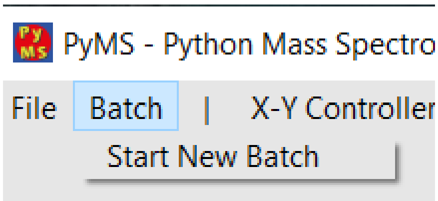
Press “**Goto Next**” the stage will move to the S1 location. Check the alignment of the microscope camera, if it needs adjusting use the manual control arrow buttons to position the Pt tube in the centre of the image, then press the “**Save Current**” button. Press “**Goto Next**” to move to the next spot. The stage will move in the following order:



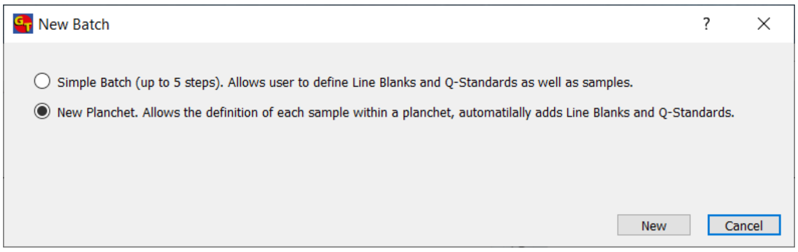
Once all the samples have been calibrated the screen can be closed-down.

Create a new batch on the PyMS application

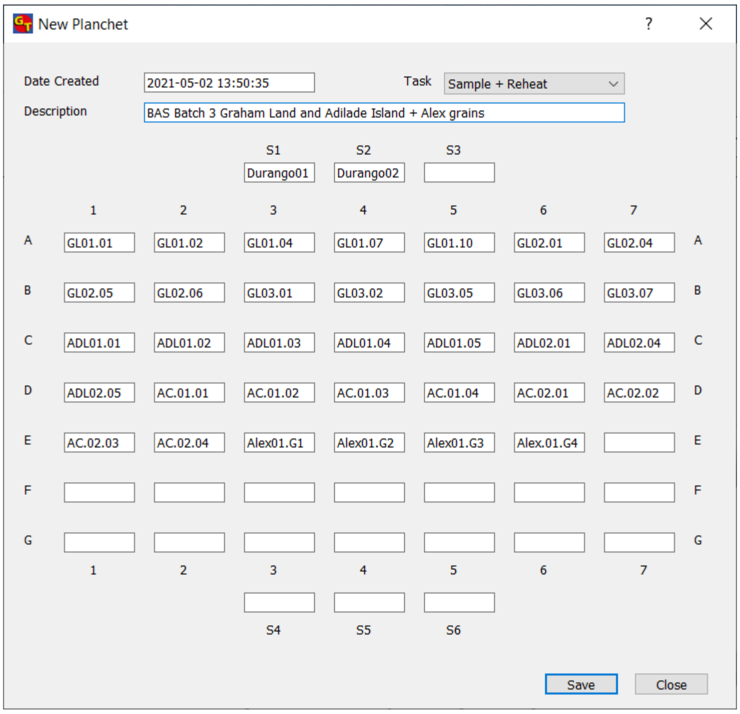
On the PyMS application, choose Batch → Start New Batch



Choose New Planchet



The new planchet form will load



The form will automatically add in the date and time the batch was created.

Give the batch a description, this will be reflected in the directory name that the final data is stored in.

Enter the grain identifier in each planchet cell that has a Pt tube/sample. Empty cells will be skipped when the system is running.

Select the task type e.g. **Apatite + Reheat**

Press **Save** to save the planchet.

The PyMS system will automatically add a line clean, line blank and Q standards at the start and end of the run and a 2-hour pump sequence half-way through the run. At the end of the run it will close the valves ready for the vacuum chamber to be unloaded.

Wait until the vacuum has pumped down.

Once the vacuum has fallen below ~ 5 x 10^{-8} mbar the system is ready to outgas the samples.

Check that there is sufficient nitrogen in the tank

The vacuum valves are worked by a N_2 pressure of 100psi, if the pressure is below 100psi then the valves will not open and correctly.

Ensure that the line pressure gauge (left) is reading 110 psi and the tank gauge (right) is above 200psi so there is sufficient capacity to complete the run.

Switch on and arm the Laser

Switch in the laser via the power switch at the back

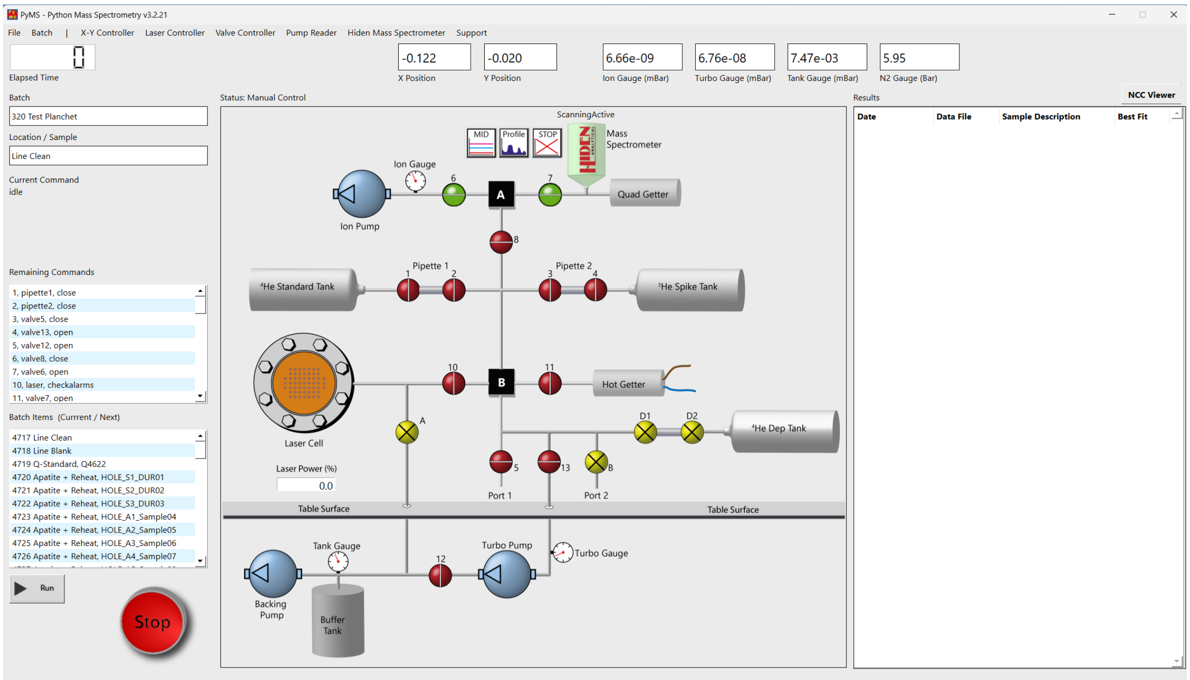
Wait ten seconds and then turn the **LASER ON** key one position clockwise

Wait one second and the turn the key clockwise to the second position

Press the **ENABLE** button

If the process is correct the Light on the Enable button will go yellow.

Run the analysis

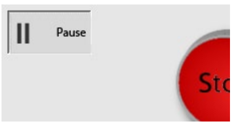


The main form will show with the batch items (grains, line blanks and standards) displayed on the bottom left list. The top left list shows the commands that are remaining on the current analysis cycle. The right-hand side lists the completed samples and their best-fit values.

To start the analysis press ►**Run**

The Elapsed time clock will start running and a message will appear with an estimated completion time.

To pause at any time, press the **Pause** button, the software will run to the end of the current task and then wait. Pressing ►**Run** again will restart on the next sample



The **STOP** button is an emergency stop and will close all valves and switch off the laser if it is activated.

Once the process has finished it will return the laser to the unload position and set the valves ready to unload the planchet.

Switch off the laser

Press the **Enable** button

Turn the **Laser On** key two positions anti-clockwise

Switch off the laser

Unload the planchet

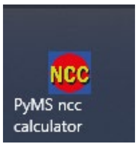
Follow the “**Sample Unloading**” process in this document.

PyMS calculating ncc

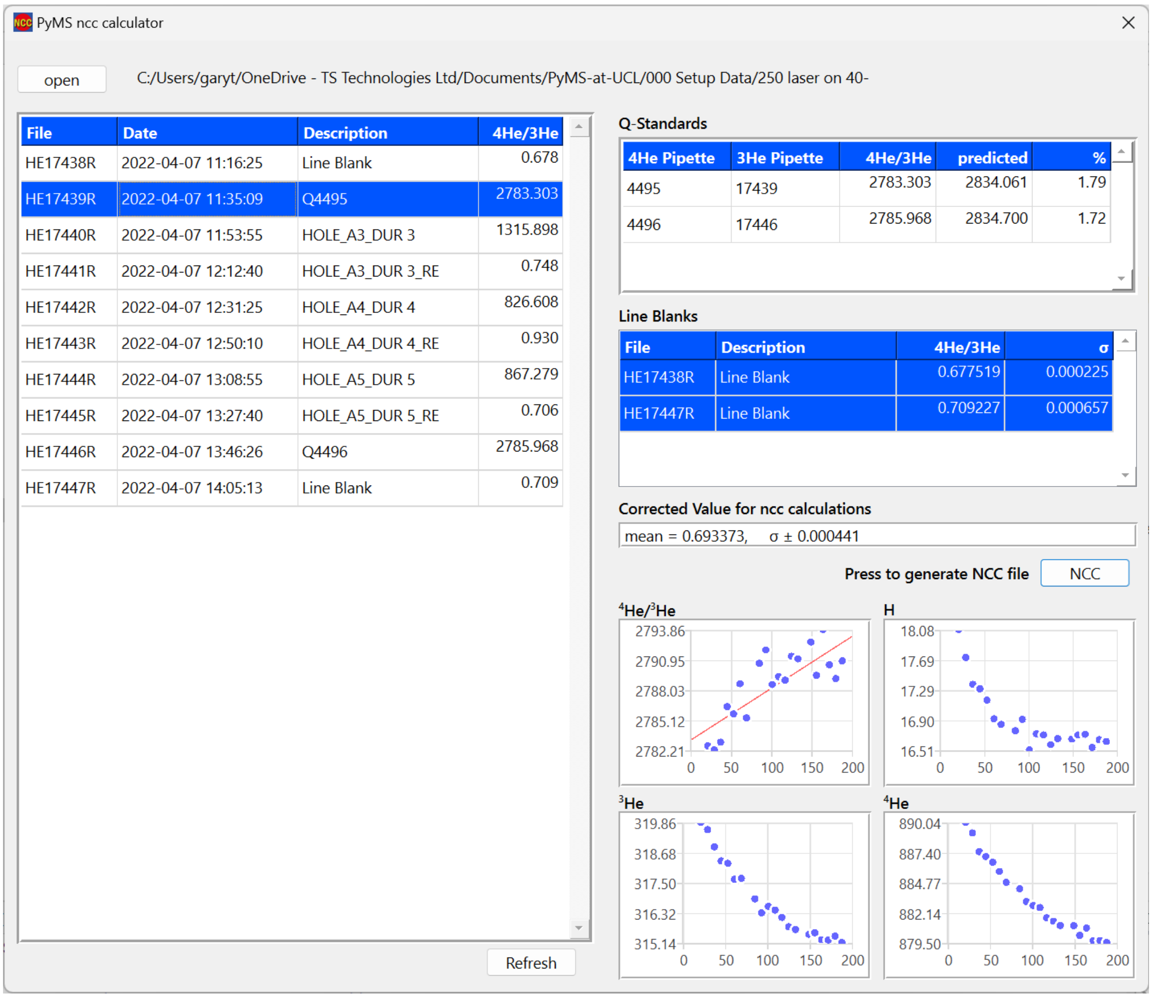
The PyMS application will automatically generate an ncc file in the data folder for your batch called “`PyMS_ncc.csv`”. The file will be based on all of the line blanks and Q-Standards run in the batch.

However if there are issues with a line blank, the ncc can be recalculated using the PyMS ncc calculator. It is an application that will read all the data files in a PyMS batch and allow a researcher to recalculate the ncc based on a selection of blanks as well as a graphical view of any of the data sets.

To open the PyMS ncc calculator click on the icon:



Once opened it will revert to the last folder opened.



Pressing the “open” button (top left) will allow you to select a PyMS folder to view.

The app will load all of the helium files into the left list along with the data the data was saved, the file description and the $^{4}He/^{3}He$ value.

The top right list shows any Q-Standards run, the $^{4}He/^{3}He$ value, the predicted value and the percentage difference.

The lower right list shows all of the line blanks run. By default they are all selected but the researcher can select and deselect any values so if there was an anomalous value it can be excluded from the line blank correction. Below the list is the line blank correction and line blank correction error that will be used for a ncc calculation. The “NCC” button will calculate the ^{4}He ncc for each data file and save it to a file “PyMS_ncc.csv” that can be used for helium age calculations.

Clicking on any file in the left list will generate 4 graphs in the bottom right corner of the app:

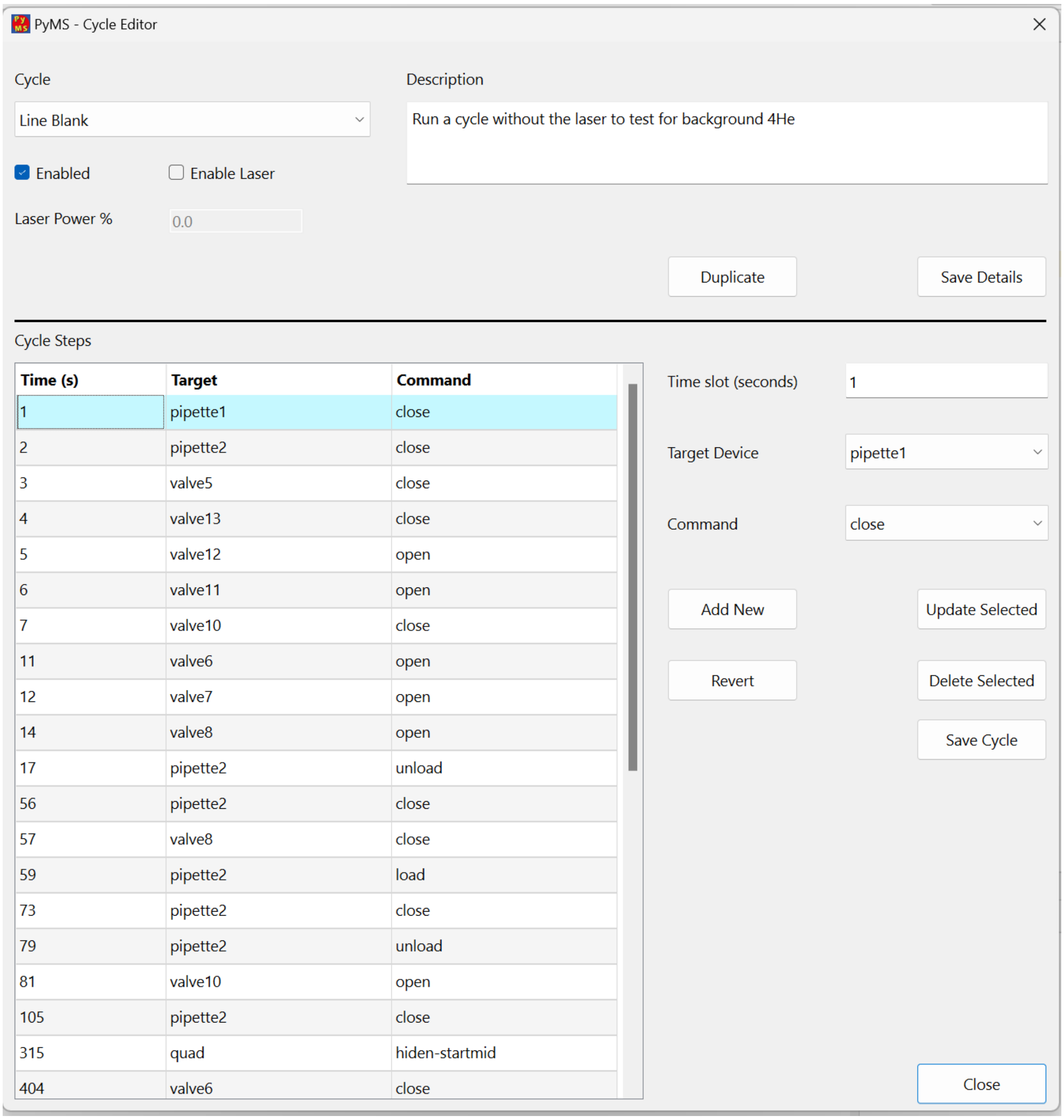
$^{4}He/^{3}He$	This is the graph of the m4 values divided by the HD corrected m3 values, a red regression best-fit line is added.
H	The uncorrected m1 values
^{3}He	The m3 values corrected for HD
^{4}He	The uncorrected m4 values

Editing PyMS Cycles

This is used for changing the run sequence or times.

Start the Cycle editor

The Cycle editor will load.



The **Cycle** dropdown is used to change the cycle type.

The **Enabled** checkbox will allow it to be used on a simple sample

The **Enable Laser** is used for sample analysis cycles and allows the laser power to be set.

Duplicate will create a duplicate of the cycle and prompt you for a new name. This would used to clone a sample cycle (e.g. Apatite + Reheat) to allow for changes in timings or laser power for a different mineral.

Save Details will save the cycle name, description, laser power and enabled status.

The list shows the current tasks in the cycle. Highlighting a line on the list will allow that line to be edited.

On the right hand side, enter the time slot (seconds), choose a target and a command for that target.

Update will update the current line

Add New will add the command as a new item. (Note you cannot have two commands with the same timeslot)

Delete will remove the item

Save Cycle will save the cycle to the database

Revert will cancel any changes to the list and reload it from the database

Close will exit this application

Detail on ncc values

Once a batch has been completed the PyMS application will automatically calculate the ncc values for each Q-standard, sample and reheat. The calculation involves measuring the ratio of ^{4}He extracted from a sample against a known quantity of ^{3}He spike.

The output from the Hiden Mass Spectrometer is recorded in a tab delimited text file during the PyMS cycles. The masses are recorded at 10 second intervals and 20 sets of values are stored. Recording starts at 20 seconds after the valve 8 is opened to allow the gas mixture from the section B to equilibrate and arrive at the Mass spectrometer. The file name is in the format HEnnnnnR where the number "nnnnn" is the total of spike pipettes used so far. The elements detected and atomic masses are in Table 1.

Atomic Mass	Elements detected
1	H
3	$^{3}He^{+}$, HD^{+}
4	$^{4}He^{+}$, $^{12}C^{3+}$
40	^{40}Ar

Table 1: Atomic masses and detected elements

The m3 signal must be corrected for interference due to HD. The assumption is that the ratio of HD^{+} to H is 0.01 so the 3He value is calculated as m3 – (m1 x 0.01). The interference of $^{12}C^{3+}$ on the m4 value is assumed to be negligible so the ^{4}He value is assumed to be the m4 value. The argon value is recorded and can be used for diagnostic purposes to indicate air leaks.

ncc calculation for samples

The $^{4}He/^{3}He$ ratio is calculated for each time period, and a regression best fit for time "zero" (when valve 8 was opened) is calculated. The zero intercept and standard error are recorded for a Line Blank ($B \pm \sigma_B$), Q-Standard ($Q \pm \sigma_Q$) and Sample ($S \pm \sigma_S$). The mean of the Line Blanks and Line Blank errors are calculated to provide a baseline value and then deleted from the Q-Standards and Samples to give the blank corrected ratios $Q_c \pm \sigma Q_c$ (Equation 1) and $S_c \pm \sigma S_c$ (Equation 2).

Equation 1

$$Q_C \pm \sigma_{Q_C} = Q - B \pm \sqrt{\sigma_Q{}^2 + \sigma_B{}^2}$$

Equation 2

$$S_C \pm \sigma_{S_C} = S - B \pm \sqrt{\sigma_S{}^2 + \sigma_B{}^2}$$

The Q-Standard is used as a way of calibrating the output from the mass spectrometer with a known value of ^{4}He mixed with the ^{3}He spike. Once the Q-Standard value in ncc (nano cubic centimetres) is known then the ratio of ^{4}He to ^{3}He for subsequent samples can also be calculated.

When the Q tank was originally filled a pipette contained 10.23 ± 0.07 ncc of ^{4}He at STP. The tank was measured at 5800 ± 6 cc (T) and the pipette is 0.3222 ± 0.0012 cc ($Q_{pipette}$ and $\sigma_{Qpipette}$).

Each time a Q-Standard is run, it depletes the ^{4}He tank by the ratio of the volumes of the tank and the tank + the pipette, thus reducing the pressure in the ^{4}He tank. The rate of depletion of the tank, the Depletion Factor (DF) can be calculated using Equation 3 from the data supplied with the tank (Table 2).

^4He Standard Tank		
Tank	5800 cc	± 6cc
Pipette	0.3222cc	±0.0012 cc
Shot 1	10.23 ncc	± 0.07 ncc

Table 2: ^4He Standard Tank Data

Equation 3

$$DF = \frac{T}{T + Q_{pipette}}$$

$$= \frac{5800}{5800 + 0.3222}$$

$$= 0.99994445$$

The depletion of the ^4He tank may deviate from the theoretical value above so a depletion tank is included in the Helium line. The depletion tank is only used every 1000 Q-Standard shots and is assumed to be at a constant pressure. By comparing the quantity of ^4He in a Q-standard with a sample using ^4He from the depletion tank, a depletion factor value can be found At the last calculation in 2017 by J Schwanethal it was found to be 0.9999526 with an error (σ_{DF}) of ± 0.0000004. By knowing the quantity of ^4He in shot 1(from Table 2) and the number of shots used so far (n) the quantity of ^4He in ncc can be calculated by Equation 4.

Equation 4

$$Qncc = 10.23 \times DF^n$$

The ratio of Q_c to Q_{ncc} is equal to the ratio of S_c to S_{ncc} where S_{ncc} is the amount of ^4He outgassed from the sample and σ_{Sncc} is the ± error (Equation 5).

Equation 5

$$\frac{Q_c}{Q_{ncc}} = \frac{S_c}{S_{ncc}}$$

Sample ncc:

Equation 6

$$S_{ncc} = 10.23 \times DF^n \times \frac{S_c}{Q_c}$$

Error in sample ncc:

Equation 7

$$\sigma_{Sncc} = S_{ncc} \times \sqrt{\left(\frac{\sigma_{Qpipette}}{Q_{ncc}}\right)^2 + \left(\frac{n}{\sigma_{DF}}\right)^2}$$

All of the constants for the calculations for both the PyMS application and the Pyms ncc calculator are stored in the settings.json file so they can be changed if needed without having to edit the application code:

Constant	Current Value	Description
HD_H	0.01	the ratio of HD^+ to H,used to correct the M3 value for ^3He
q_dep_factor	0.9999526	Depletion factor of the ^4He Q tank when a Q-standard is run (calculated in 2017 by J Schwanethal)
q_depletion_err	0.0000004	^3He Q tank depletion error
q_pipette_ncc	10.23	Pipette quantity of ^4He in first shot from the Q tank
q_pipette_err	0.07	^3He pipette error
s_dep_factor	0.99996107	Depletion factor of the ^3He spike tank when a sample, line blank or Q-standard is run (calculated in 2017 by J Schwanethal)
s_offset	231	Offset on the number of spikes run to account for increments in the HE number
s_pipette_ncc	5.7	Pipette quantity of ^3He in first shot from the spike tank

Table 3: Constants used for ncc calculations

UCL Helium Line Depletion Factor Calculation

^3He tank calibration

The ^3He tank is depleted every time a line blank, Q-standard and sample is run. Whilst it is not required for calculation of the sample ^4He ncc values it is something that needs to be monitored. The PyMS ncc calculator application calculates a predicted Q ratio from the Q_{ncc} value calculated from above based on the depletion factor of the ^3He tank (DF_{3He}), the Number of Spike shots used so far (N) and the initial pipette capacity of 5.7 ncc when the instrument was first commissioned. There was a rounding up of the spike shot value in 2003 by 231 shots (s_{offset}) which also needs to be taken into account of. The value of ^3He ncc in the Q-standard can be calculated by Equation 8.

^4He Spike Tank		
Tank	5800 cc	± 6cc
Pipette	0.2258 cc	±0.0012 cc
Shot 1	5.7 ncc	

Table 4: ^3He Spike Tank Data

Equation 8

$$^3He_{\ ncc}\ \ 5.7\ X\ DF_{3He}{}^{N-S_{offset}}$$

The predicted $^4He/^3He$ value is then calculated in Equation 9.

Equation 9

$$Q_{pred}\ = 1000\ \times\ \frac{Q_{ncc}}{^3He_{ncc}}$$

The PyMS ncc calculator shows the Q_{pred} as a percentage of the measured Q value. If needed the Q number in the PyMS application can be adjusted if it more than a few percent out after a depletion test has been run to recalculate and re-adjust the depletion factor. The depletion factor is entered into the settings.json file in the PyMS application.

Depletion Test Instructions

Aim to do a depletion run every 1000 Q shots

1. Ensure valves D1 and D2 are closed
2. Clean up pipette volume between valve D1 and D2 by:
    a. Close valve 13, 10, and 11
    b. Open valve 6 and 8 on the PyMS software
    c. Open manual valve D1
    d. leave open for at least 30 seconds
    e. then manually close D1
3. Create a new short batch in PyMS
    - Line Clean
    - Line blank
    - Q-standard
    - Q-standard
    - Depletion Test
    - Q-standard
    - Q-Standard
4. Run the batch
    a. The line clean, blank and Q-Standards will run
    b. When the Depletion Test starts:
        i. Make sure D1 and D2 are closed
        ii. At the 8s mark you will be prompted to open D2 to fill the pipette
        iii. At the 28s mark you will be prompted to close D2 to seal the pipette
        iv. At the 75s mark you will be prompted to open D1 to unload the pipette
        v. At the 1030s mark you will be prompted to close D2 to seal the pipette
    c. The remaining Q-Standards will run followed by another line blank.
5. Use average of all 4 Q shots to get Q/DT ratio – value added to excel sheet called "Q-DT" plot to see if falls on line (expected depletion trend)

Hiden Residual Gas Analyser

Starting the Hiden

Switch on the Hiden controller using the power switch at the front of the controller

Wait for the startup sequence to complete and then Start the Mass Spectrometer Measure application on the PyMS PC.

Start the MASsoft 10 software from the start menu

This will start the Mass Spectrometer software and connect it to the mass spectrometer

Once the application has started it will show the status screen

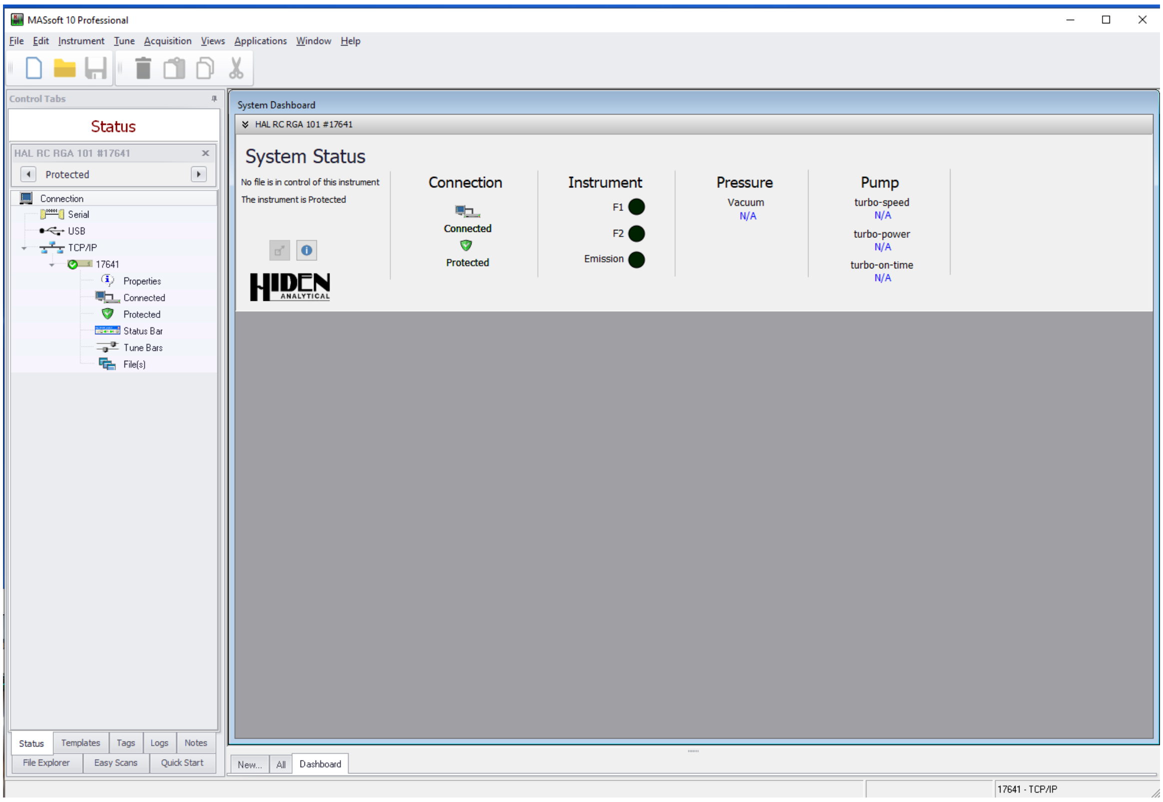
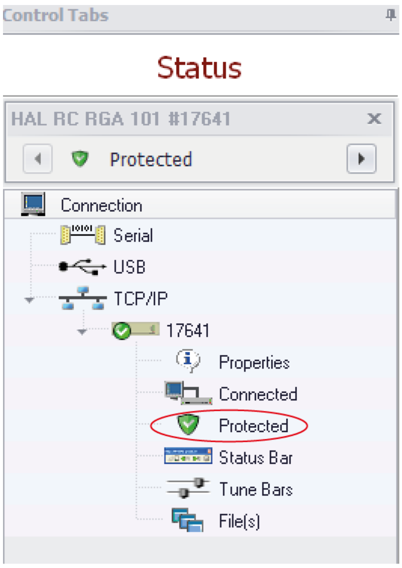


Figure 3: Hiden MAS 10 Software

From the status pane on the left of the screen it can be see that the instrument is in protected mode. In the Protected state the values of the Shutdown Settings are applied in the instrument and power supplies are disabled. The instrument boots up in the Protected state.

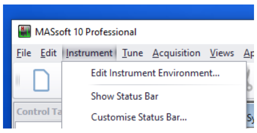


When an experiment is running it will show Running a scan

Editing the Hiden Default Settings

The Hiden has two sets of settings, a low power mode for when it is shut down and an RGA mode for when it is running. There is no ramp up of emission current, when an experiment starts it will switch to the running settings.

To edit the settings choose the “Edit Instrument Environment” option from the Instrument menu.



Initial shutdown settings

These were the initial settings when the Hiden was commissioned (Nov 2022)

Name	Value	Description (from Hiden software)
Multiplier	0 V	Controls the gain of the Secondary Electron Multiplier (SEM)
Focus	-90 V	Focusses the ions from the source into the quadrupole mass filter
Delta-m	0 %	Controls the peak width, especially at lower masses
Resolution	0 %	Controls the peak width, especially at higher masses
Cage	0 V	Attarcts Electrons into the source, determines the enerygy of the resultant ions
Electron-energy	70 V	Controls the energy of the electrons used to ionise and fragment neutral molecules in the source
Emission	20 µA	Controls the electron surrent produces by thermionic emission
Mass	5.5 amu	Controls the mass to charge ratio (m/z) of ions transmitted by the quadrupole mass filter
Mode-change-delay	0 ms	Allow voltages to settle when changing mode. Allows emission to establish in RGA mode

Initial running settings

These were the initial running settings when the Hiden was commissioned (Nov 2022)

Name	Value	Description (from Hiden software)
Multiplier	810 V	Controls the gain of the Secondary Electron Multiplier (SEM)
Focus	-90 V	Focusses the ions from the source into the quadrupole mass filter
Delta-m	0 %	Controls the peak width, especially at lower masses
Resolution	0 %	Controls the peak width, especially at higher masses
Cage	3.00 V	Attarcts Electrons into the source, determines the enerygy of the resultant ions
Electron-energy	70 V	Controls the energy of the electrons used to ionise and fragment neutral molecules in the source
Emission	200 µA	Controls the electron surrent produces by thermionic emission
Mass	5.5 amu	Controls the mass to charge ratio (m/z) of ions transmitted by the quadrupole mass filter
Mode-change-delay	1000 ms	Allow voltages to settle when changing mode. Allows emission to establish in RGA mode

Shutting down the Hiden

Stop any running experiments by pressing the red square

Close the experiment file

Exit the MAS application

Switch off the Controller using the green power switch on the front panel.

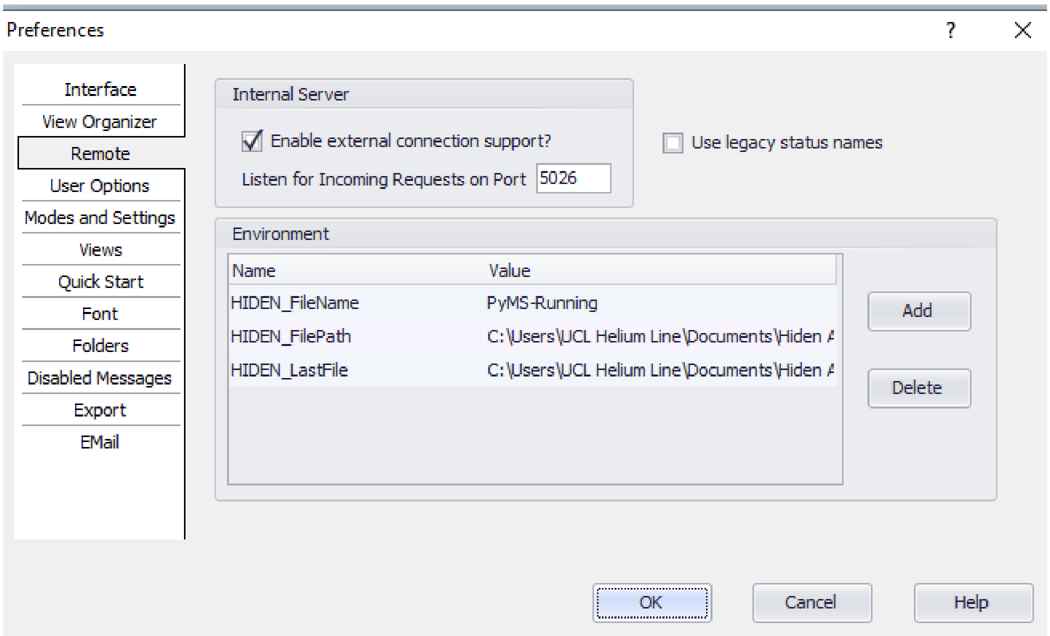
Hiden MASoft 10 Network Interface

The MASoft 10 application has the network access interface exposed to allow other applications to control it.

Host: localhost (127.0.0.1)
Port: 5026

It allows network socket connections with a limited set of commands (please see the Hiden MAS Sockets manual for full details)

Remote access was enabled in the preferences



Hiden Hardware Connection Diagram

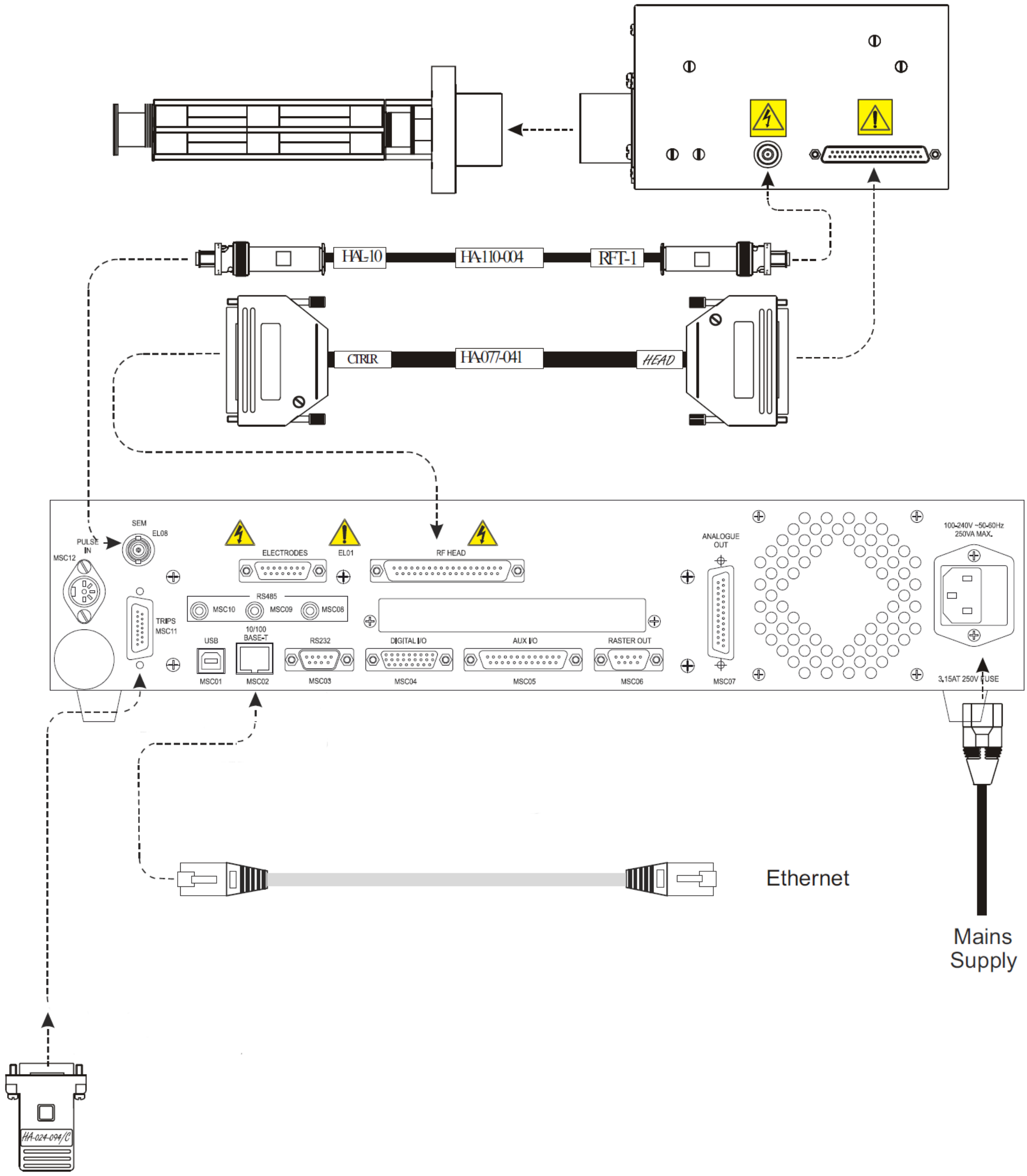


Figure 4: Hiden Connections

Hiden RGA - Configured Scans

Line Scan

The line scan is used to check the quality of the line after maintenance. It runs from 1 to 100 AMU with an increment of 0.1 AMU, it runs 20 full scans then shuts down. The output can be used to run further analysis

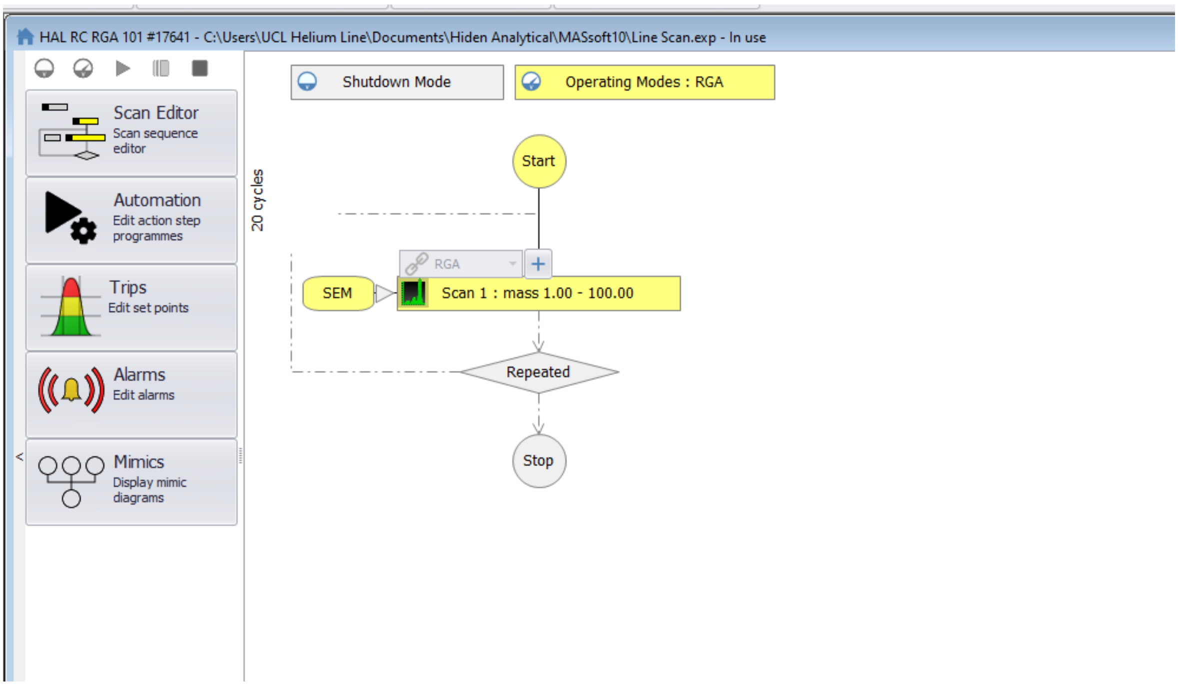


Figure 5: Line scan settings

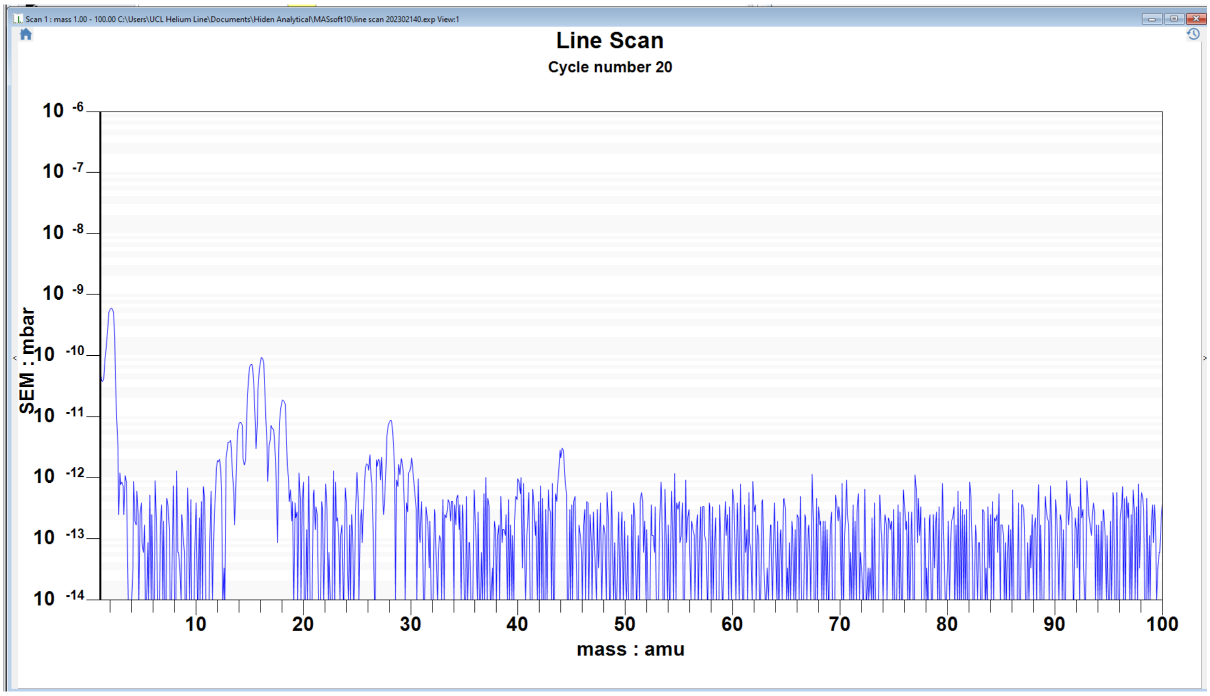


Figure 6: Line scan output

PyMS MID Scan

The Multiple Ion Detector scan is set to run continuously until it is stopped, either manually or programmatically via the network socket. It scans the following AMU:

- 1.0 H
- 3.0 ^{3}He
- 4.0 ^{4}He
- 40.0 ^{40}Ar

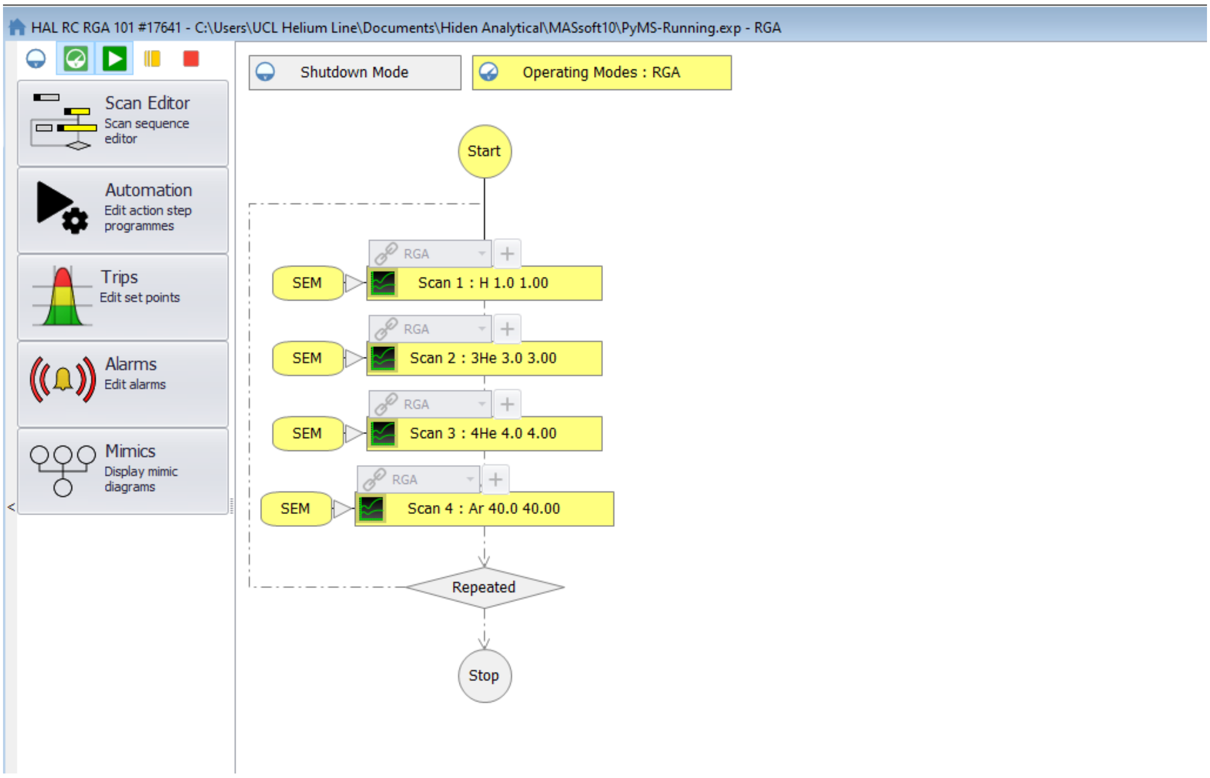


Figure 7: PyMS Mid Scan settings

Output is via a graph showing the trends over the past 30 minutes and a table showing the last values.

The expected values are:

- H ~5.0×10^{-11}
- 3He ~1.0×10^{-12}
- 4He ~4.0×10^{-14}
- 40Ar ~ 5.0×10^{-13}

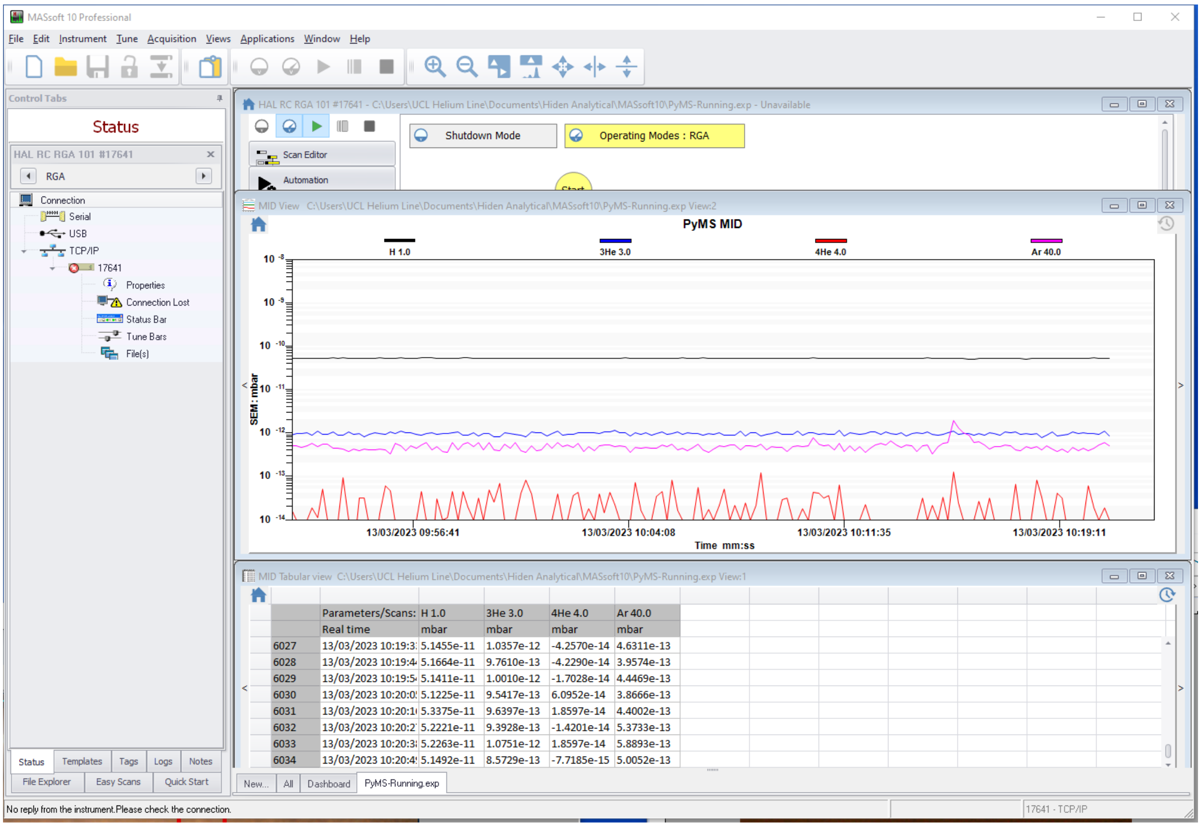


Figure 8: PyMS Mid Scan Output

PyMS Profile Scan

The profile scan scans from 1 to 50 AMU with an increment of 0.25 AMU, it runs 5 full scans then shuts down. The PyMS application calls the profile scan after a MID scan.

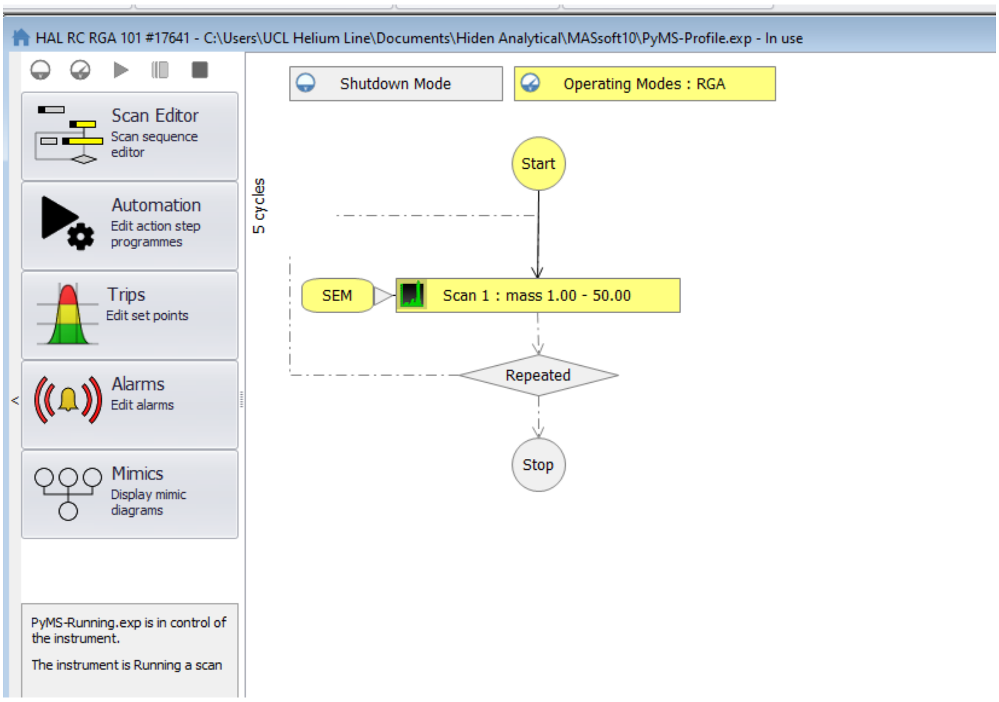


Figure 9: PyMS Profile scan settings

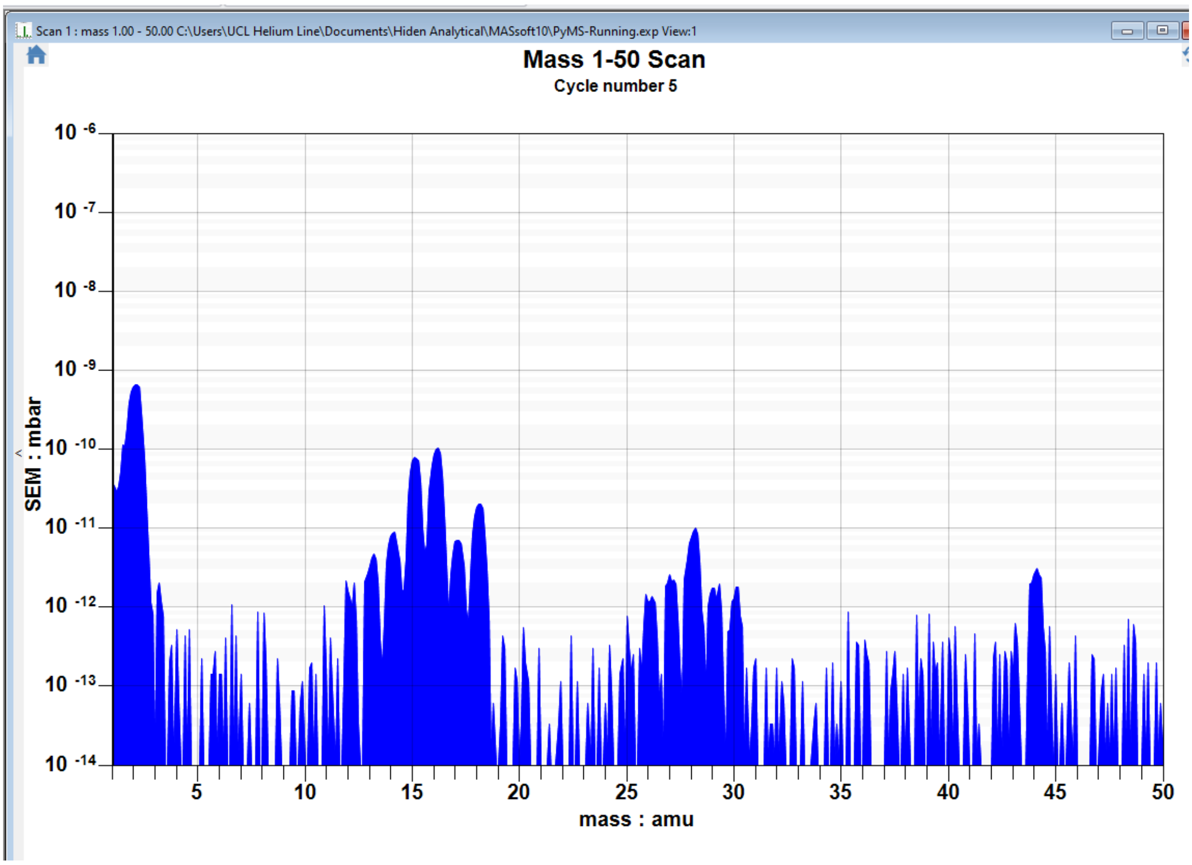


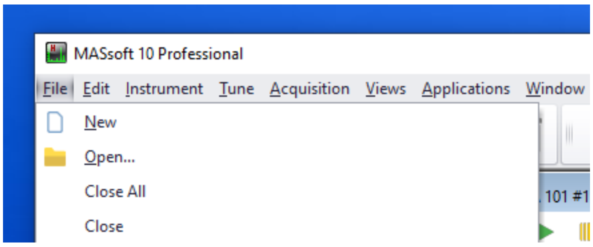
Figure 10: PyMS Profile Scan output

Manually running scans on the Hiden RGA

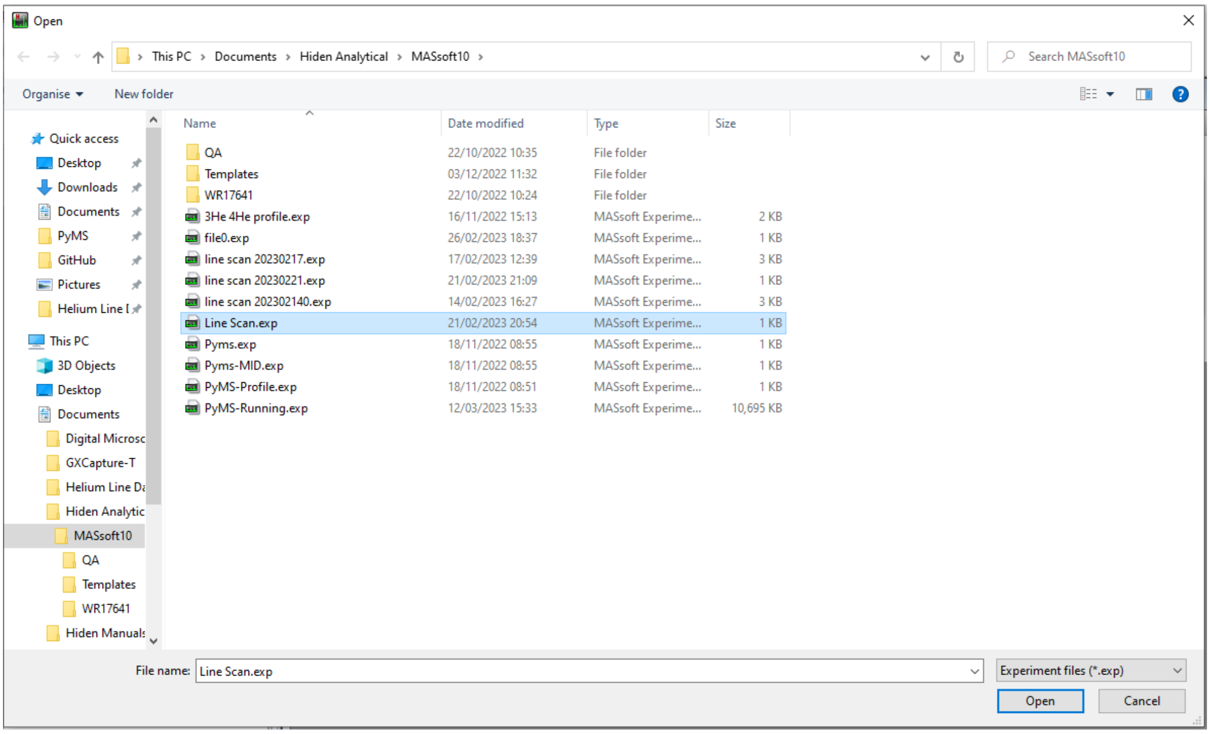
To run a Line scan

On the MASoft 10 Professional application:

“File” – “Open”

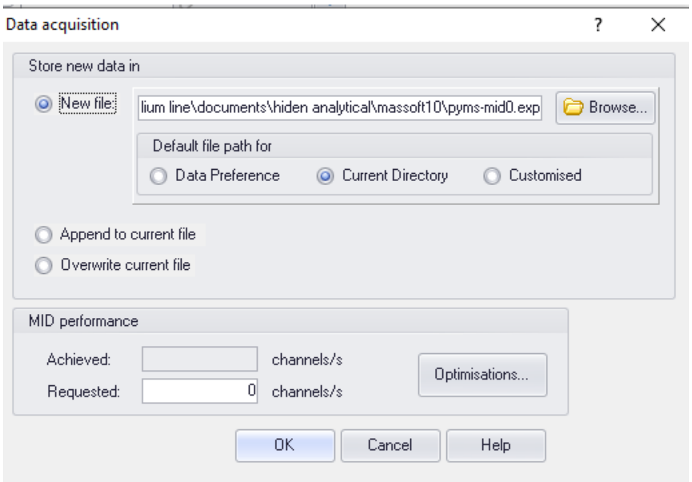


Choose the `Line Scan.exp` file and click open

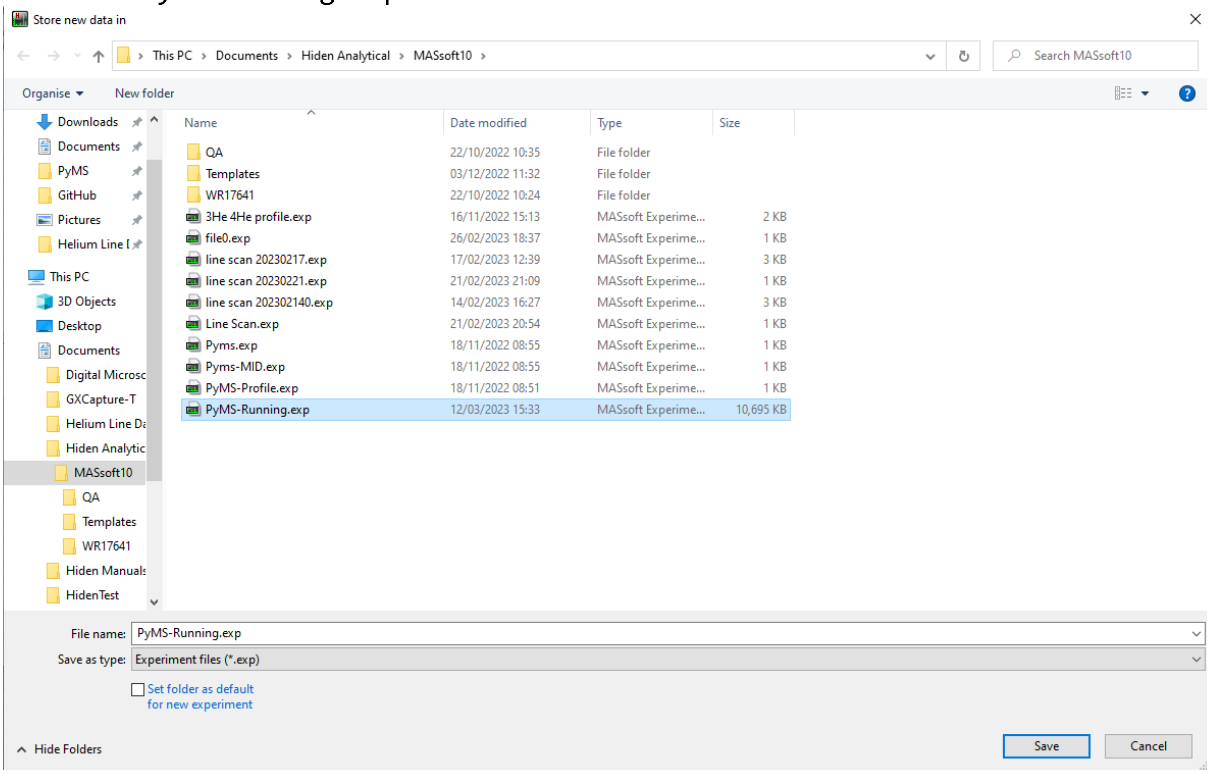


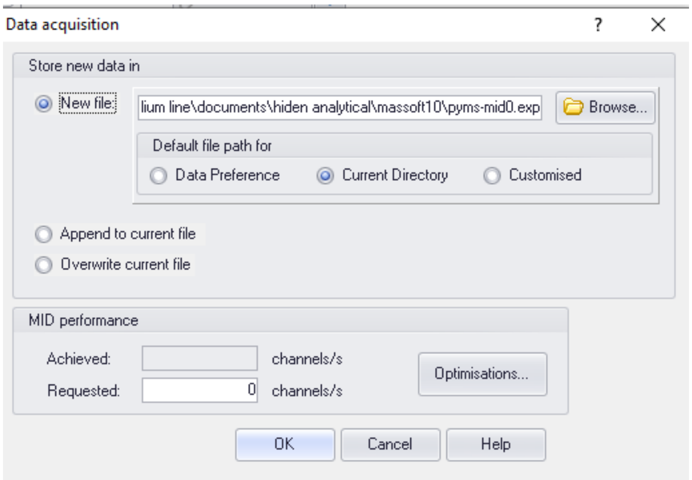
To start the scan click on the green “run” button at the top of the screen

A data acquisition box will appear, choose the “browse” button



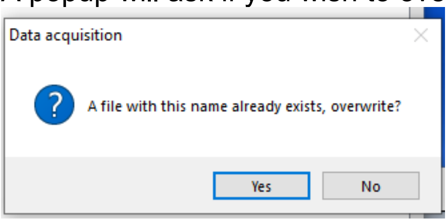
Select the `Pyms-Running.exp` file name and click “save”





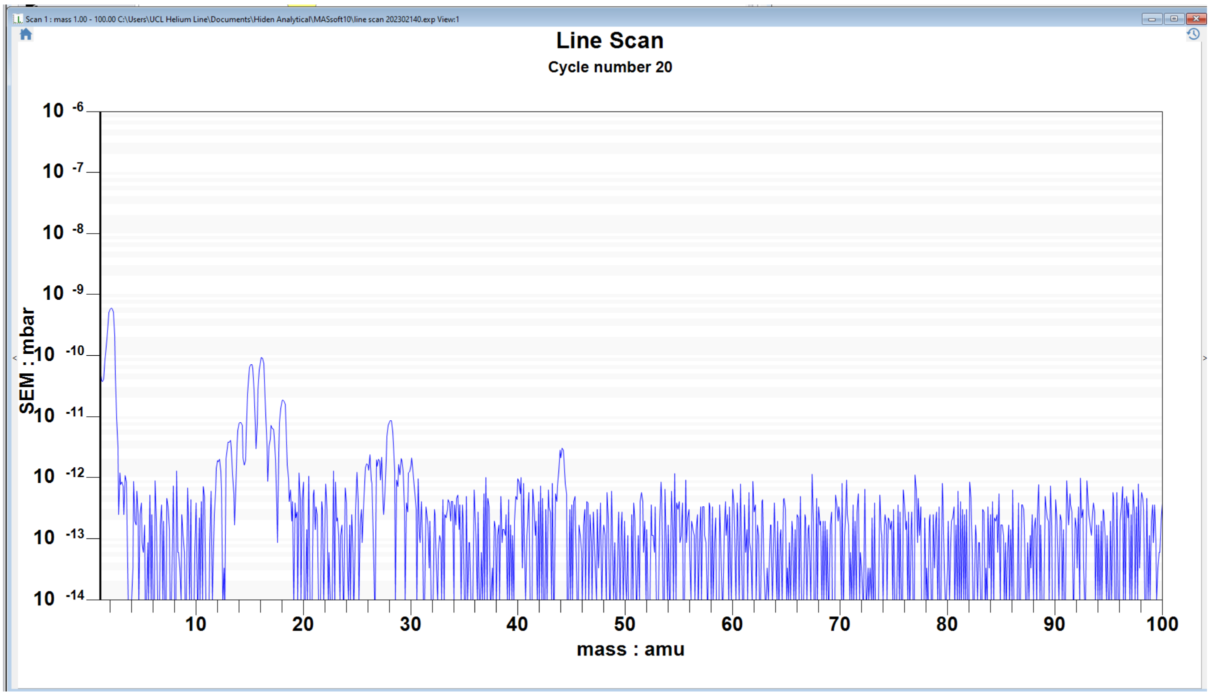
Press “OK”

A popup will ask if you wish to overwrite the file

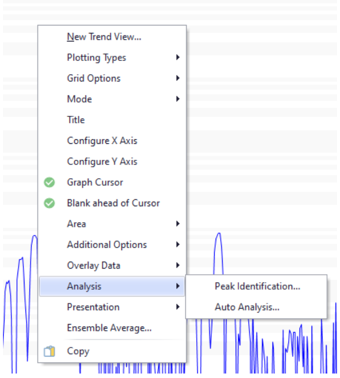


Choose “Yes”

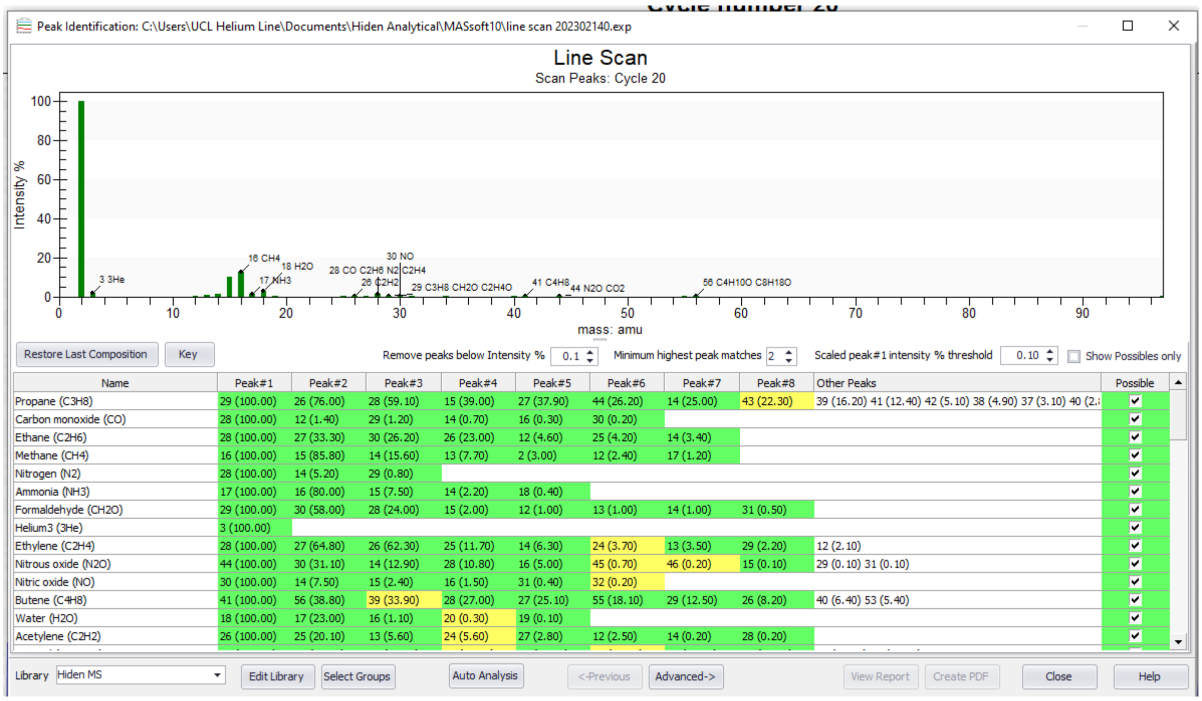
The Scan will run 20 cycles and then stop.



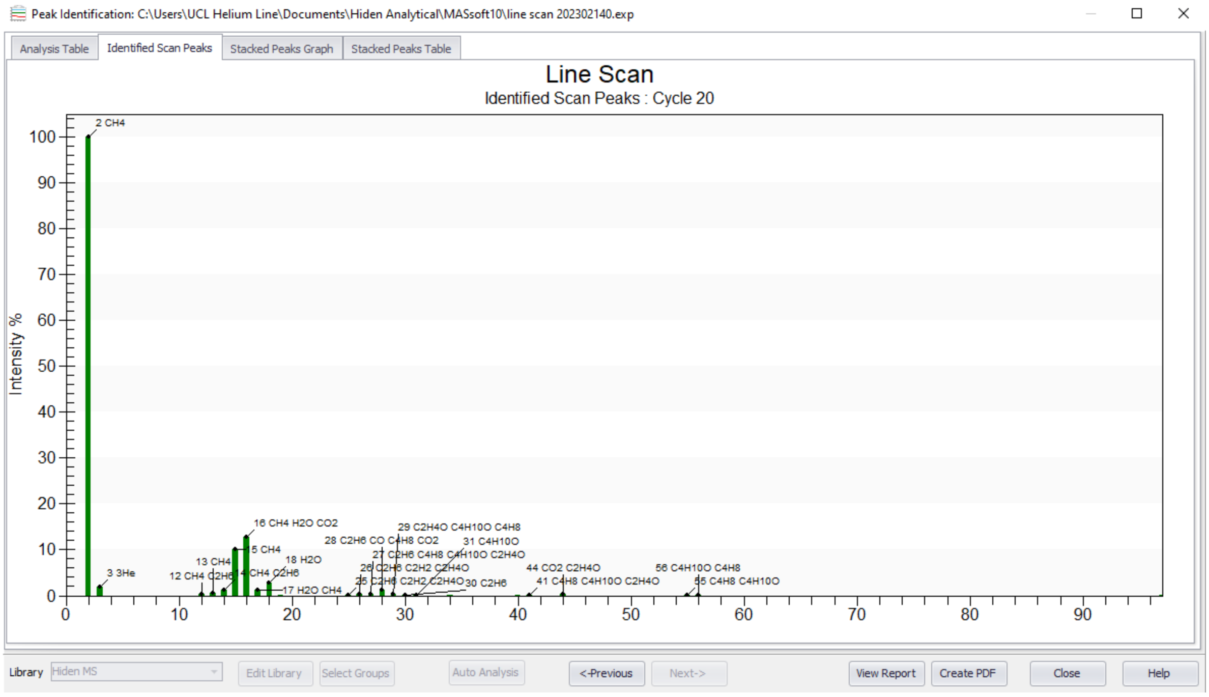
Once the 20 scans complete, to interpret the graph, right click on the graph and use Analysis



Peak Identification



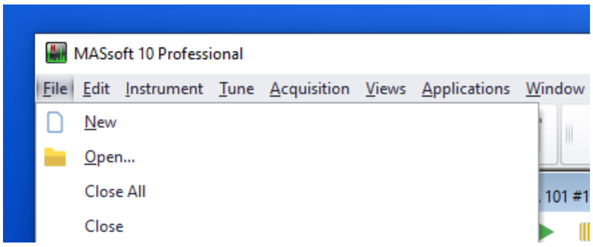
Auto Analysis



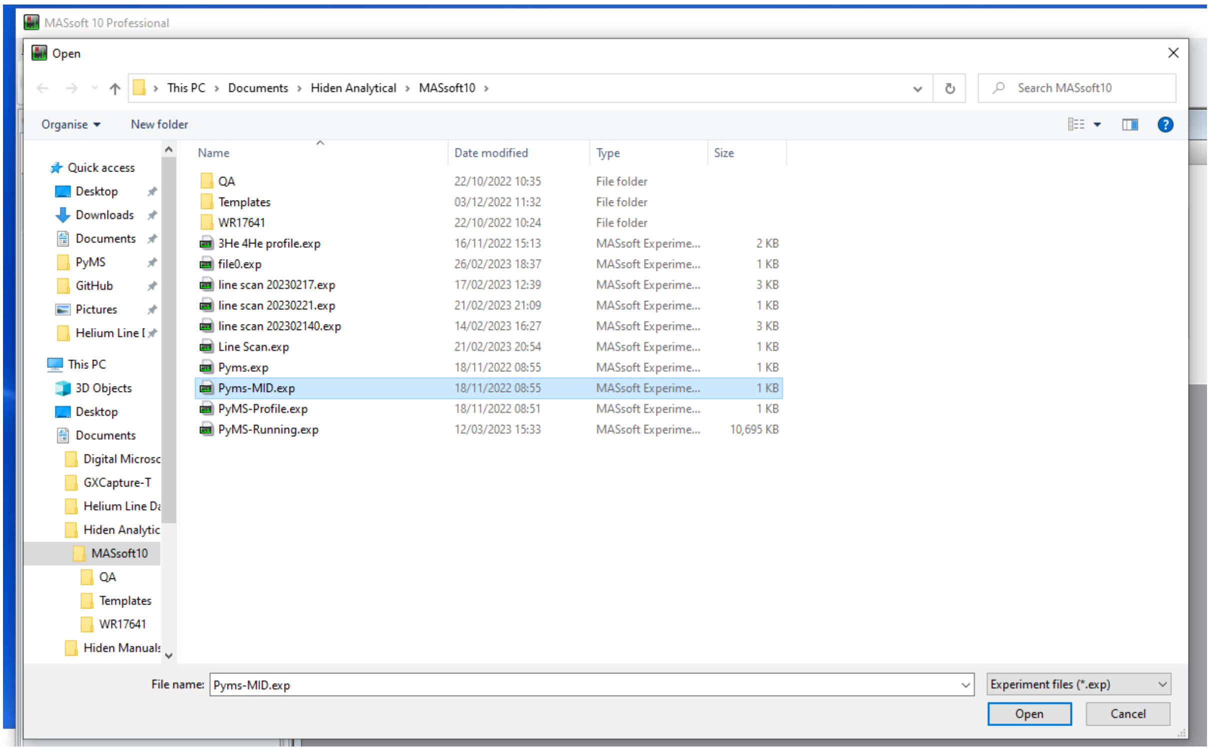
To run a MID scan

On the MASoft 10 Professional application:

“File” – “Open”

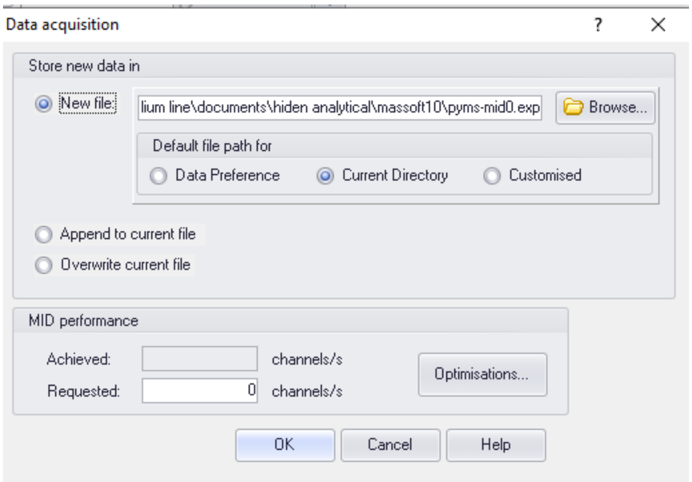


Choose the `Pyms-MID.exp` file and click open

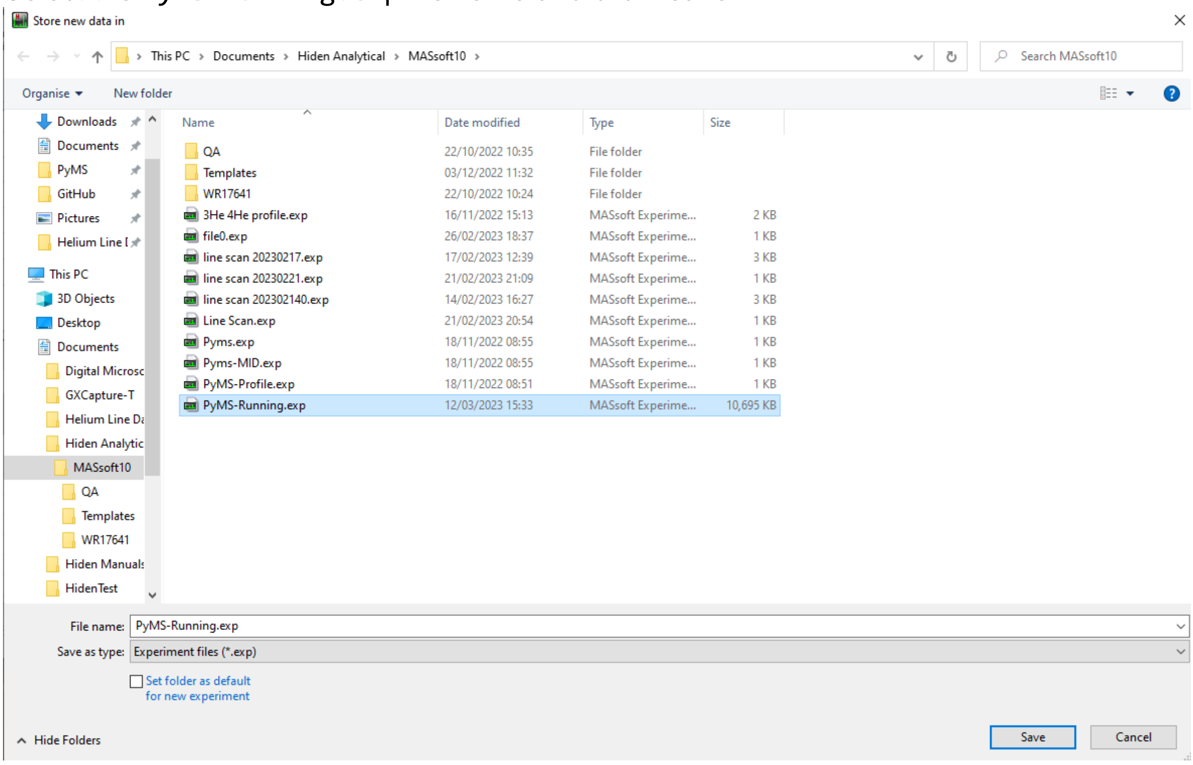


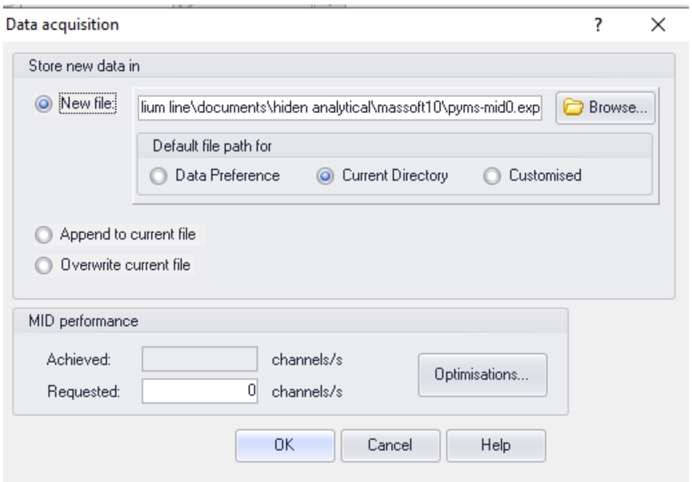
To start the scan click on the green “run” button at the top of the screen

A data acquisition box will appear, choose the “browse” button



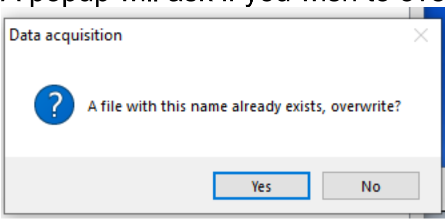
Select the `Pyms-Running.exp` file name and click “save”





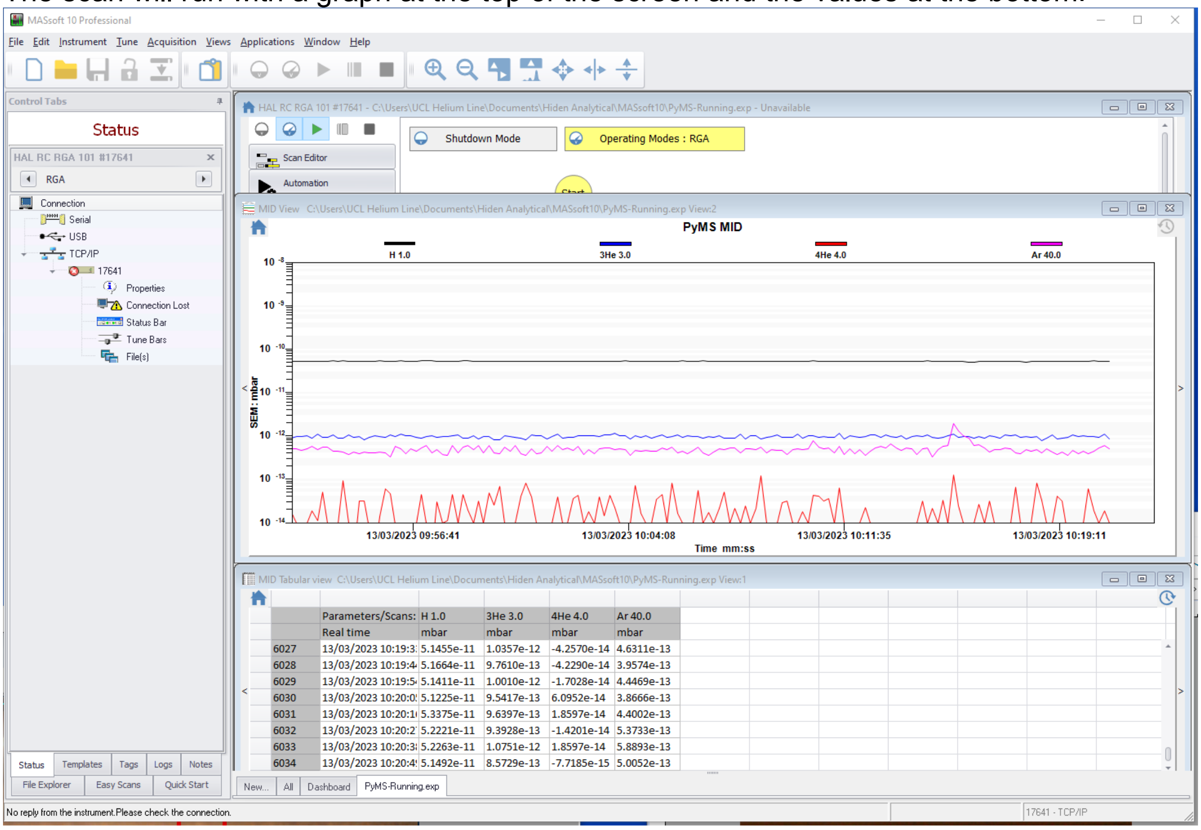
Press “OK”

A popup will ask if you wish to overwrite the file



Choose “Yes”

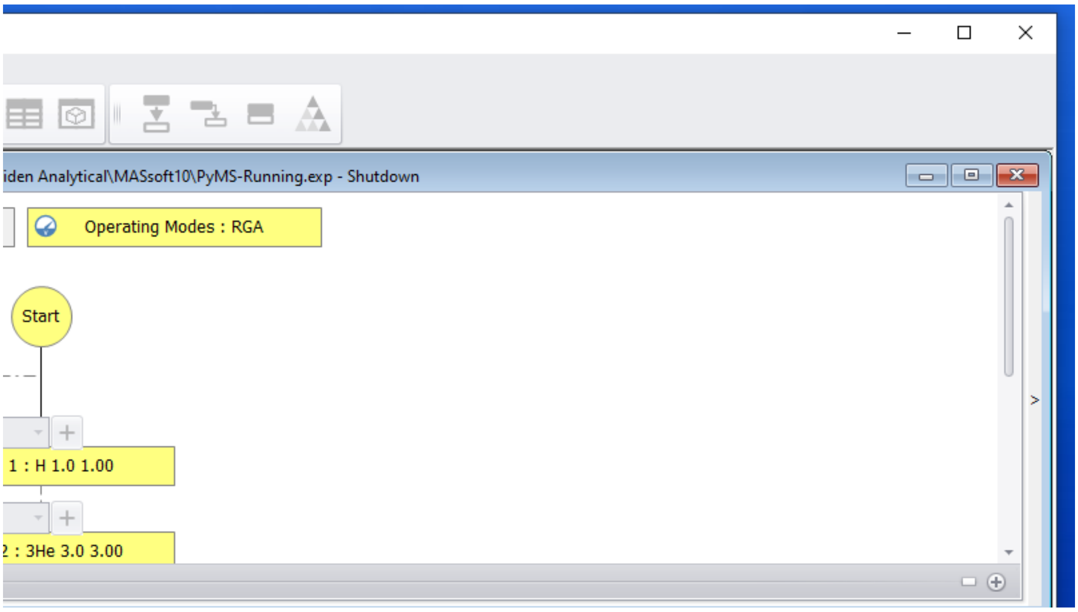
The scan will run with a graph at the top of the screen and the values at the bottom.



To stop the MID scan

Press the red square to stop the scan

Press the red x to close the window.



PyMS Software Design

The PyMS User Interface is the main control console for the Helium Line software.

It comprises of:

- A control panel that shows the status of the line including: all valves, laser, vacuum pumps, location of X-Y stage and the cameras.
- A database to store run-books containing the cycle sequencing for Line-Blanks, Standards (Q-Standards) and samples
- A database to store the contents of a planchet and the run order for processing
- A database that stores the absolute x and y locations of pits on the planchet
- A process for running a line clean at the start of a planchet run
- A mechanism to run two cycles for each grain location and track where the run is
- A process for running a line blank and Q-Standard at the start of a planchet run
- A process to purge the line mid-way through the run
- A process to run a line blank and Q-Standard mid-way through the run
- A process for running a line blank and Q-Standard at the end of a planchet run
- A process for setting the valves ready for the planchet to be extracted
- A mechanism to manually pause the planchet run at the end of the current cycle and restart from the beginning of the next cycle
- A mechanism to manually over-ride the automated system to allow for MS tuning, loading planchets or other manual processes
- A mechanism to grab images from the dynolite camera at the start of a cycle
- A mechanism to grab an image from the microscope camera 10 seconds after the laser starts
- An “emergency stop” process to shut down the laser, close all valves, halt the X-Y stage and stop processing.
- An error detection mechanism to pause the process on certain conditions e.g. low vacuum on the ion or turbo pumps, error code feedback from the other modules or a time-out on any request to another module that could represent an issue.

A process to transfer the readings from the Mass Spectrometer to a database along with metadata including the grain identifier, planchet run identifier, date, previously run Q-Standard.

The application is written in Python 3.8 (Python Software Foundation, 2020) and uses the PySide6 libraries (The Qt Company, 2020) to manage the user interface screens. The application is multi-threaded with separate threads updating valve statuses, X-Y locations and pump pressures. The application uses the requests library to communicate across the network with the other modules.

SQLite3 databases (SQLite Consortium, 2020) holds planchet pit locations, steps for the analysis cycles, details of each batch and readings from the Mass Spectrometer

Work Flow

A typical workflow would be:

1. **Manual Loading Processes**
   1.1. Load the grains into the planchet
      1.1.1. Load a Durango standard into position S1 to run as a first grain (S2 and S3 can also be used)
      1.1.2. Load the grains into locations starting at A1 through to G7
      1.1.3. Load a Durango into the location S4 to run as a last grain (S5 and S6 can also be used)
   1.2. Place the planchet into the vacuum chamber, seal and pump-down
   1.3. Create a new planchet object in the database from the PyMS Application
   1.4. Label each location that is occupied with a grain identifier
   1.5. Calibrate the X-Y table so that each location is aligned above a sample
   1.6. Arm the laser
   1.7. Press the run button on PyMS
2. **Automated processes**
   2.1. Move the laser to first sample position (normally S1)
   2.2. Run a line clean task
   2.3. Run a line blank task
   2.4. Run a Q-Sample task
   2.5. Run sample task
   2.6. Run a second sample task (as a reheat)
   2.7. Move the laser to the next sample position that contains a grain
   2.8. Repeat steps 2.5 to 2.7 until half-way through or last sample has been run
   2.9. If total sample count > 37 and process is half way through the samples
      2.9.1. Open valve to turbo pump and laser chamber
      2.9.2. Wait 2 hours while pump scavenges system
      2.9.3. Run a line blank task
      2.9.4. Run a Q-Sample task
      2.9.5. Return to step 2.4
   2.10. If last grain has been sampled
      2.10.1. Run a Q-Sample task
      2.10.2. Run a line blank task
      2.10.3. Return laser to position UL
      2.10.4. Return to manual control
      2.10.5. Close all valves
3. **Manual process**
   3.1. Unload planchet ready for dissolution

Task types

Line Clean

The first run of the day is invariably inaccurate. Gas builds up on valve seats, and operation of the valves will release this gas into the line, causing small but non-negligible systematic errors. Line clean is a 30 minute run cycle without loading the pipettes used to purge the line. This is the first task on a planchet to ensure the pipettes and line are suitably clean.

Line Blank

The line blank provides a background value for ^{4}He, through a process of purging the line and then adding a known pipette volume of ^{3}He before reading the value on the Mass Spectrometer

Q-Standard

The Q-shot, or Standard, is used to calibrate the mass spectrometer during the analytical session. The 4/3 ratio, corrected for standard tank depletion, is used for calculating the amount of sample ^{4}He in ncc. A standard run involves taking a shot of ^{3}He and ^{4}He.

Sample

After cleaning the line, a sample is heated to c. 900°C using a diode laser (808 nm) to liberate helium. A pipette of ^{3}He is added and the ratio of ^{4}He and ^{3}He measured. At the end of the Sample task the laser moves on to the next sample.

Sample + Reheat

Run the sample process twice in succession against the same sample location. The 2nd sample run (reheat) should give a value close to that of the of a line blank. If the reheat value is higher than the line blank, the sample could contain inclusions which release the helium at slower rates and will cause inaccuracy when calculating the closing age of the sample.

Pump

Run a 2 hour purging cycle to remove any build-up of gasses that accumulate during sample testing. After a pump task there should be a fresh line blank and Q-Standard to reset the background readings.

Unload

Close all valves and move the laser to the centre of the planchet ready for unloading.

Communications with equipment

Mass Spectrometer

The Hiden DLS-1 mass spectrometer is read by the “MASsoft” application included with the Mass Spectrometer. The PyMS system communicates with it via a network socket connection on port 5026. It has commands to start a Multiple Ion Detection (scanning for AMU 1, 3, 4 & 40) or a profile scan where it scans every AMU between 1 and 50.

Vacuum Valves

The Vacuum valves 1 – 13 are controlled via a command to the Valve Controller. Commands are sent via HTTP network traffic using a JSON message. The status of each valve is returned by a JSON response message from a query request to the Valve Controller.

X-Y Stage

The X-Y stage is controlled via a command to the X-Y Controller. Commands are sent via HTTP and the current position is returned if the controller is queried.

Vacuum Pressure Gauges

The pressures from the three vacuum gauges are read via an HTTP query to the Pump Reader.

N_2 Pressure Guage

The pressure of the N_2 used to operate the valves is read via an HTTP query to the Pump Reader. The laser range-finder on the pyrometer can be switched on and off by an HTTP command.

HTTP JSON Commands

JSON messages will be sent using a “PUT” request to the RESTful APIs endpoint (/api) on the valve controller and x-y controller.

The message will be in the format:

```
{
    "item": @target,
    "command": @value
}
```

Where `@target` could be a valve or stepper motor and `@value` will be the value or command needed to be implemented

e.g.

```
{
    "item": "valve11",
```

```
        "command": "close"
}
```

or

```
{
        "item": "xmoveto",
         "command": 1.05
}
```

HTTP JSON Queries

To get the status of a controller a query to the API will return a JSON message with the status of that controller. The valve controller will return the state of each valve, the X-Y controller will return the position of the x and y linear position indicators and the pressure controller will return the line pressure of the vacuum pumps and the pyrometer temperature.

Laser

The laser power is controlled via a serial connection from the PyMS application. The serial port and baud rate is configured via the Settings JSON file. The TTL control to switch on and off the laser is controlled via an HTTP command to the Valve Controller.

Dynolite Camera

The Dynolite camera is used to view the location of the laser over the sample and displays images via the DynoCapture application. The PyMS application will take a screen capture when the task item is run, it will save the image into the batch data folder and identify it by the hole and sample name.

GX Camera

The GX camera is a high-definition image down the microscope and is used to view the sample during heating. The image field of view is smaller than the size of a pit in the planchet so the dynolite is used to check location. The image from the GX Camera is stored while the laser is running and will show the sample glowing as it is heated. The PyMS application will take a screen capture and store it in the batch data folder.

PyMS Screens

Main Form

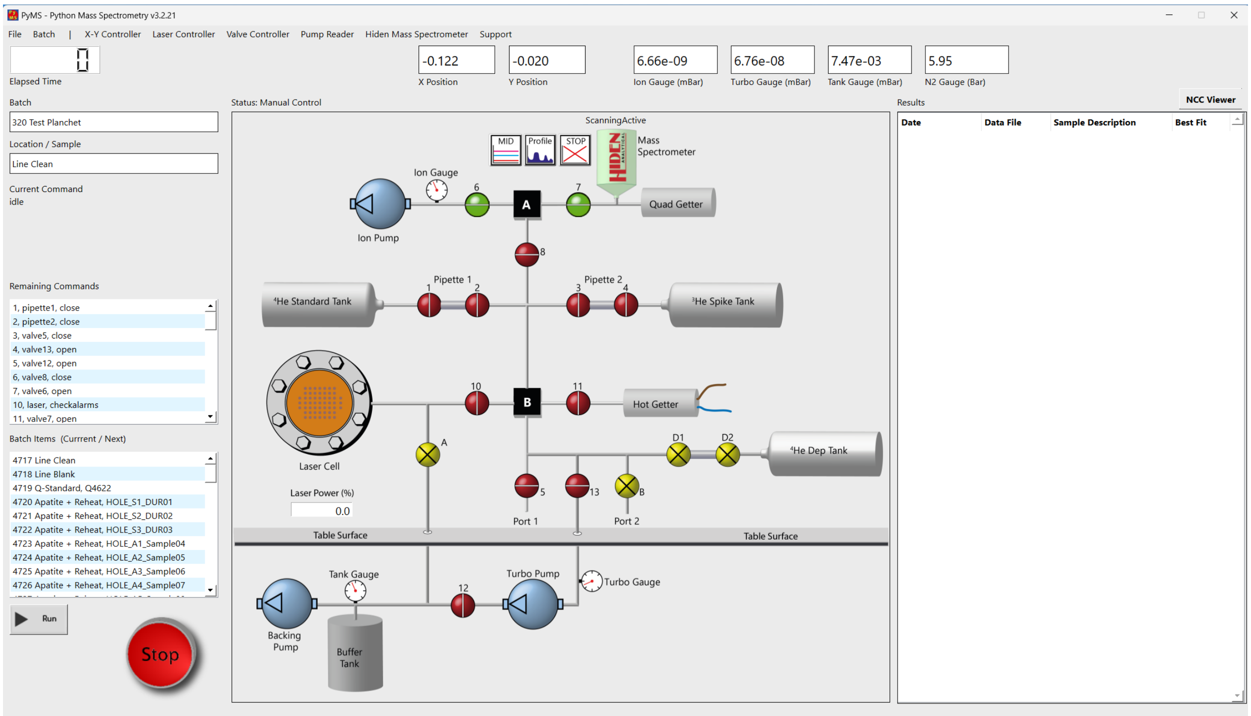


Figure 11: PyMS Main Form

The centre section of the main form shows a representation of the helium line with indicators to show the state of each valve, when the laser is running the Laser Cell will be filled with an indicator. At the top of the screen are the coordinates of the X-Y stage, the pressures of each of the vacuum gauges and the temperature reading from the Pyrometer.

The left-hand side shows the elapsed time on the current cycle, the identifier for the batch or planchet, the description of the current item being processed, a list of the steps yet to be completed along with the time they should be run (in seconds) and a list of the samples yet to be processed in the current batch or planchet.

The table on the right-side shows the results of the line-blanks, Q-Standards, samples and reheats. The same data is also written into a text file in the sample results directory.

At the bottom of the left-hand side there is a run/pause button that will stop and start the process and an emergency stop which will close all valves, stop the X-Y table moving and switch off the laser.

If the process is not running samples, the valves can be manually operated by clicking on each valve to open or close it. The valve controller has safeguards built in to prevent valves on either side of a pipette opening at the same time and allowing the ^{3}He or ^{4}He tanks to drain.

At the top of the screen are menus:

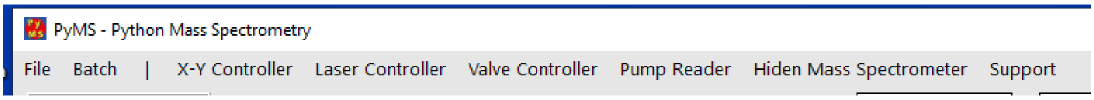


Figure 12: PyMS menus

File → Exit to close the application

Batch → Start New Batch
If there is no batch loaded (previous batch ended successfully) this will give the user the option to create a new simple batch (up to 5 tasks) which is useful for testing (e.g. line blank after line maintenance) or a new planchet. If there is an incomplete batch loaded this form will allow that batch to be edited or a new batch to be created.

X-Y Controller → Open Log Page
This opens the X-Y controllers application log page in the default browser of the computer.

X-Y Controller → Open Status Page
This opens the X-Y controllers web status page in the default browser of the computer. The page shows the x and y location returned by the position sensing potentiometers.

X-Y Controller → Reboot X-Y Raspberry Pi
This restarts the X-Y controller.

X-Y Controller → Calibrate X-Y Stage
This will open the X-Y Calibrate screen where the X-Y table can be set to move to locations from the database, manually controlled and the database locations updated. It is recommended to run a quick calibrate check before each full planchet run

Laser Controller → Open Log Page
This opens the X-Y controllers application log page in the default browser of the computer.

Laser Controller → Open Status Page
This opens the X-Y controllers web status page in the default browser of the computer. The page shows the x and y location returned by the position sensing potentiometers.

Laser Controller → Reboot Laser Raspberry Pi
This restarts the Laser controller.

Laser Controller → Manual Control
If the PYMS system is in manual control this will allow the manual laser controller to show.

Valve Controller → Open Status Page
This opens the valve controller's web status page in the default browser of the computer. The status page shows the status of each valve and the status of the main laser.

Valve Controller → Open Log Page
This opens the valve controller's application log page in the default browser of the computer.

Valve Controller → Reboot Valve Raspberry Pi
This restarts the Laser controller.

Pump Reader → Open Status Page
This opens the pump reader's web status page in the default browser of the computer. This status page shows the pressures from each gauge and the temperature the pyrometer is reading.

Pump Reader → Open Log Page
This opens the pump reader's application log page in the default browser of the computer.

Pump Controller → Reboot Pump Raspberry Pi
This restarts the Laser controller.

Hiden Mass Spectrometer → Start MID Scan
Start a MID scan on the Hiden MASSoft 10 application, the MID scan will run continuously until it is stopped.

Hiden Mass Spectrometer → Start Profile Scan
Start a Profile scan on the Hiden MASSoft 10 application, the profile scan will run for 5 cycles and then stop scanning.

Hiden Mass Spectrometer → STOP Scan
Stop the currently running scan and unload it from the Hiden software.

Support → View PyMS Log
Opens the logfile in a viewer

Support → View PyMS Settings
Opens the settings.json file in a viewer

Support → About PyMS
Shows the version and copyright notices

New Batch Form

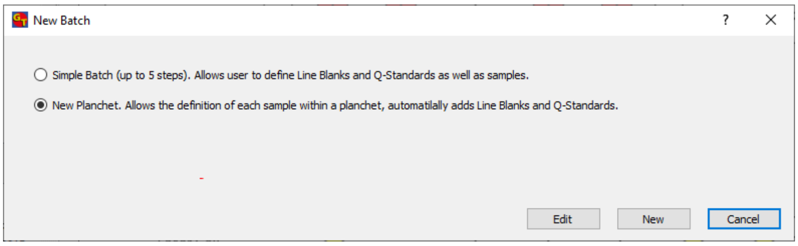


Figure 13: New Batch dialog box

Form gives option to edit or create a new planchet or simple batch. If there is an existing batch or planchet in progress, pressing “New” will close the current one and mark it as completed.

New (/edit) Planchet Form

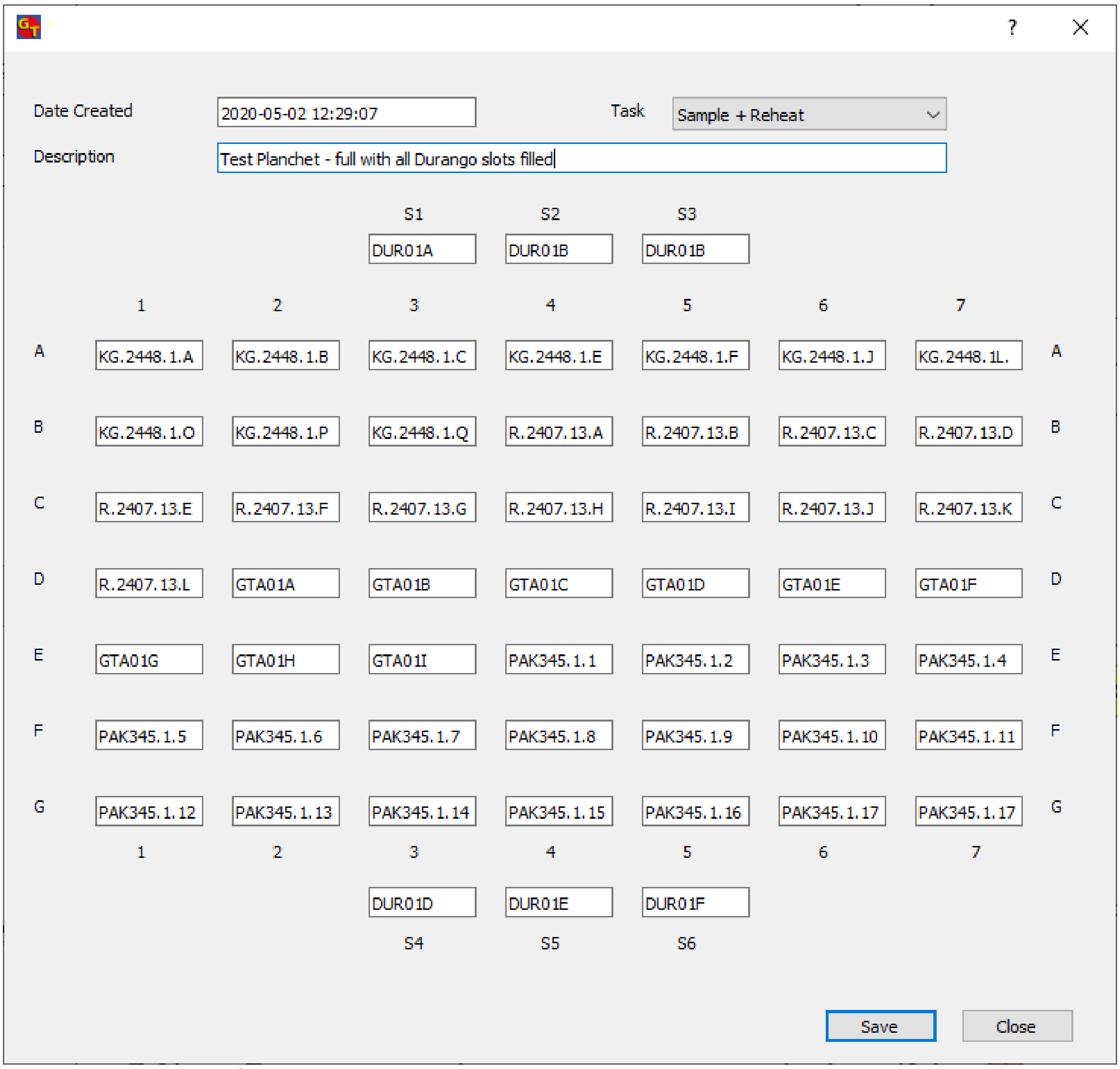


Figure 14: Planchet entry form

This form is used to create a new planchet of samples, it requires a meaningful description that will be used in the data folder the results are stored in as well as an identifier (normally a sample number for each sample loaded into the planchet). Any location that is left blank will be skipped when the planchet is processed. When the save button is pressed the planchet will automatically add line blanks and q-standards and, if the total run time exceeds 24 hours a 2 hour purging pump cycle will also be added.

There is an option to run a sample + reheat or just a single heat per location. Normally a sample + reheat would be run.

If the form was opened in edit mode from the new batch form it will display previously entered data.

New (/edit) Simple Batch

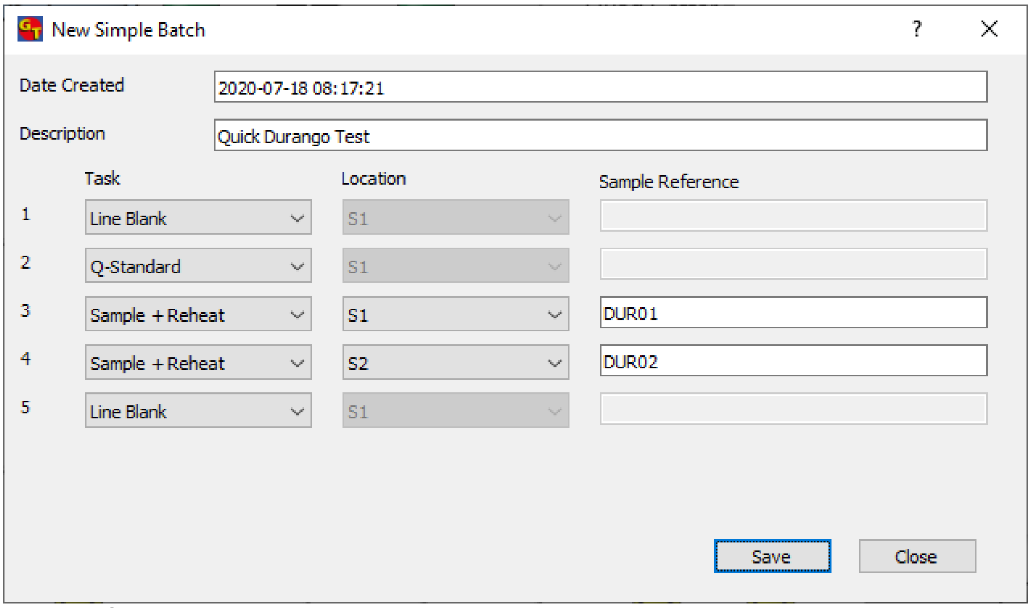


Figure 15: Simple Batch Entry Form

This form allows up to 5 steps to be added. It is mainly used for calibrating the Mass Spectrometer and running tests on the system.

X-Y Calibration Screen

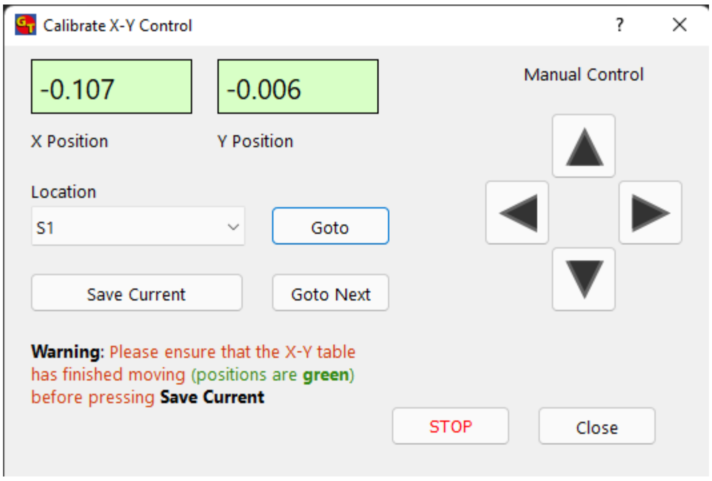


Figure 16: X-Y Stage calibration dialog

This screen allows for manual movement of the laser and microscope assembly and replaces a conventional joystick found on manual X-Y stages.

The position information comes from the linear displacement potentiometers attached to the X-Y stage. The values range from -2.5 V to +2.5 V with 1mm of displacement equating to 0.1V change. If the stage is moving, the X or Y position will have a red background. Green means the stage is stationary.

A location can be selected and then the **Goto** button will tell the X-Y controller to move the stage to the values stored in the database.

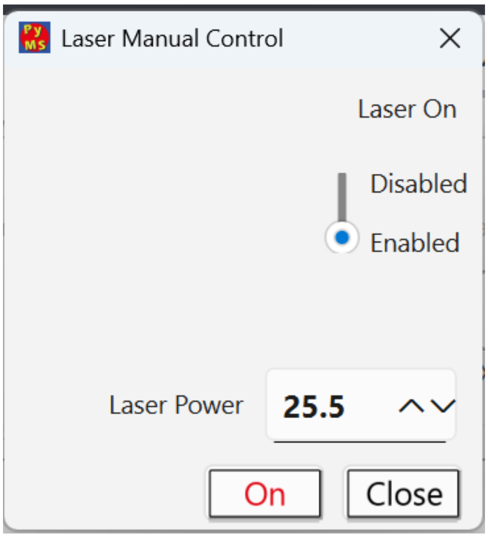
Goto Next will move the stage in a sequence from S1 to S3 and then A1 to G7 followed by S4 to S6.

Save Current will write the current X and Y position to the database for the location shown on the form.

Stop will send an “All Stop” command to the X-Y controller and stop the stage moving.

Laser Manual Control Screen

There are times when it is useful to manually control the laser.



Power control

Dial in the required power

Disabled / Enabled Slider

This needs to be moved to **Enabled** for the ON button to function

On

This will switch on the Laser. Pressing again will switch it off.

Databases

Database	Table	Description
databasepath	Locations	X and Y values for each location
	Batches	Planchet or Batch Header, id number, date created and description.
	BatchSteps	Each step on a batch or each sample within a planchet
	Cycles	Task header, name of task and id
	CycleSteps	Each command within a task, details of the command and the time it has to happen
resultsdatabasepath	HeliumRuns	Header per sample, reheat, line blank or Q-Sample. Contains date and time run, location details, laser power
	MSRawData	Data points at each reading time point from Mass Spectrometer

Table 5: Database files and data tables

Data Files

Data files are stored in the “documents\Helium Line Data” folder.

The application will create a new folder for each sample batch or planchet, the folder will contain the batch description, created when the planchet was loaded.

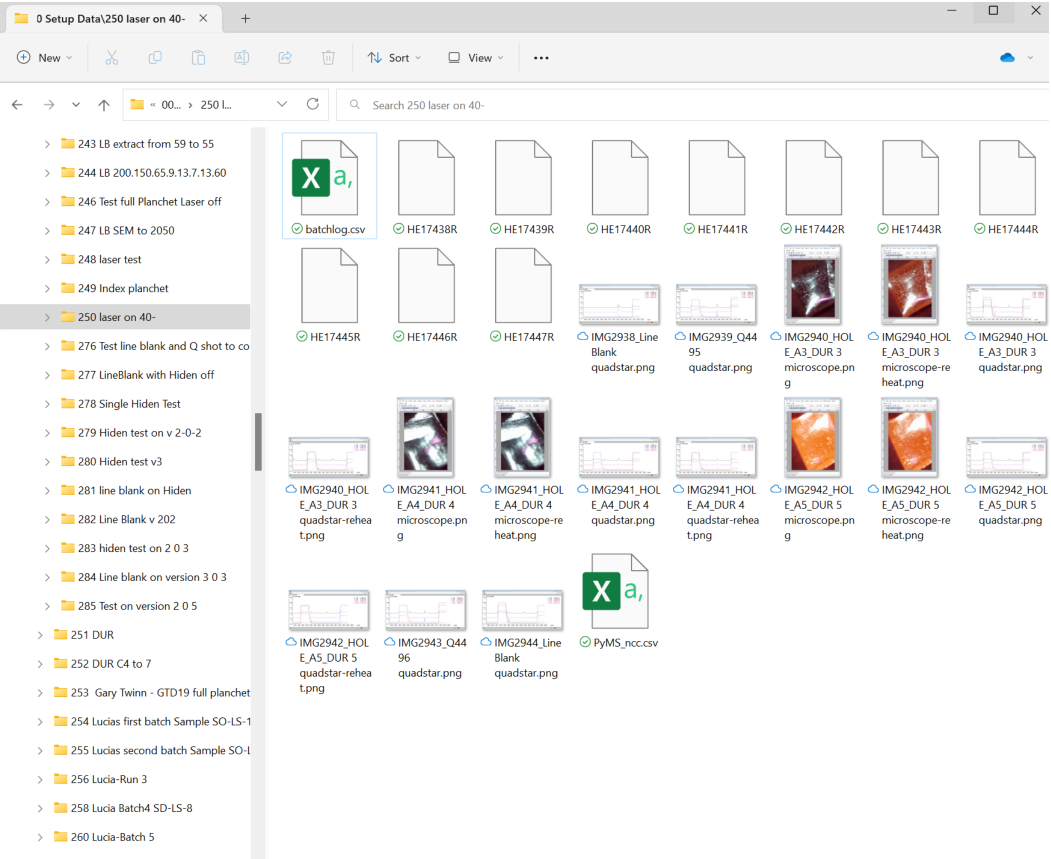


Figure 17: Location of data files

The folder will contain the following files:

Batch log “batchresults.csv”

Contains the time each sample was taken, the details of the sample (hole number and grain identifier) and the best-fit value. This data is similar to data that is logged in the UCL helium Line lab book. The file is a CSV file that can be read directly into Excel.

Data file “HEnnnnnR”

The file contains the Mass Spectrometer data in a format that can be processed via the PyMS Ncc Calculator application. One file will be created for each Line Blank, Q-Standard, Sample and sample reheat.

Ncc File "PyMS_ncc.csv"

File contains the results of the ncc calculations for each Data File. It is an Excel readable file and has the 4He/3He ratio and standard error as well as the blank corrected version and the ncc value with the ncc error.

Image files "IMG** hiden-mid"

Image of the Mass Spectrometer application showing readings from the MID Scan.

"IMG** hiden-profile"

Image of the Mass Spectrometer application showing readings from the Profile Scan.

"IMG*** dynolite.png"

Image from the dynolite camera showing the general location of the camera.

"IMG*** microscope.png"

Image from the microscope camera taken when the laser is activated.

"batchresults.csv"

Contains the time each sample was taken, the details of the sample (hole number and grain identifier) and the best-fit value. This data is similar to data that is logged in the UCL helium Line lab book. The file is a CSV file that can be read directly into Excel.

Settings file

The configuration of the application is held within the “settings.json file” within the application directory. The file contains locations of the mass spectrometer, pump reader, x-y host and valve controller.

Changes to the settings file can be made by shutting down PyMS application and editing the settings file in notepad.

The sections of the file are:

Mass spectrometer location and data output directory

```
"MassSpec": {
        "datadirectory": "C:\\Users\\UCL Helium Line\\Documents\\Helium Line Data\\",
        "hidenMID": "C:\\Users\\UCL Helium Line\\Documents\\Hiden Analytical\\MASsoft10\\PyMS-MID.exp",
        "hidenProfile": "C:\\Users\\UCL Helium Line\\Documents\\Hiden Analytical\\MASsoft10\\PyMS-Profile.exp",
        "hidenRunfile": "C:\\Users\\UCL Helium Line\\Documents\\Hiden Analytical\\MASsoft10\\PyMS-Running.exp",
        "hidenhost": "192.168.1.6",
        "hidenport": 5026,
        "multiplier": 1e-12,
        "nextH": "HE18890R",
        "nextQ": 4553,
        "startimeoffset": 15,
        "timeoutretries": 10
    },
```

Database locations

```
    "database": {
        "databasebackuppath": ".\\database\\PyMs.backup.db",
        "databasepath": ".\\database\\PyMs.db",
        "resultsdatabasebackuppath": ".\\database\\HeliumResults.db.backup.db",
        "resultsdatabasepath": ".\\database\\HeliumResults.db"
    },
```

Raspberry Pi function controllers

```
    "hosts": {
        "pumphost": "http://192.168.2.5/api",
        "valvehost": "http://192.168.2.3/api",
        "xyhost": "http://192.168.2.4/api"
    },
```

Applications to take images of

```
    "image": {
        "dynolite": "DinoCapture 2.0",
        "microscope": "GXCapture-T",
        "microscope-reheat": "GXCapture-T",
        "Mass Spectrometer": "[M1] PyMS MASS SPECTROMETER 32-bit Measurement - [MID  < kens4he.mip >]",
        "Mass Spectrometer-reheat": "[M1] PyMS MASS SPECTROMETER 32-bit Measurement - [MID  < kens4he.mip >]"
```

```
    },
```

Laser configuration

```
    "laser": {
        "power": 30.0
    },
```

Location of log files

```
    "logging": {
        "logappname": "PyMS",
        "logfilepath": ".\\logs\\"
    },
```

Vacuum gauge settings

```
    "vacuum": {
        "ion": {
            "high": 2.1e-09
        },
        "tank": {
            "high": 0.0001
        },
        "turbo": {
            "high": 9.9e-08
        }
```

Valve Controller Raspberry Pi Microservice

The valve and laser control software consists of a python (Python Software Foundaton, 2020) application to control the 12 valves of the Helium line and a single channel Transistor - Transistor Logic (TTL) output to switch on/off the Laser.

Valve	Description
1	^{4}He pipette input
2	^{4}He pipette output
3	^{3}He pipette output
4	^{4}He pipette input
5	port 1(not currently used)
6	ion pump
7	gas analyser
8	gallery A
10	laser cell
11	getter
12	buffer tank
13	turbo pump

Table 6: He line valves

The computer that controls the valves is a Raspberry Pi 4B (Raspberry PI Foundation, 2020), it uses a 40 way connector to break out the required 12 GPIO pins and connect to the driver boards.

Each valve is controlled by a dedicated GPIO line on the raspberry Pi computer. The valves run on 24V DC and the Pi uses a 3.3V signal, so a driver circuit is required to step up the voltage and current from the Pi.

The computer is controlled via a RESTful API listening on port 80 for a valid json message.

The software contains logic to prevent the input and output valves on a pipette opening at the same time in the case of a mistake in the commands sent to the computer. (valve 1 and valve 2), (valve 3 and valve 4)

To simplify charging and unloading of pipettes a single command was implemented to load and unload each pipette. The pipette command contains a 0.5s delay between one valve closing and one opening.

A single command was also implemented to close all valves and switch off the laser in case of an issue.

Input Messages

Json messages in the following formats are accepted:

Open a single valve (nn):

```
{
    "item": "valvenn",
    "command": "open"
}
```

Close a single valve (nn):

```
{
    "item": "valvenn",
    "command": "close"
}
```

Close all valves:

```
{
    "item": "closeallvalves"
    "command": ""}
```

Get Valve Status: (for all valves and laser)

```
{
    "item": "getvalvestatus",
    "command": ""
}
```

Output Messages

Following any valid command, the following data is returned: (ss = open or closed, xx = on or off)


```
{
    "status": "ss",
    "valve": 1
},
{
    "status": "ss",
    "valve": 2
},
{
    "status": "ss",
    "valve": 3
},
{
    "status": "ss",
    "valve": 4
},
{
    "status": "ss",
    "valve": 5
},
{
    "status": "ss",
    "valve": 6
},
{
    "status": "ss",
    "valve": 7
},
{
    "status": "ss",
    "valve": 8
},
{
    "status": "ss",
    "valve": 10
},
{
    "status": "ss",
    "valve": 11
},
{
    "status": "ss",
    "valve": 12
},
{
    "status": "ss",
    "valve": 13
}
```

Valve to GPIO assignment

Valve	Designation	Connector		Designation	Valve
	3v3	1	2	**5v**	
	GPIO 02	3	4	**5v**	
	GPIO 03	5	6	**GND**	
	GPIO 04	7	8	GPIO 14	
	GND	9	10	GPIO 15	
Valve 2	**GPIO 17**	11	12	**GPIO 18**	Valve 5
Valve 6	**GPIO 27**	13	14	**GND**	
Valve 10	**GPIO 22**	15	16	**GPIO 23**	Valve 1
	3v3	17	18	**GPIO 24**	Valve 8
	GPIO 10	19	20	**GND**	
Valve 7	GPIO 09	21	22	GPIO 25	
Valve 11	GPIO 11	23	24	**GPIO 08**	
	GND	25	26	**GPIO 07**	
	GPIO 00	27	28	GPIO 01	
	GPIO 05	29	30	**GND**	
	GPIO 06	31	32	**GPIO 12**	Ready LED
Valve 3	**GPIO 13**	33	34	**GND**	
Valve 4	**GPIO 19**	35	36	**GPIO 16**	
Valve 13	**GPIO 26**	37	38	GPIO 20	
	GND	39	40	**GPIO 21**	Valve 12

Table 7: GPIO to valve assignments

Table 6 shows GPIO channels on a Raspberry Pi 4B, channels in green are available and remain at 0v during the Pi boot up sequence, it is important that no lines are used that could cause the laser to turn on or a valve to open before the software is ready.

Once the boot sequence has completed and the software has started the final command on the initialising function will be to light the “Ready LED” (GPIO12) to give a visual indication the Raspberry PI had booted and is ready to accept commands on the REST api.

Driver circuits and power supply

The Raspberry Pi 4 uses a 5v power supply provided via a USB-C connector. A 5V power supply rated at 3A will be required. The Raspberry Pi internal voltage and the outputs of the GPIO connectors is 3.3v.

The valves are opened by applying 24V DC to the solenoid, when the power is removed the valves will close against a spring. The assumption is that the maximum number of valves that could be actuated (during a pump-down cycle) would be 10 valves so the supply should be able to supply a minimum of 1.0A continuously. In order to provide this level of current reliably a 24V power supply with a continuous rating of 3.0A will be required.

In order to protect the Raspberry Pi from any voltage spikes that may be generated by the solenoids as the valves close, or a faulty component, an opto-isolator will be required to provide full electrical isolation between the Pi and the driver circuit.

The solenoids on the valves will be driven via Metal Oxide Field Effect Transistors (MOSFET). MOSFETs have a very low forward voltage once switched on and have a high-power capacity (Inchange Semiconductor, 2016).

As the valves are operated by inductive solenoids, when the power is removed from the solenoids there is a chance voltage spikes will be induced into the circuit so a Schottky diode will protect the MOSFET and opto-isolator.

Each valve driver board has 4 channels so the valve controller requires 3 driver boards to control the 12 valves (Figure 18).

The driver circuit will need to be housed in an enclosure that has external connectors for the mains supply, ethernet cable and connectors for the 12 valve cables. It must have ventilation holes to prevent the Raspberry Pis, power supplies or driver MOS-FET transistors overheating. Indicator lights should show the status of the 5v PSU for the Raspberry Pi, the 24V valve PSU and the software ready light.

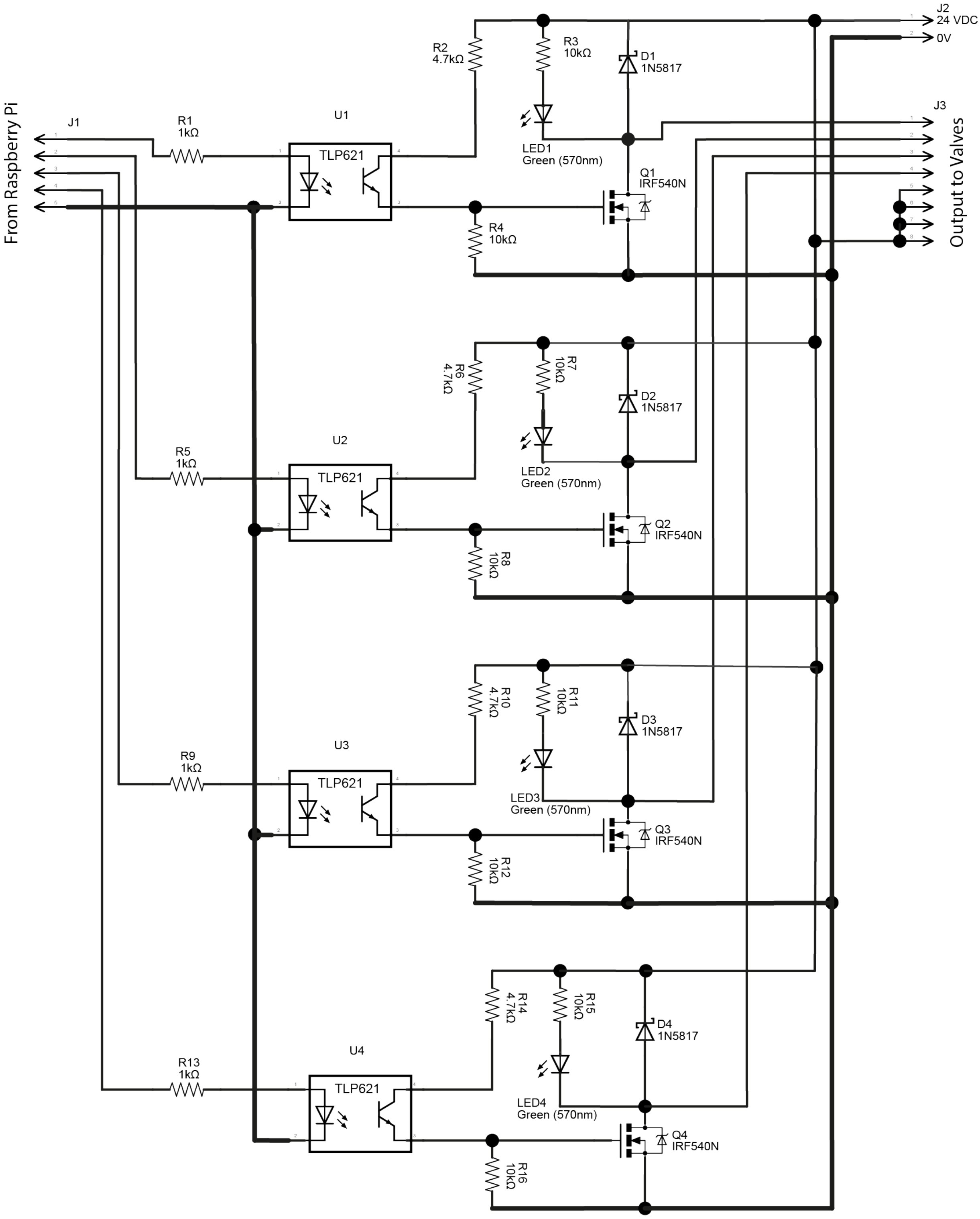
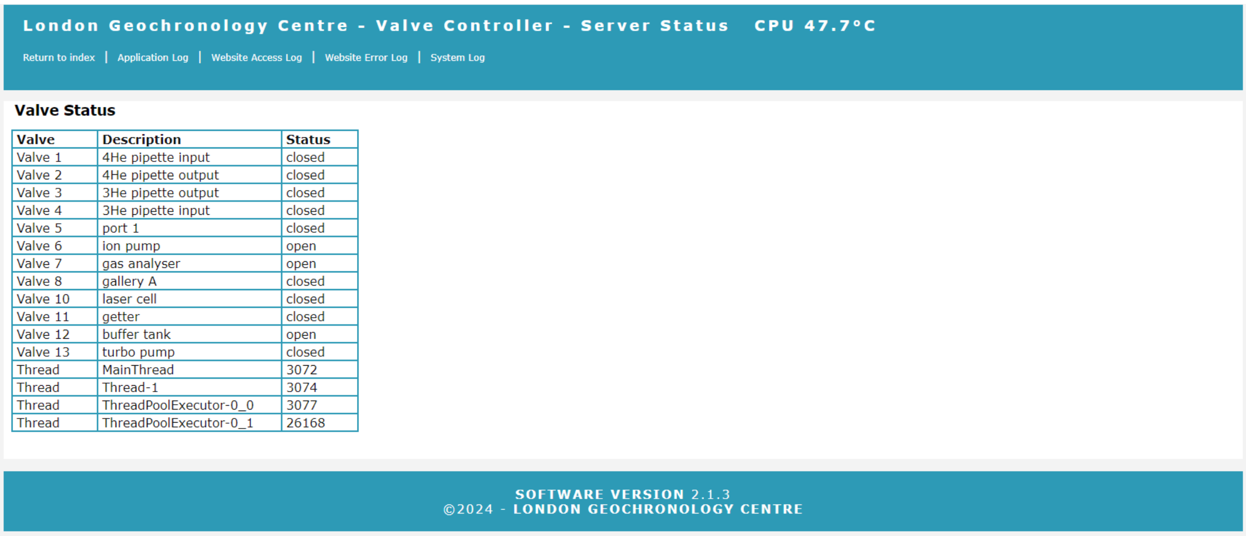


Figure 18: Schematic for a valve driver board

Web interface

As well as accessing the status of the valves via the RESTful API a read-only web interface that can be accessed directly from a browser is will be available to view the valve status, application and web server logs.



London Geochronology Centre - Valve Controller - Server Status CPU 47.7°C

Return to index | Application Log | Website Access Log | Website Error Log | System Log

Valve Status

Valve	Description	Status
Valve 1	4He pipette input	closed
Valve 2	4He pipette output	closed
Valve 3	3He pipette output	closed
Valve 4	3He pipette input	closed
Valve 5	port 1	closed
Valve 6	ion pump	open
Valve 7	gas analyser	open
Valve 8	gallery A	closed
Valve 10	laser cell	closed
Valve 11	getter	closed
Valve 12	buffer tank	open
Valve 13	turbo pump	closed
Thread	MainThread	3072
Thread	Thread-1	3074
Thread	ThreadPoolExecutor-0_0	3077
Thread	ThreadPoolExecutor-0_1	26168

SOFTWARE VERSION 2.1.3
©2024 - LONDON GEOCHRONOLOGY CENTRE

Figure 19: Web status page

Interfaces

Valve Cable Specification

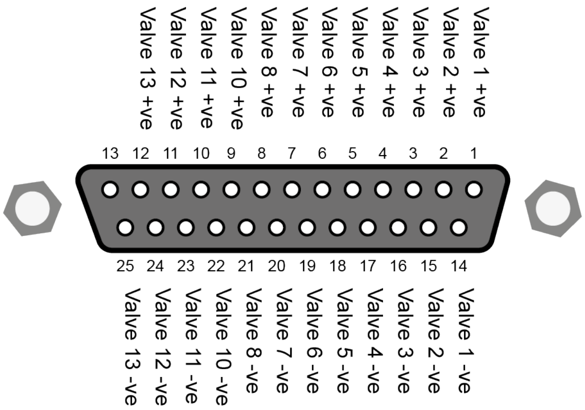


Figure 20: Valve controller D plug assignments

A 25 pin D plug connected to 12 individual cables, each terminated with an SMC 24v latching power plug.

X-Y Controller Raspberry Pi Microservice

The X-Y controller software consists of a python (Python Software Foundation, 2020) application to control the 2 stepper motors on the X-Y Stage of the Helium line.

The X-Y table consists of 2 Bipolar stepper motors to drive the table in two dimensions. A stepper motor consists of a pair of windings A-AA and B-BB surrounding a rotor containing permanent magnets (Figure 21)

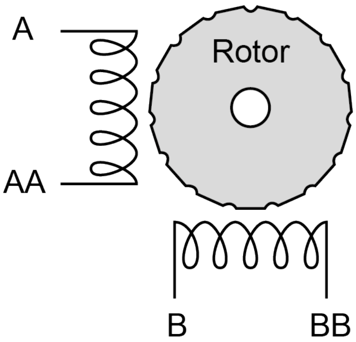


Figure 21: Bipolar Stepper Motor

The stepper motor is operated by applying a voltage across the A and B windings which causes the permanent magnets on the rotor to either attract or repel. The voltage is applied a with the positive to A and B and the negative to AA and BB. To move the rotor one winding will have the voltage reversed in turn. At each transition where the voltage is reversed on a winding the rotor will rotate 1.8° clockwise. By reversing the sequence the rotor will rotate anti-clockwise. By grounding both connections during of one winding between the reversals will cause the rotor to take a half step. Adding a delay between transitions will set a speed for the motor, a fast speed would have 25ms between transitions whereas a slower speed is used for accurately locating the laser over the sample.

	A	AA	B	BB	
0	+	-	+	-	Initial State
1	+	-	-	+	B reversal
2	-	+	-	+	A reversal
3	-	+	+	-	B revert
4	+	-	+	-	A revert

Table 8: Full step sequence for bipolar stepper motor

	A	AA	B	BB	
0	+	-	+	-	Initial State
1	+	-	-	-	½ step ground B
2	+	-	-	+	B reversal
3	-	-	-	+	½ step ground A
4	-	+	-	+	A reversal
5	-	+	-	-	½ step ground B
6	-	+	+	-	B revert
7	-	-	+	-	½ step ground A
8	+	-	+	-	A revert

Table 9: Half step sequence for bipolar stepper motor

There are two 2KΩ linear displacement sensors with a 50mm travel mounted on the x-y stage, one in each direction. By applying a voltage to each end of the sensor, the location of the wiper along the sensor can be determined. The travel along the planchet is 10.5mm in each of the X directions (from the origin) and 17.5mm from the origin in each of the Y directions.

The 2KΩ displacement sensor will have a reference voltage of +5.08 volts applied to one static connection, the other static connection will be held to 0V. That will give a voltage gradient along the 50mm of 0.1V per mm.

A joystick will be required for manual control of the X-Y table.

The computer required to control the valves will be a Raspberry Pi 4 (Raspberry PI Foundation, 2020), it will require a connector to break out the required 12 GPIO pins and connect to the driver boards.

Each stepper motor will be connected to 4 dedicated GPIO lines on the raspberry Pi computer. The valves run on 5V DC @ 1 A, so a driver circuit will be required to step up the voltage and current from the Pi. The linear

The Raspberry Pi computer will be controlled via a RESTful API listening on port 80 for a valid json message.

Input Messages

Json messages in the following formats will be accepted:

Move in X direction to a location (voltage). Where n is a floating-point number between -2.5 and +2.5. Once started the stepper will move at a relative high speed (30 steps per second) until the position voltage is within 0.2V (~ 2mm) at which point it will slow to a rate of 10 steps a second until the position voltage is within 0.1V (~ 1mm). At 0.1V the rate will slow again to 3 steps per second until the desired position has just passed, at that point the stepper will reverse for 1 step to position as the table as accurately as possible with the stepper motors attached to the x-y stage.

```
{
    "item": "xmoveto",
    "command": n
}
```

Move in X direction nn steps. Where nn is a positive or negative integer depending on the direction or motor rotation. Sending a value of 0 will cause the stepper motor to halt immediately and disengage power to the x direction stepper motor.

```
{
    "item": "xmove",
    "command": nn
}
```

Move in Y direction to a location (voltage). Where n is a floating-point number between -2.5 and +2.5.

```
{
    "item": "ymoveto",
    "command": n
}
```

Move in Y direction nn steps. Where nn is a positive or negative integer depending on the direction or motor rotation. Sending a value of 0 will cause the stepper motor to halt immediately and disengage power to the y direction stepper motor.

```
{
    "item": "ymove",
    "command": nn
}
```

Output Messages

Following any valid command, the following query is returned: (ss = open or closed)


```
{
    "xpos": xnn,
    "ypos": ynn
}
```


Where xnn and ynn are floating point numbers denoting the output voltages of each displacement sensor.

Control to GPIO assignment

Control	Designation	Connector		Designation	Control
	3v3	1	2	**5v**	
i2c Data	GPIO 02	3	4	**5v**	
i2c Clock	GPIO 03	5	6	**GND**	
GP clock	GPIO 04	7	8	GPIO 14	
	GND	9	10	GPIO 15	
X Stepper A	**GPIO 17**	11	12	**GPIO 18**	Y Stepper A
X Stepper AA	**GPIO 27**	13	14	**GND**	
X Stepper B	**GPIO 22**	15	16	**GPIO 23**	Y Stepper AA
	3v3	17	18	**GPIO 24**	Y Stepper B
	GPIO 10	19	20	**GND**	
Y Stepper BB	**GPIO 09**	21	22	GPIO 25	
Joystick X+	**GPIO 11**	23	24	GPIO 08	
	GND	25	26	GPIO 07	
	GPIO 00	27	28	GPIO 01	
	GPIO 05	29	30	**GND**	
	GPIO 06	31	32	**GPIO 12**	Ready LED
X Stepper BB	**GPIO 13**	33	34	**GND**	
	GPIO 19	35	36	**GPIO 16**	Joystick X-
Joystick Speed	**GPIO 26**	37	38	**GPIO 20**	Joystick Y+
	GND	39	40	**GPIO 21**	Joystick Y-

Table 10: Control to GPIO Assignments

The above assignments are based on a raspberry Pi 4B. GPIO channels in green are available and remain at 0v during the Pi boot up.

Once the boot sequence has completed and the software has started the final command on the initialising function will be to light the “Ready LED” to give a visual indication the raspberry PI has booted and is ready to accept commands on the REST api.

The Joystick connections will need to be set to inputs with an internal pull-up resistor, thus requiring a connection to GND in order to trigger movement.

Driver circuit and power supply

The Pi 4B uses a 5v power supply provided via a USB-C connector. A 5V power supply rated at 3A will be required.

The GPIO outputs are 3.3v at 50mA, the stepper motors require 5V DC @1A to rotate but the current could increase above 1A under a heavy load. In order to drive the 4 windings in the stepper motors a driver circuit will be required to step up the voltage and current (Figure 22). A power supply with a continuous rating of 6.0A will be required to ensure reliability. To protect the Pi Zero from any voltage spikes that may be generated by the steppers, or a faulty component on the hight voltage/high current side of the driver, an opto-isolator will be required to provide full electrical isolation between the Pi and the driver circuit.

The stepper motors will be driven via an integrated circuit containing four Metal Oxide Field Effect Transistors (MOSFET). MOSFETs have a very low forward voltage once switched on and have a high-power capacity. As the stepper windings are inductive loads, when the power is removed from the windings there is a chance voltage spikes will be induced into the circuit so a Schottky diode will protect the MOSFET and opto-isolator.

An analogue to digital convertor will be required to read the voltage from the position sensors and turn it into a signal the Raspberry Pi computer can read. The ADC chosen was from Able Electronics (Able Electronics UK 2020) and a resolution of 12 bits was chosen giving the controller the ability to detect 4096 voltage steps. Given that a 50mm travel on the position sensor would give a voltage reading between 0V and 5V the voltage to distance conversion is 0.1V / mm. Dividing the 5V supply by the step resolution steps gives 0.0012V per step or a resolution of 0.012 mm.

The driver circuits will need to be housed in an enclosure that has external connectors for the mains supply, ethernet cable and connectors for the x and y steppers, x and y position sensors and joystick cables. It must have ventilation holes to prevent the Raspberry Pi's, power supplies or driver MOS-FET transistors overheating. Indicator lights should show the status of the 5v PSU for the Raspberry Pi, the 5V Stepper PSU and the ready light to show the software has started.

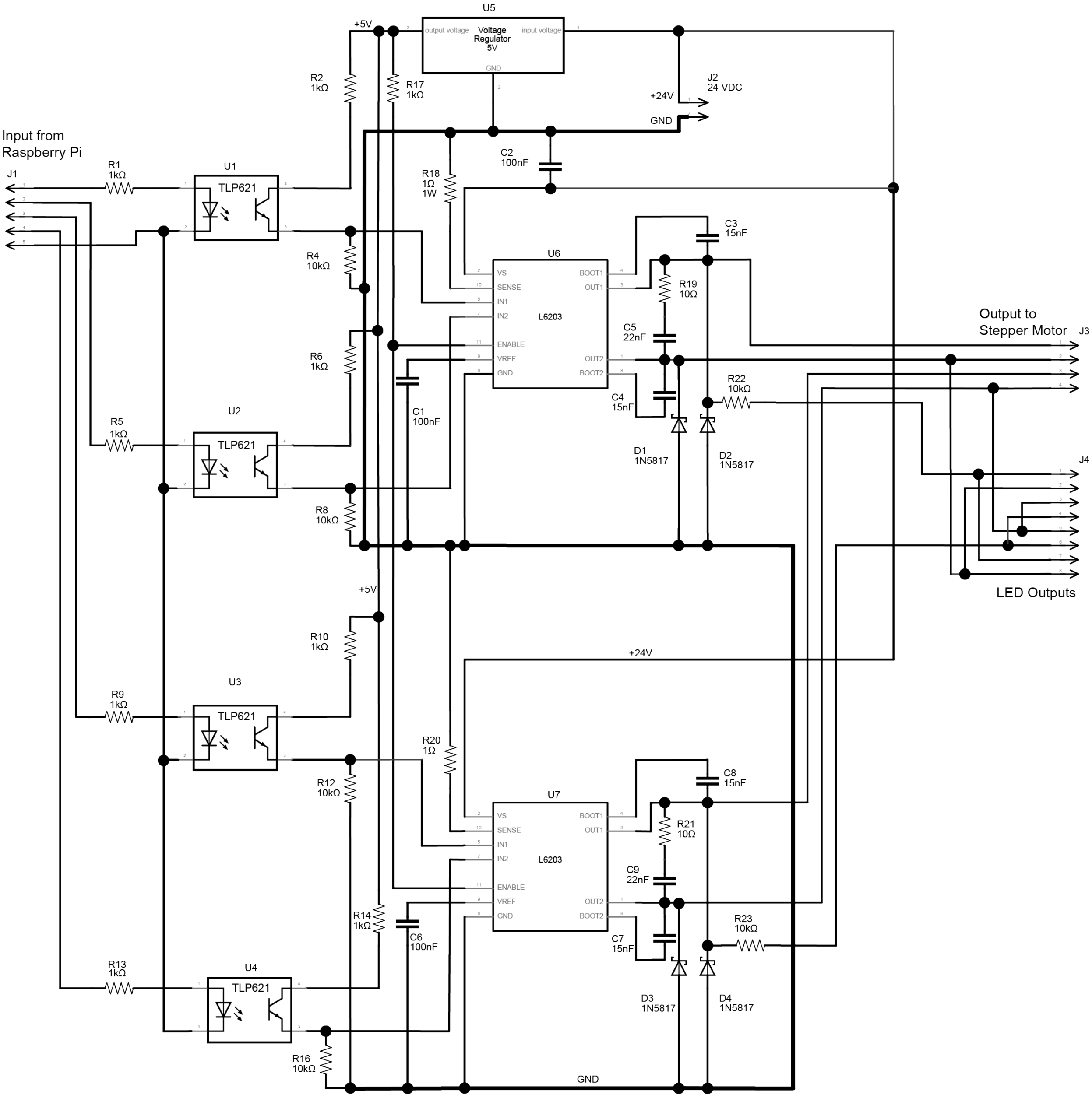


Figure 22: Schematic for a valve / stepper driver board

Web interface

As well as accessing the status of the valves via the RESTful API a read-only web interface that can be accessed directly from a browser is will be available to view the valve status, application and web server logs.

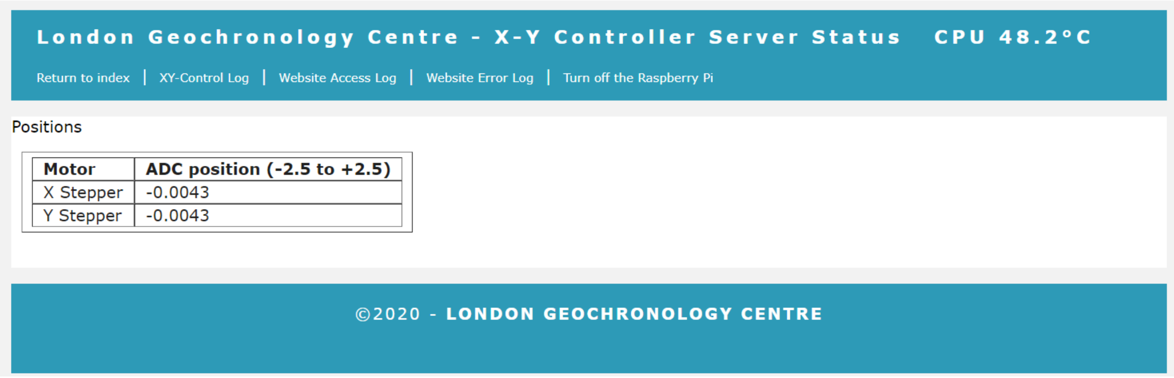


Figure 23: Web status page

Interfaces

The connection from the back of the housing X and Y steppers, X and Y position sensors and Joysticks will be a standard 9pin D Plugs

Stepper drive cable specification

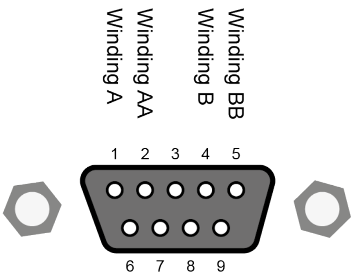


Figure 24: Stepper motor D plug assignments

The stepper drive cable consists of a male 9 pin D-plug for the controller connection, a 6 way cable with a minimum of a 1 amp capacity and a female 9 pin D-Plug for the stepper motor connector. Two cables are required, one for each stepper

Position sensor cable specification

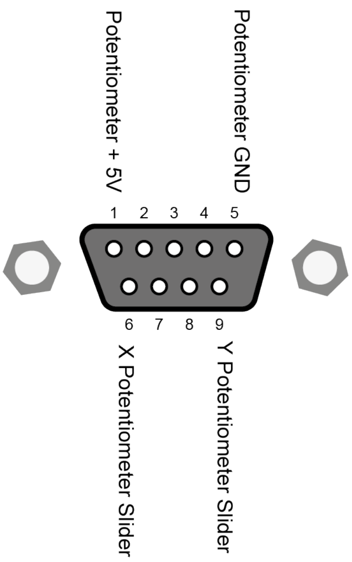


Figure 25: Position sensor D plug assignments

The position sensor cable consists of a male 9 pin D-plug for the controller connection, a 4 way shielded cable and a 4 pin DIN connector for the X-Y table connection.

Joystick Connection Specification:

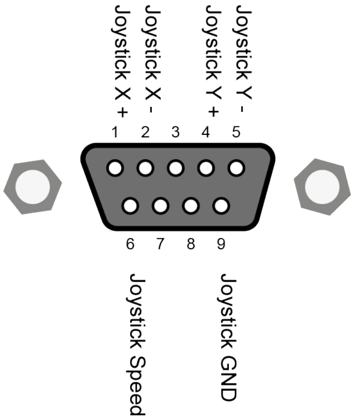


Figure 26: Joystick D plug assignments

Laser Controller Raspberry Pi Microservice

The control software consists of a python (Python Software Foundaton, 2020) application to control the Transistor Logic (TTL) output to switch on/off the Laser and RS232 serial control for laser settings.

The computer that controls the valves is a Raspberry Pi 4B (Raspberry PI Foundation, 2020), it uses a 40 way connector to break out the required 12 GPIO pins and connect to the driver boards.

The computer is controlled via a RESTful API listening on port 80 for a valid json message.

The controller also operates a single 5V TTL line to switch on and off the laser. The Raspberry Pi outputs are 3.3v and TTL is based on 5V so a buffer was needed to step the voltage up from 3.3V to 5V.

A single command was also implemented to close all valves and switch off the laser in case of an issue.

Input Messages

Json messages in the following formats are accepted:

Switch off the laser

```
{"laser": "off"}
```

Switch on the laser

```
{"laser": "on"}
```

Set the laser power to nn.n%

```
{"setlaserpower": nn.n}
```

Read the laser Alarm status

```
{"laseralarm": 1}
```

Read the laser status (returns pawer and if the laser is firing)

```
{"laserstatus": 1}
```

change the default maximum time the laser can fire to nnn seconda (default is 300)

```
{"setlasertimeout": nnn}
```

Restart the rsapberry pi after a 15 second delay

```
{"restart": "pi"}
```

Output Messages

Following a “laserstatus” command, the following data is returned:

```
{
    "laser": ss,
    " power ": pp.p
}
```

Where ss is the status (1 = laser on, 0 = laser off) and pp.p is the laser power in %

Following a “laseralarm” command, the following data is returned:

```
{
    "laser": ss,
    " power ": pp.p,
    " status ": aaa
}
```

Where ss is the status (1 = laser on, 0 = laser off) and pp.p is the laser power in % and alarm is in the table below:

aaa	Description
129	Emergency Active (key switch off and enable button off)
130	Wait for Start (key switch position 1 and enable button off)
132	Laser stand by (key switch position2 and enable button off)
133	Laser ready (laser ready to go)
141	Enable not allowed (laser after a power fail and key switch position 2 and enable button off)

Table 11: Laservall Alarm codes

Valve to GPIO assignment

Designation	Connector		Designation	Valve
3v3	1	2	**5v**	
GPIO 02	3	4	**5v**	
GPIO 03	5	6	**GND**	
GPIO 04	7	8	GPIO 14	
GND	9	10	GPIO 15	
GPIO 17	11	12	**GPIO 18**	
GPIO 27	13	14	**GND**	
GPIO 22	15	16	**GPIO 23**	
3v3	17	18	**GPIO 24**	
GPIO 10	19	20	**GND**	
GPIO 09	21	22	GPIO 25	
GPIO 11	23	24	**GPIO 08**	
GND	25	26	**GPIO 07**	
GPIO 00	27	28	GPIO 01	
GPIO 05	29	30	**GND**	
GPIO 06	31	32	**GPIO 12**	
GPIO 13	33	34	**GND**	
GPIO 19	35	36	**GPIO 16**	Laser TTL
GPIO 26	37	38	**GPIO 20**	
GND	39	40	**GPIO 21**	

Table 12: GPIO to valve assignments

Table 6 shows GPIO channels on a Raspberry Pi 4B, channels in green are available and remain at 0v during the Pi boot up sequence, it is important that no lines are used that could cause the laser to turn on or a valve to open before the software is ready.

Once the boot sequence has completed and the software has started the final command on the initialising function will be to light the “Ready LED” (GPIO12) to give a visual indication the Raspberry PI had booted and is ready to accept commands on the REST api.

Driver circuits and power supply

The Raspberry Pi 4 uses a 5v power supply provided via a USB-C connector. A 5V power supply rated at 3A will be required. The Raspberry Pi internal voltage and the outputs of the GPIO connectors is 3.3v.

The laser requires a TTL control signal with a voltage higher than 4v for an ‘ON’ and a voltage lower than 1v for an ‘OFF’. A SN74HC125NE4 quad buffer integrated circuit (Texas Instruments, 1984) will be used to step the 3.3v signal up to a TTL signal for the Laser control (Figure 27).

The driver circuit will need to be housed in an enclosure that has external connectors for the mains supply, ethernet cable and connectors for the laser.

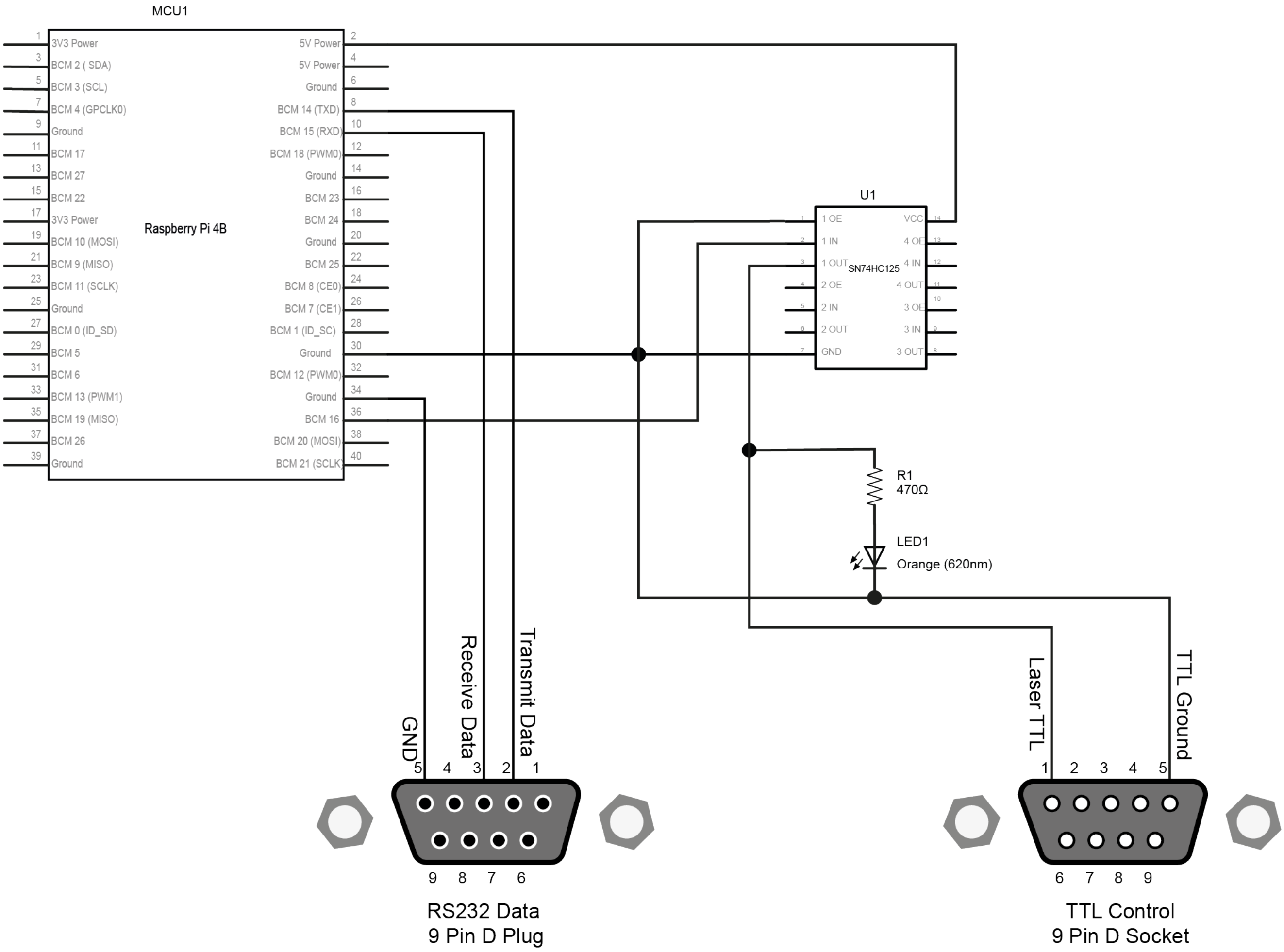
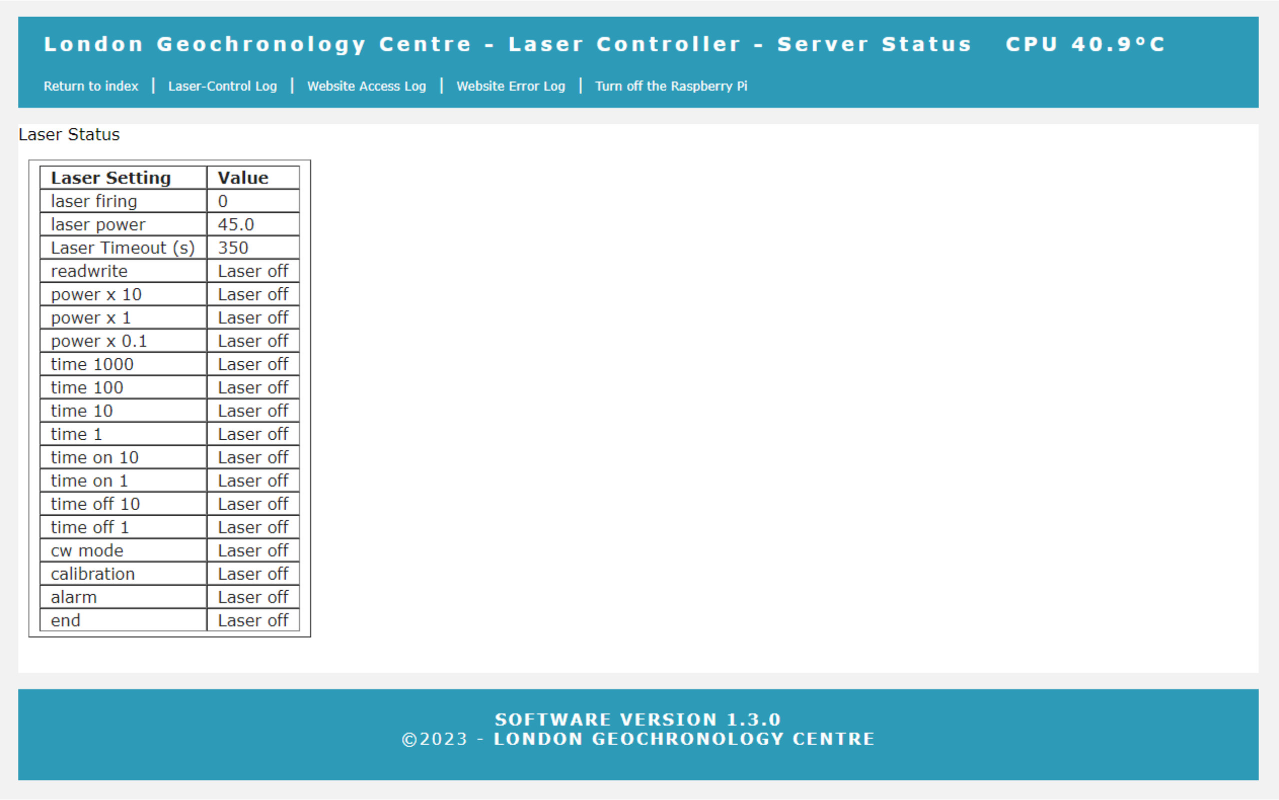


Figure 27: Schematic for Raspberry Pi connections and TTL buffer for Laser

Web interface

As well as accessing the status of the valves via the RESTful API a read-only web interface that can be accessed directly from a browser is will be available to view the valve status, application and web server logs.



London Geochronology Centre - Laser Controller - Server Status CPU 40.9°C

Return to index | Laser-Control Log | Website Access Log | Website Error Log | Turn off the Raspberry Pi

Laser Status

Laser Setting	Value
laser firing	0
laser power	45.0
Laser Timeout (s)	350
readwrite	Laser off
power x 10	Laser off
power x 1	Laser off
power x 0.1	Laser off
time 1000	Laser off
time 100	Laser off
time 10	Laser off
time 1	Laser off
time on 10	Laser off
time on 1	Laser off
time off 10	Laser off
time off 1	Laser off
cw mode	Laser off
calibration	Laser off
alarm	Laser off
end	Laser off

SOFTWARE VERSION 1.3.0
©2023 - LONDON GEOCHRONOLOGY CENTRE

Figure 28: Web status page

Interfaces

Laser TTL Cable specification

The connection from the back of the housing to the valves is a standard 9 pin D Socket

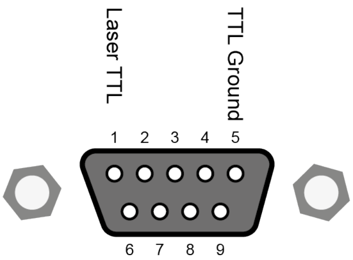


Figure 29: Laser TTL D Plug Assignment

A 9 way D plug linked to a female SMB mini coaxial connector with RG 174/U cable is required, pin 1 of the D plug should be connected the centre tap of the SMB connector.

9 Pin RS232 Cable specification

The connection from RS232 port a standard 9 pin D Socket, a null modem switching pins 2 and 3 (receive and transmit) is used to connect to the laser.

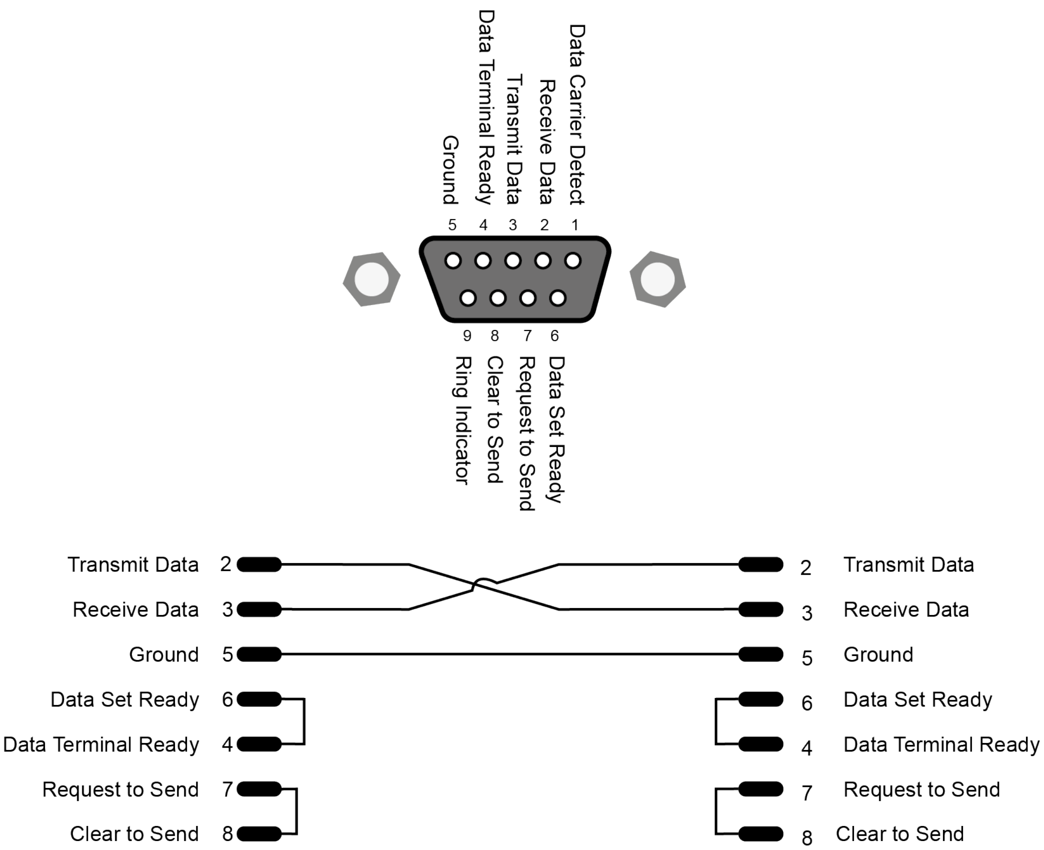


Figure 30: RS232 D Plug Assignment and null modem cable

Pump Reader Raspberry Pi Microservice

The Pump Reader control software consists of a python (Python Software Foundation, 2020) application to read the pressures from the Vacuum Gauges and a pressure gauge for the N_2 line that operates the vacuum valves. The application is written to use the Flask web interface to provide an HTTP Status page as well as serving requests via the API (Application Programming Interface)

The computer that reads the gauges is a Raspberry Pi 4B (Raspberry PI Foundation, 2020), it uses 4 x CH340 USB to RS232 controllers to access the gauges.

It has an internal scheduler that poles each gauge every 4 seconds with a query to read back the pressure.

The computer is accessed via a RESTful API listening on port 80 for a valid json message.

Query Messages

Json messages in the following formats are accepted:

Read Pressures:


```
{
    "item": "getpressures",
    "command": "read"
}
```

Output Messages

Following a “`getpressures`” command, the following data is returned:


```
[
    {
        "pressure": 6.7804e-08,
        "pump": "turbo",
        "units": "mBar"
    },
    {
        "pressure": 0.0074888,
        "pump": "tank",
        "units": " mBar "
    },
    {
        "pressure": 6.679200000000001e-09,
        "pump": "ion",
        "units": " mBar "
    },
    {
        "pressure": 4.9588,
        "pump": "gas",
        "units": "Bar"
    }
]
```


Pressures will be in floating point notation when connected to the gauge. When disconnected they will return zeros.

Web interface

As well as reading the status of the pumps and pyrometer via the RESTful API a read-only web interface that can be accessed directly from a browser and will be available to view the pump pressures, N_2 pressure, application and web server logs.

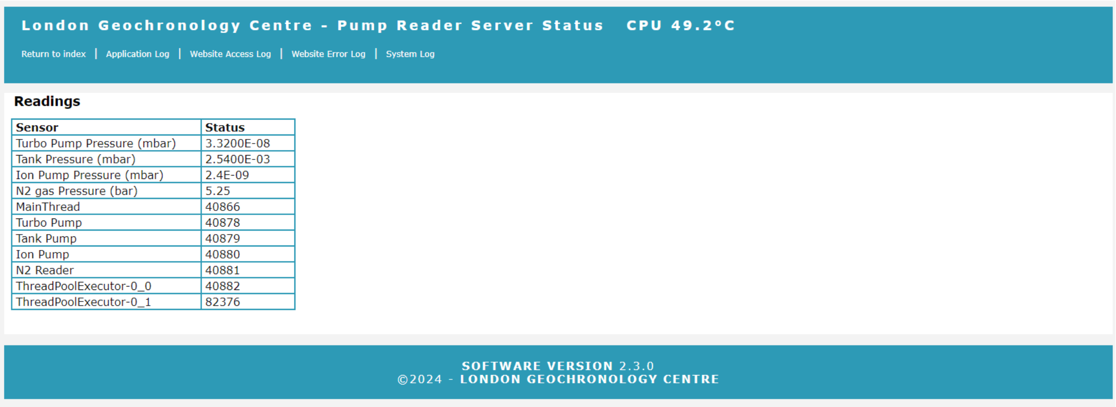


Figure 31: Web status page

Interfaces

9 Pin RS232 Cable specification

The connection from RS232 port a standard 9 pin D Socket, a null modem switching pins 2 and 3 (receive and transmit) is used to connect to the gauges and pyrometer.

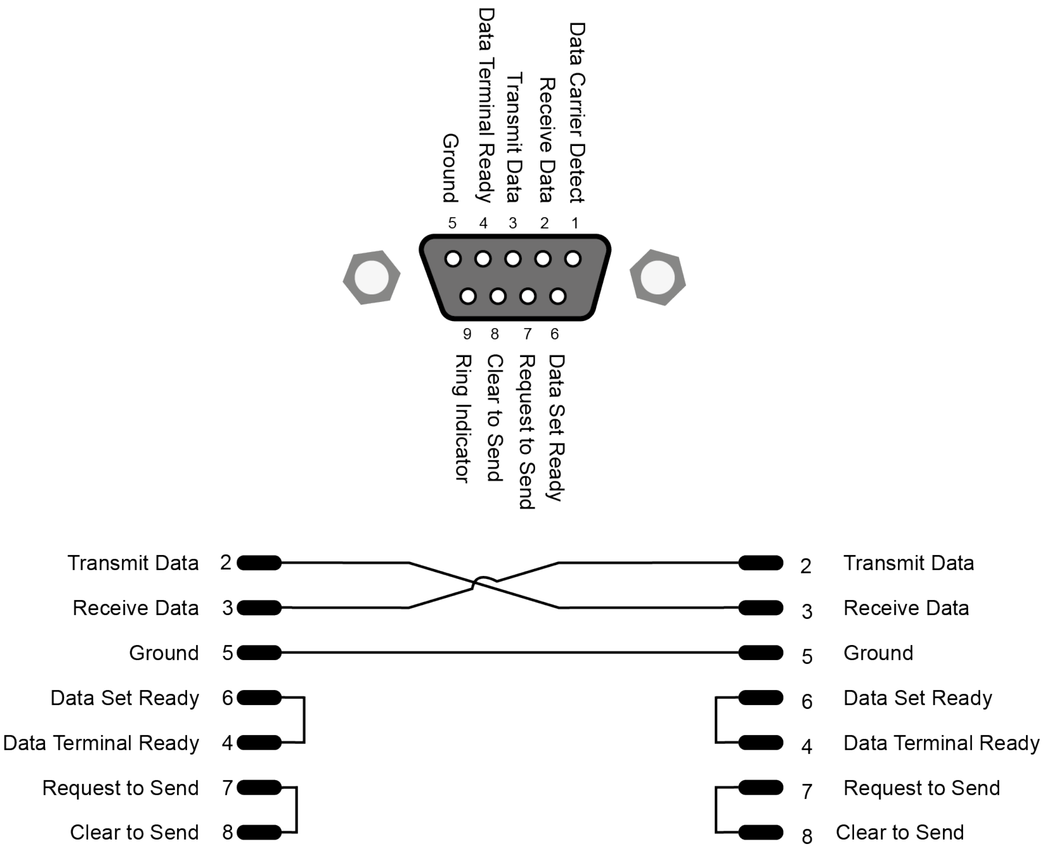


Figure 32: RS232 D Plug Assignment and null modem cable

Connections

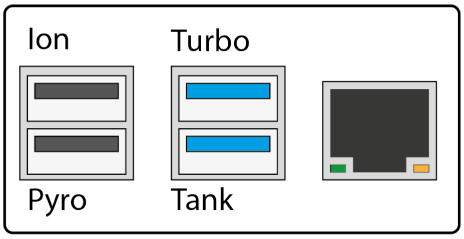


Figure 33: USB connections to the Pump Reader Raspberry Pi

Connection	USB	Serial Device	Equipment Connector
Turbo Pump	USB3 port 0	/dev/ttyUSB0	RS232 Plug (Male)
Tank Pump	USB3 port 1	/dev/ttyUSB1	RS232 Plug (Male)
Ion Pump	USB2 port 0	/dev/ttyUSB2	RS232 Socket (female)
N2 Guage	USB2 port 1		

Table 13: External Equipment Connections

Request Commands

Equipment	First Command (Hexadecimal)	Second Command (Hexadecimal)
Pfeiffer Vacuum TPG 1 (Tank and Turbo)	50 52 31 0D (Pressure Gauge 1)	05 (Enquiry)
Gamma Vacuum SPe (Ion Pump)	7E 20 32 35 20 30 42 20 30 30 0D (read pressure from ID 05)	

References

Python Software Foundation (2020) *Python 3 Programming Language*. Online. Available online: https://www.python.org [Accessed July 2020].

SQLite Consortium (2020) *SQLite Database*. Available online: https://sqlite.org/index.html [Accessed July 2020].

The Qt Company (2020) *Qt for Python (PySide2)*. Available online: https://wiki.qt.io/Qt_for_Python [Accessed July 2020].

Appendix 1 PyMS Task Sequences

Line Blank

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
10	valve13	close
11	valve11	open
12	valve10	close
15	valve6	open
16	valve7	open
17	valve8	open
20	valve3	open
56	valve3	close
57	valve8	close
59	valve4	open
73	valve4	close
79	valve3	open
81	valve10	open
105	valve3	close
315	quad	hiden-startmid
404	valve6	close
420	valve8	open
435	quad	starttimer
722	quad	writefile
724	image	hiden-mid
725	quad	hiden-stop
735	quad	hiden-startprofile
850	image	hiden-profile
855	quad	hiden-stop
856	valve6	open
859	xytable	move
946	valve3	open
1038	valve3	close
1239	xytable	move
1300	quad	check-stopped
1339	end	end

Q-Standard

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
10	valve13	close
11	valve11	open
12	valve10	close
15	valve6	open
16	valve7	open
17	valve8	open
17	valve3	open
19	valve2	close
20	valve2	open
20	valve3	close
57	valve1	open
59	valve4	open
72	valve1	close
73	valve4	close
75	valve8	close
77	valve2	open
79	valve3	open
81	valve10	open
104	valve2	close
105	valve3	close
315	quad	hiden-startmid
404	valve6	close
420	valve8	open
435	quad	starttimer
722	quad	writefile
724	image	hiden-mid
725	quad	hiden-stop
735	quad	hiden-startprofile
850	image	hiden-profile
855	quad	hiden-stop
856	valve6	open
859	xytable	move
946	valve2	open
948	valve3	open
976	valve2	close
978	valve3	close
1239	xytable	move
1300	quad	check-stopped
1339	end	end

Apatite (No Reheat)

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
7	laser	checkalarms
8	image	dynolite
9	laser	setpower
10	valve13	close
11	valve11	open
12	valve10	close
13	laser	on
15	valve6	open
16	valve7	open
17	valve8	open
18	valve3	open
56	valve3	close
59	valve4	open
61	image	microscope
73	valve4	close
75	valve8	close
79	valve3	open
81	valve10	open
105	valve3	close
223	laser	off
315	quad	hiden-startmid
404	valve6	close
420	valve8	open
435	quad	starttimer
722	quad	writefile
724	image	hiden-mid
725	quad	hiden-stop
735	quad	hiden-startprofile
850	image	hiden-profile
855	quad	hiden-stop
856	valve6	open
859	xytable	move
946	valve3	open
1038	valve3	close
1239	xytable	move
1300	quad	check-stopped
1339	end	end

Apatite + Reheat

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
7	laser	checkalarms
8	image	dynolite
9	laser	setpower
10	valve13	close
11	valve11	open
12	valve10	close
13	laser	on
15	valve6	open
16	valve7	open
17	valve8	open
18	valve3	open
56	valve3	close
59	valve4	open
61	image	microscope
73	valve4	close
75	valve8	close
79	valve3	open
81	valve10	open
105	valve3	close
283	laser	off
315	quad	hiden-startmid
404	valve6	close
420	valve8	open
435	quad	starttimer
722	quad	writefile
724	image	hiden-mid
725	quad	hiden-stop
735	quad	hiden-startprofile
850	image	hiden-profile
855	quad	hiden-stop
856	valve6	open
946	valve3	open
1038	valve3	close
1100	quad	check-stopped
1246	valve10	close
1252	laser	on
1253	valve8	open
1256	valve3	open
1295	valve3	close
1298	valve4	open
1300	image	microscope-reheat
1312	valve4	close
1314	valve8	close
1318	valve3	open
1320	valve10	open
1344	valve3	close
1522	laser	off
1554	quad	hiden-startmid
1643	valve6	close
1659	valve8	open
1674	quad	starttimer-reheat
2005	quad	writefile
2007	image	hiden-mid-reheat
2008	quad	hiden-stop
2019	valve6	open
2022	xytable	move
2109	valve3	open
2201	valve3	close

Time (s)	Object	Command
2402	xytable	move
2430	quad	check-stopped
2502	end	end

Unload

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
7	xytable	move
10	valve13	close
11	valve11	close
12	valve10	close
15	valve6	open
16	valve7	open
17	valve8	close
20	end	end

Pump

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
10	valve13	close
11	valve11	open
12	valve10	open
15	valve6	open
16	valve7	open
17	valve8	close
20	valve13	open
7130	valve13	close
7140	valve8	open
7200	end	end

Line Clean

Time (s)	Object	Command
1	valve12	open
2	valve1	close
3	valve2	close
4	valve3	close
5	valve4	close
6	valve5	close
10	valve13	close
11	valve11	open
12	valve10	open
15	valve6	open
15	valve13	open
16	valve7	open
17	valve8	close
1500	valve13	close
1502	valve8	open
1502	valve2	open
1592	valve2	close
1602	valve3	open
1692	valve3	close
1807	valve10	close
1857	valve8	close
1859	valve4	open
1873	valve4	close
1879	valve3	open
1881	valve10	open
1905	valve3	close
2115	quad	hiden-startmid
2204	valve6	close
2220	valve8	open
2235	quad	starttimer
2522	quad	writefile
2523	image	hiden-mid
2525	quad	hiden-stop
2535	quad	hiden-startprofile
2650	image	hiden-profile
2655	quad	hiden-stop
2656	valve6	open
2659	xytable	move
2746	valve3	open
2838	valve3	close
3039	xytable	move
3100	quad	check-stopped
3139	end	end