

Quickstart Guide

Microscope startup and initialisation

1. Boot the software by running the `interface_run_me.py` script. **ENSURE the power to the motors is OFF.**
2. Once provided a `Enter command` prompt, turn on the motor power at the controller switch.
3. Type `enable` to begin first phase of laser warmup
4. `homelaser` to home all laser motors
5. `homegratings` to home all grating motors
6. Use `sall λ` to go to a specific wavelength.
 - NOTE: A go-to command is required after homing any motor.
 - Homing motors brings them to the zero steps position. All motors at zero does not equate to any calibrated wavelength, so a go-to command is required to realign to a chosen wavelength.
7. If not yet done, open or cycle the shutter `cycleshutter` or `shutteron`
8. Confirm the laser is lasing at the desing wavelength. At the desing wavelength, the laser will make it to the pinhole and should pass through the rest of the microscope to the sample.
- **Troubleshooting:** If the laser does not lase properly, or is weak, sometimes the I2 motor is not where it should be after homing (this is not common but can happen). Rehome just I2 with `home I2` and follow up with `sI λ`. If this does not fix it, manually jog the motor until lasing is restored. Find the motor position that centres the alignment on the lasing condition, then use `reference λ` to reference this set of motor positions to this wavelength.
9. Ensure lasing at the sample (sulfur test).
10. Type `run` to initiate the continuous acquisition for focus and alignment. NOTE: The `data_viewer_run_me.py` should be running in a separate window. If not, go to the RamanMicroscope code folder and run this file by double clicking.
11. In the data viewer UI, ensure the ROI start and end are set to a logical range. Usually 100-2000 respectively. Toggle autoscale as required to view the spectrum and lock/normalise the intensity range.
12. Also ensure the Y centre and Y width are set to capture the data. This is the binning on the CCD array, and the data should be centred around pixel 83 with ~10 pixel range. On rare occasions, this can change, so if no peaks are present try widening the Y width value (and click Apply Y ROI)
13. Upon running, the data viewer should update with live data each frame. The sulfur spectrum should be just barely visible at 1 second acquisition time if you're lucky. Not not:
 1. Check the sample is in focus
 2. **the motors may need to be jogged into alignment**

Jogging motors:

After homing, sometimes the motors can be slightly off still. I believe this is a consequence of inaccuracy in the limit switch positions and the absolute positioning of the grating motor drivers. The fix is to massage the setup into its final alignment.

1. start with motor G1. You can place a target card in the cage system near the 45 degree vertical turn (just after the beamsplitters). The laser should pass dead centre through the target card. If not, use `g1`

1. x to move the motor x steps. Do this until it passes through the target card centre.
2. Remove the target card.
3. Next, work through the other motors by observing changes to the spectrum.
 - g2, g3, g4
 - The intensity should reach a maximum within 1-3 steps of thier current positions. Find the maximum value for the motor, then move on to the next.
 - Once all motors have reached their maximum intensities, you can cycle back and repeat once more.
 - Once you are satisfied with the intensities, reference this position (reference λ) and move on.

The microscope should now be aligned and calibrated, ready for scans.

Running a Scan

Control Notes

The primary form of interaction with the microscope is via the command line interface. This will change over time to a Graphical User Interface, but the command line interface will always be accessible for debugging and convenience.

How To Use Commands

All commands can be passed arguments in the form of characters separated by `<space>` keys. For instance, `s1` is the "go to laser wavelength command", and it requires a laser wavelength as a number for its argument. The command to move to 750 nm laser excitation is `s1 750`.

Frequent Commands

- `wai`: "where am I" prints the current motor positions and calculates the laser and grating wavelengths using the calibrations
- `s1 $lambda`: set laser wavelength to specified value (e.g. `s1 800` for 800 nm)
- `sg $lambda`: set wavelength on all gratings (microscope filter and double monochromator)
- `sm $lambda`: set wavelength on just the monochromator gratings. Useful for moving the detection closer to the laser line (for seeing low-frequency modes) or further from the laser line (to reduce stray light and Rayleigh scattering signal)
- `st $lambda`: set TIRAX spectrometer wavelength
- `sall $lambda`: set all wavelengths (for changing the whole setup to a specific wavelength)
- `homelaser`: homes the three laser motors in sequence
- `homegratings`: homes all grating motors in sequence
- `home $motorlabel`: homes the motor of \$motorlabel\$ (e.g. `home 11`)

- **reference** λ : reassigns the current motor positions to the wavelength of choice. Used when making minor adjustments to the alignment of the laser or gratings.
- **run**: runs the camera in continuous acquisition mode. Data is plotted to the data_viewer_run_me.py - **stop**: stops a continuous acquisition.
- **filename** $string$: sets the current filename ($string$) for saved data
- **gui**: brings up the GUI for preparing and performing multidimensional scan datasets
- **enable**: cycles through the startup sequences of the laser. Repeated calls to **enable** progress the startup sequence:
 1. Checks the warmup start and initiates diode warmup
 2. Enables the laser at non-lasing power (0.05 W) for second warmup.
 3. Turns the laser to 4.0 W - laser is fully on, but shutter is closed still
- **shutteron**: opens the laser shutter
- **shutteroff**: closes the shutter
- **cycleshutter**: closes then reopens the shutter. Useful for startup because the shutter sometimes doesn't open when it should

Individual Motor Motion

Sometimes you need to jog the motors a small number of steps, for alignment or for calibration. This is done by typing the motor label and the number of steps you want to move (relative motion not absolute), either positive or negative direction.

- **l1** x : move laser motor 1 x steps
- **l2** x : move laser motor 2 x steps
- **l3** x : move laser motor 3 x steps
- **g1** x : move grating motor 1 x steps
- **g2** x : move grating motor 2 x steps
- **g3** x : move grating motor 3 x steps
- **g4** x : move grating motor 4 x steps

Additional Commands

Many of these commands are required for maintenance but some have calibration-breaking consequences (not permanent but annoying). Only use commands you are familiar with and know what the consequences are.

System Control

- **report**: Show current system status (pass **initialise=True** to refresh all parameters).
- **loadconfig**: Load system configuration.
- **gui: Launch GUI interface (if available).**
- **cancel**: Cancel current scan.
- **runscan**: Run scan using current acquisition parameters.

- `refresh`: Refresh camera connection.
 - `closecamera`: Fully close camera interface (for external GUI takeover).
 - `allocate`: Allocate buffer for camera acquisition
 - `deallocate`: Deallocate camera buffer and stop acquisition.
-

Motor Homing and Position

- `home(motor_label)`: Home specific motor, e.g., '1X', '2Y'.
 - `homeall`: Home all motors.
 - `homelaser`: Home laser motors.
 - `homemono`: Home monochromator motors.
 - `homegratings`: Home grating motors.
 - `stagehome`: Set current XY(Z) stage position as home.
 - `testhoming`: Cycle home movement to test reproducibility.
-

Motor Positioning (Steps)

- `slsteps(target_dict)`: Move laser motors to target steps (e.g., {'l1': 1000}).
 - `smsteps(target_dict)`: Move monochromator motors to target steps.
 - `sgsteps(target_dict)`: Move grating motors to target steps.
 - `allmotors`: Print current step positions of all motors.
 - `laserpos`: Get laser motor positions.
 - `monopos`: Get monochromator/grating motor positions.
 - `rg`: Get current spectrometer motor step position.
-

Wavelength-Based Motion

- `sl(wavelength_nm)`: Move laser to specified wavelength.
 - `sm(wavelength_nm)`: Move monochromator to specified wavelength.
 - `sg(wavelength_nm)`: Move all gratings to specified wavelength.
 - `st(wavelength_nm)`: Move spectrometer to specified wavelength.
 - `sall(wavelength_nm, shift=True/False)`: Move all components; `shift=True` maintains Raman shift.
-

Calibration & Referencing

- `reference(wavelength_nm, shift=True)`: Reference system to known laser wavelength.
- `referencetriax(step_pos, pixel_pos)`: Reference spectrometer step to a CCD pixel (manual).

- `invertcal`: Invert calibration mappings (laser & monochromator).
 - `recmot`: Save current motor positions.
 - `writemotors(label, pos_dict=None)`: Write positions with optional manual override.
 - `calhome`: (TBD)
-

Camera Control

- `camera`: Connect to camera.
 - `camclose`: Disconnect hardware interface.
 - `caminfo`: Print camera info.
 - `temp`: Get camera temperature.
 - `roi(x, y, width, height)`: Set region of interest.
 - `setbin(binning_level)`: Set pixel binning.
 - `setgain(value)`: Set camera gain.
 - `camspec`: Switch to spectrum mode.
 - `camimage`: Switch to image mode.
 - `acquire`: Acquire a single frame (if implemented).
 - `run`: Start continuous acquisition.
 - `stop`: Stop continuous acquisition.
-

Stage Control

- `x(um), y(um), z(um)`: Move stage in micrometers along X, Y, Z.
 - `stagehome`: Set current XYZ position as home.
-

Misc Functions

- `rldr`: Read light-dependent resistor for laser detection (used in autocal).
- `autocal`: (TBD)
- `mshut`: Close mechanical shutter.
- `mopen`: Open mechanical shutter.
- `ramanmode`: Set system to Raman acquisition mode.
- `imagemode`: Set to widefield image acquisition mode.

- `polin(angle)`, `polout(angle)`: Set polarizer angle (entry/output).
- `wai`: (TBD)
- `wavelengthaxis`: (TBD)
- `acqtime`, `filename`, `ramanshift`, `laserpower`: (TBD – placeholder or legacy)