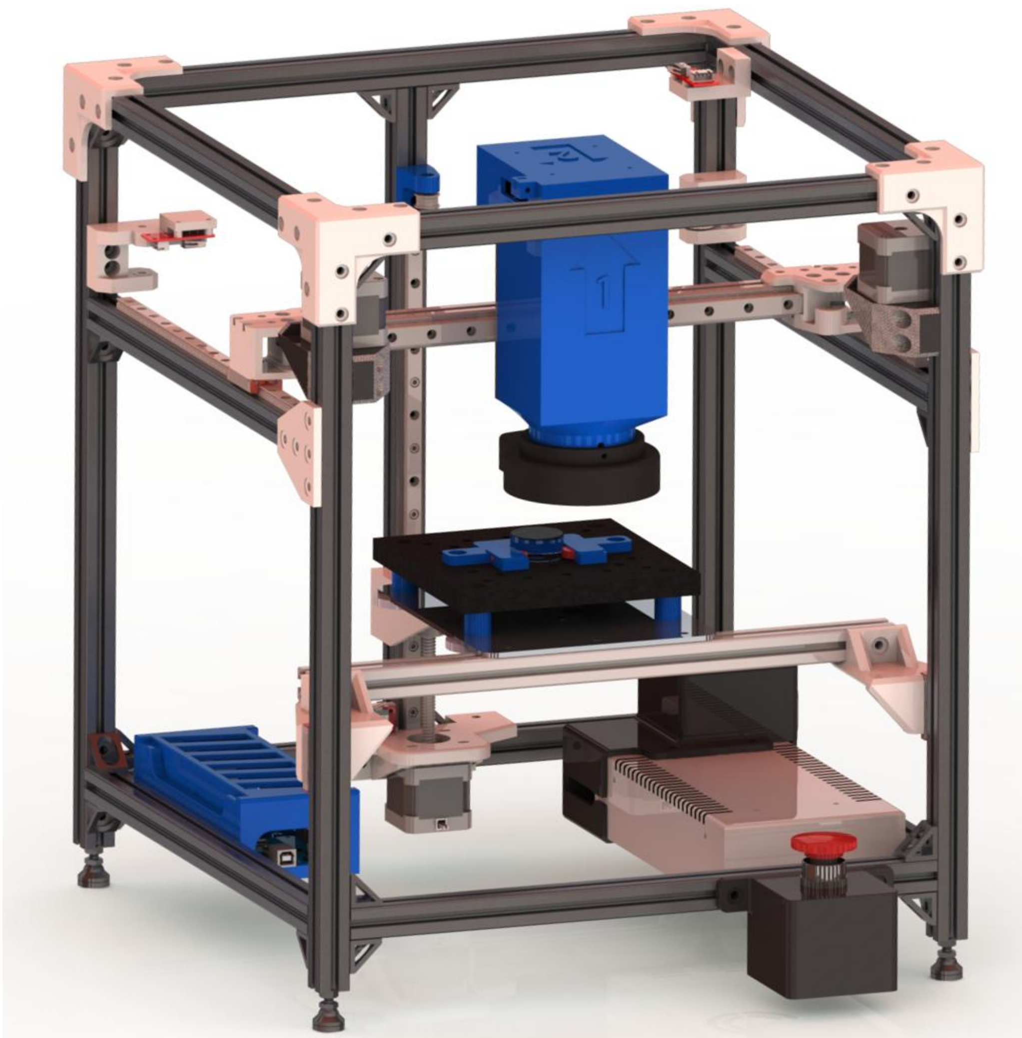
MANUAL

Optical Module



Contents

1. Introduction

This document is designed to explain how to use and maintain the functionality of the optical module. The optical module was designed by a group of manufacturing engineering students at the University of British Columbia (UBC) for their capstone project as prescribed by Professor Ben Britton and PhD student Graeme Francolini.

The optical module is an automated optical microscope designed to function within Prof. Britton's self-driving lab for investigating metallurgical samples. The project is open-source and the GitHub repository for the project can be found here: https://github.com/ManuCapstone1/OpticalModule. For CAD modeling, BOM, STL files, design documents, assembly documents and all pertaining programming files please visit the GitHub repository.

This document is solely designed for instructions on calibration, maintenance, and how to use the optical module.

2. Operation Instructions

2.1 Quick Guide

The default procedure of operation of the module is the following:

1. Start the GUI program

2. Click "Create a new sample" button

3. Enter sample parameters requested in the window (Figure 1):

   A) Select one of the mount types. Initially, available mount types are "Puck" and "Stub".

   B) Enter sample name.

   C) Enter the starting sample height. This parameter defines how tall the sample is. The more accurate the measurement is, the better the quality of autofocus is going to be.

D) Enter sample layer height. This affects the height change after each round of polishing to reduce the time it takes to autofocus.

E) Enter the bounding box for the sample. This is used to define the sample size inside of a puck or on a stub.

F) Click “Ok” to complete the entry or “Cancel” to close the window without saving the new sample parameters.

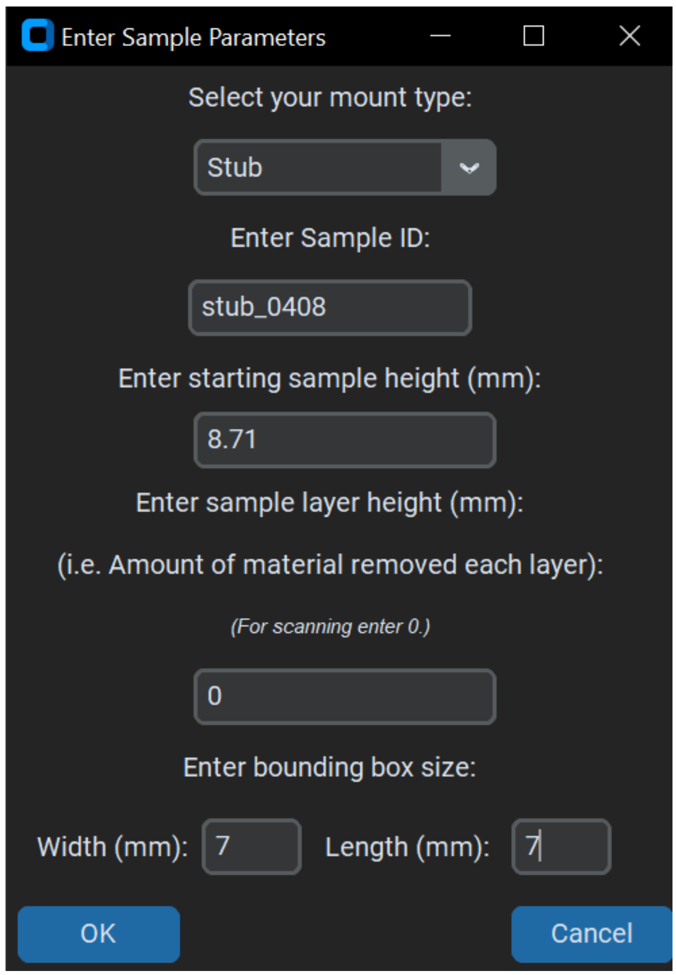


Figure 1 - Sample parameters pop-up window.

4. Select the desired mode of operation and click either “Random Sampling” or “Scanning”. Check the “Random Image Sampling” or “Scanning” sections of the manual for more information.

5. Enter requested parameters for each of the modes and wait until the camera is done taking pictures.

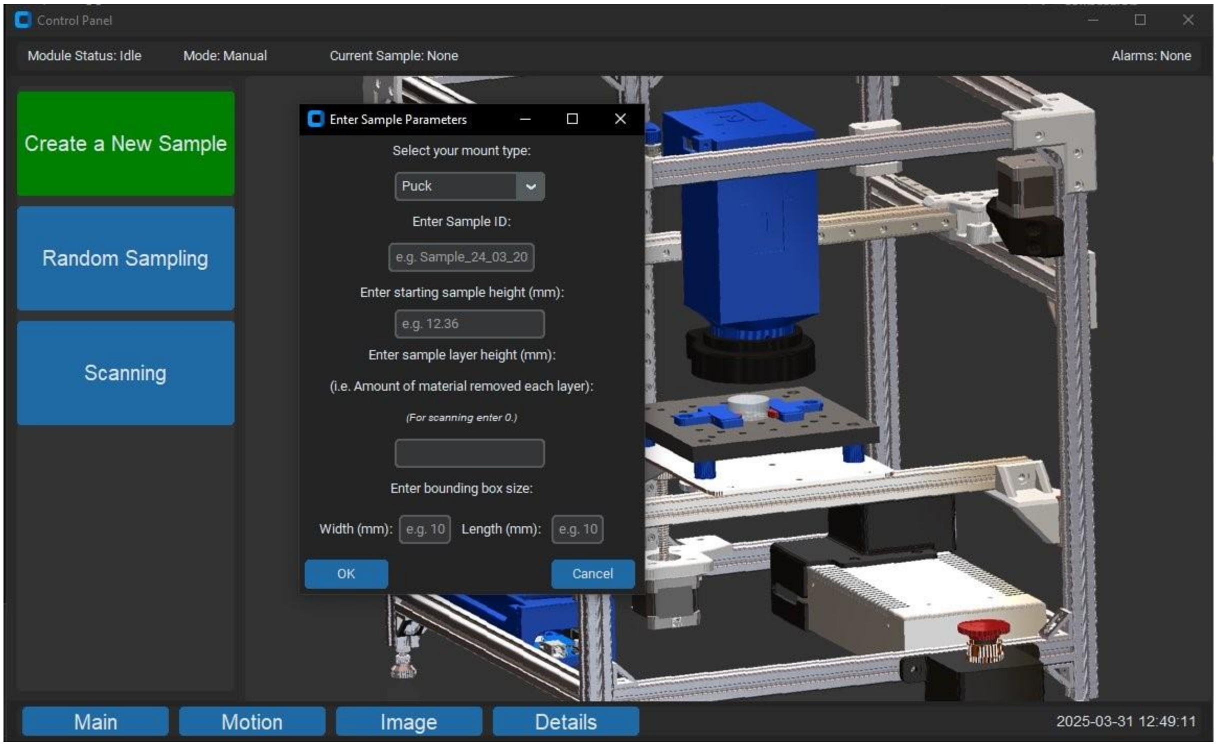


Figure 2 - Image sample parameters pop-up window.

2.2 Random Image Sampling

1. Once the “Random Sampling’ button is clicked, a pop-up window prompting for the number of required images will appear. Enter the number of images you want taken.

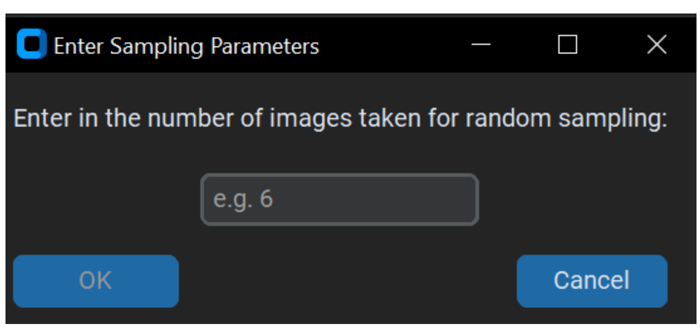


Figure 3 - Random sampling pop-up window.

2. Then, a new loading screen in the right part of the window will appear. It indicates that the camera is currently taking images. You can press “Stop” button to interrupt the process at any point. This will require re-homing the system before you can restart the task.

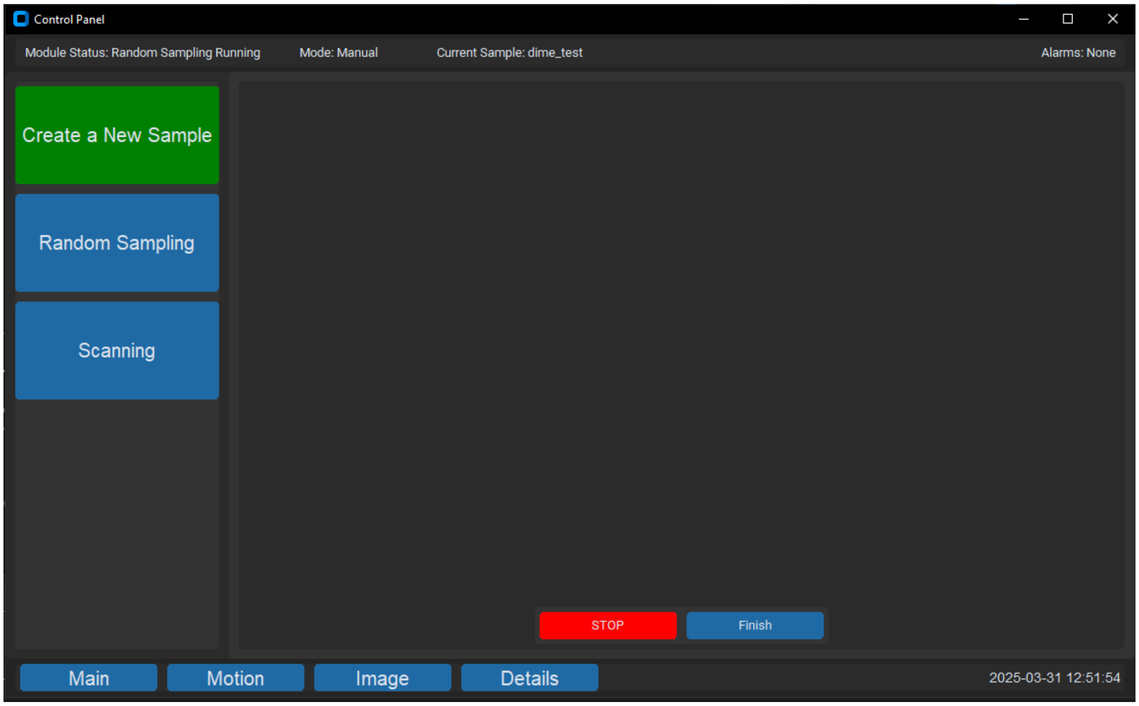


Figure 4 - Random sampling mode loading screen.

3. After the module is done taking images, a new screen with all images will be displayed in the right part of the window. The images are stored in the directory specified on the “Details” tab of GUI.

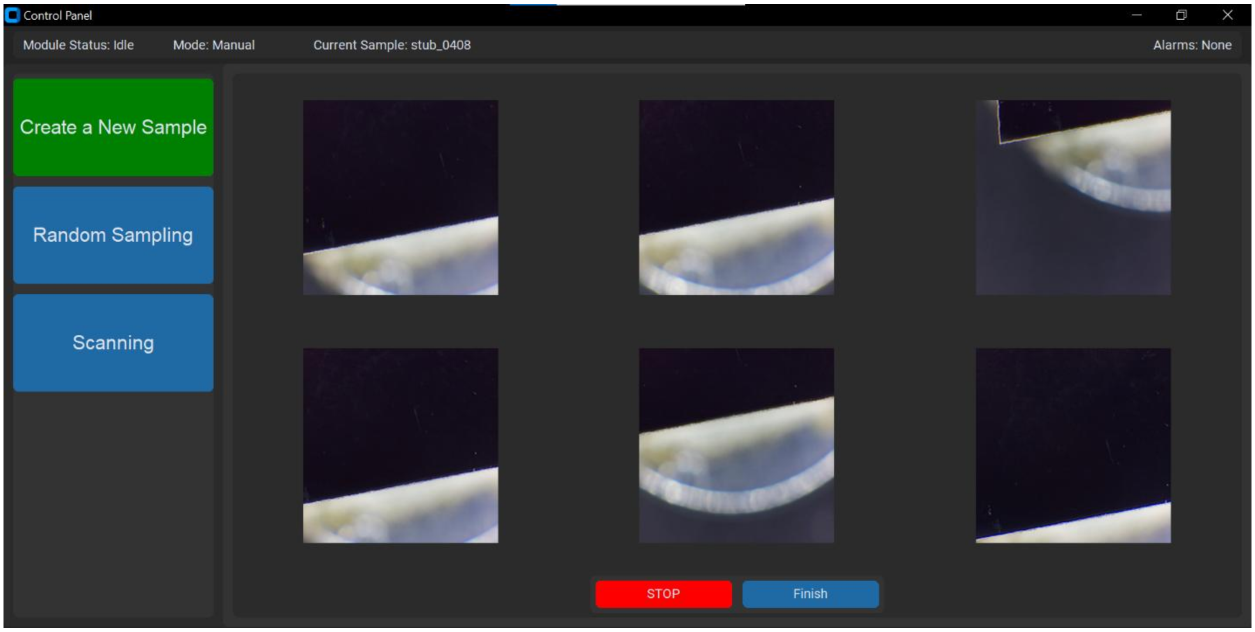


Figure 5 - Random sampling window, after the process has been completed.

4. You can expand each image by clicking on it. A new window with the image will appear. Only one image can be expanded at a time.

5. Click “Finish” to close the “Random Sampling” window. The “Finish” button will also relocate the files to their permanent location as specified in the “Details” tab of the GUI, labelled with the sample ID and time stamp.

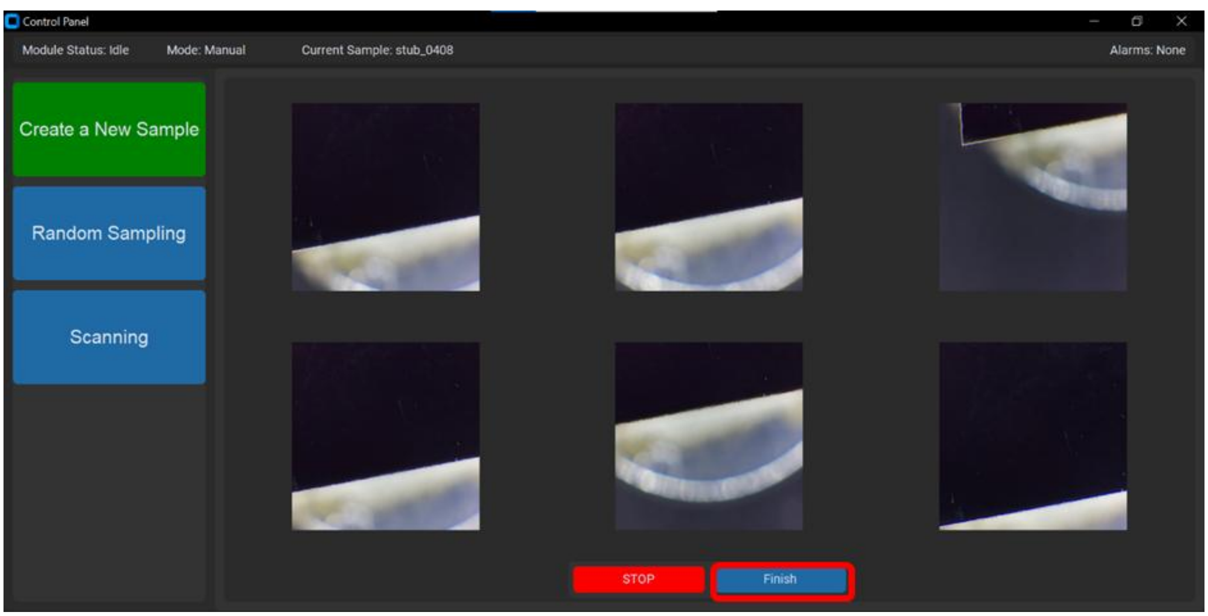


Figure 6 - Completed Random Sampling Screen with highlighted “Finish” button.

2.3 Scanning

1. Once the “Scanning’ button is clicked, a pop-up window prompting for the sample length and width. This would describe the bounding box around the sample to calculate the number of images to be taken. Recommended parameters are 5 for X direction and 4 for Y direction.

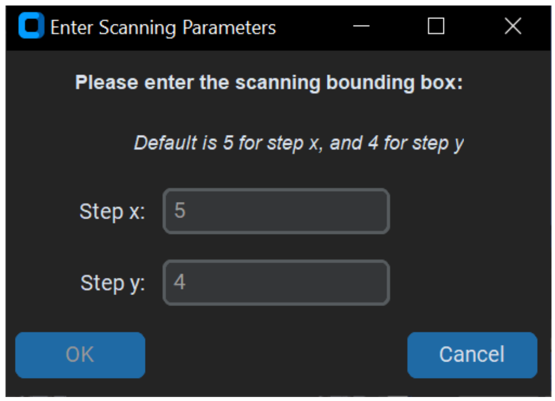


Figure 7 - Scanning parameters pop-up window.

2. Then, a new loading screen in the right part of the window will appear. It indicates that the camera is currently taking images. You can press “Stop” button to interrupt the process at any point. This will require re-homing the system before you can restart the task.

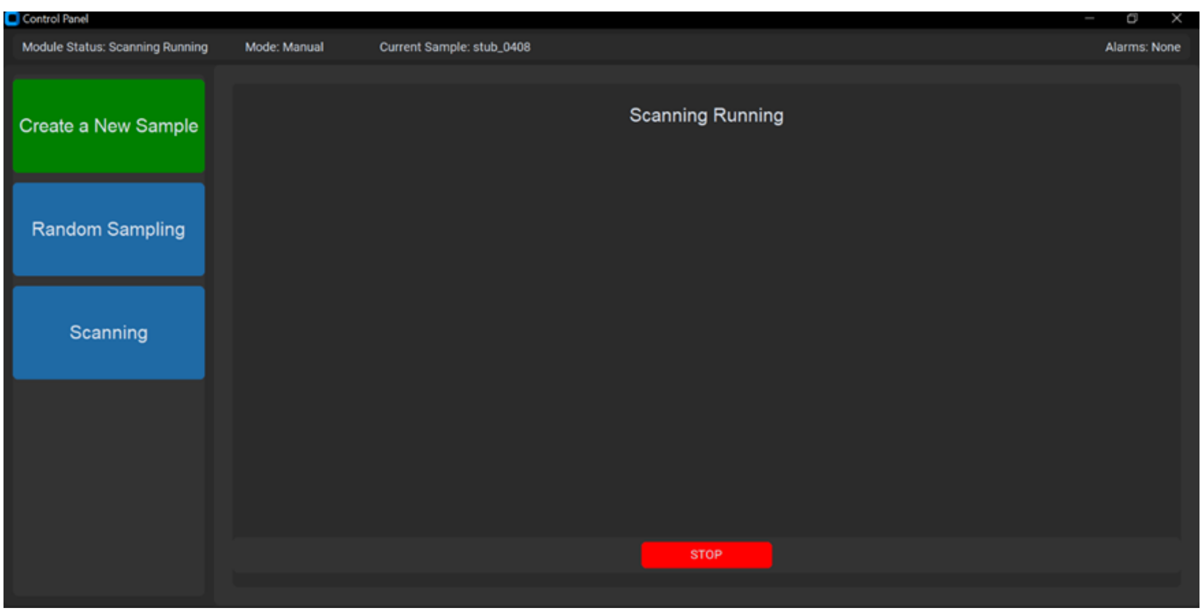


Figure 8 - Screenshot of GUI when scanning is in the process of running.

3. After the module is done taking images, a new screen with all images will be displayed in the right part of the window. The images are temporarily stored in the “stitching” folder directory specified on the “Details” tab of GUI.

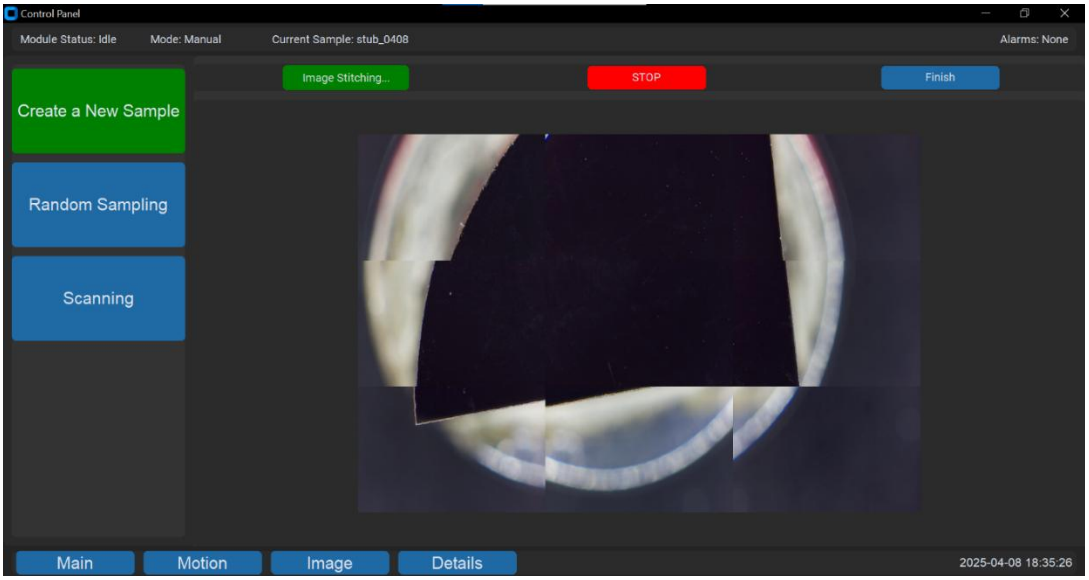


Figure 9 - Image of stitching when the images have been taken.

4. You can expand each image by clicking on it. A new window with the image will appear. Only one image can be expanded at a time.

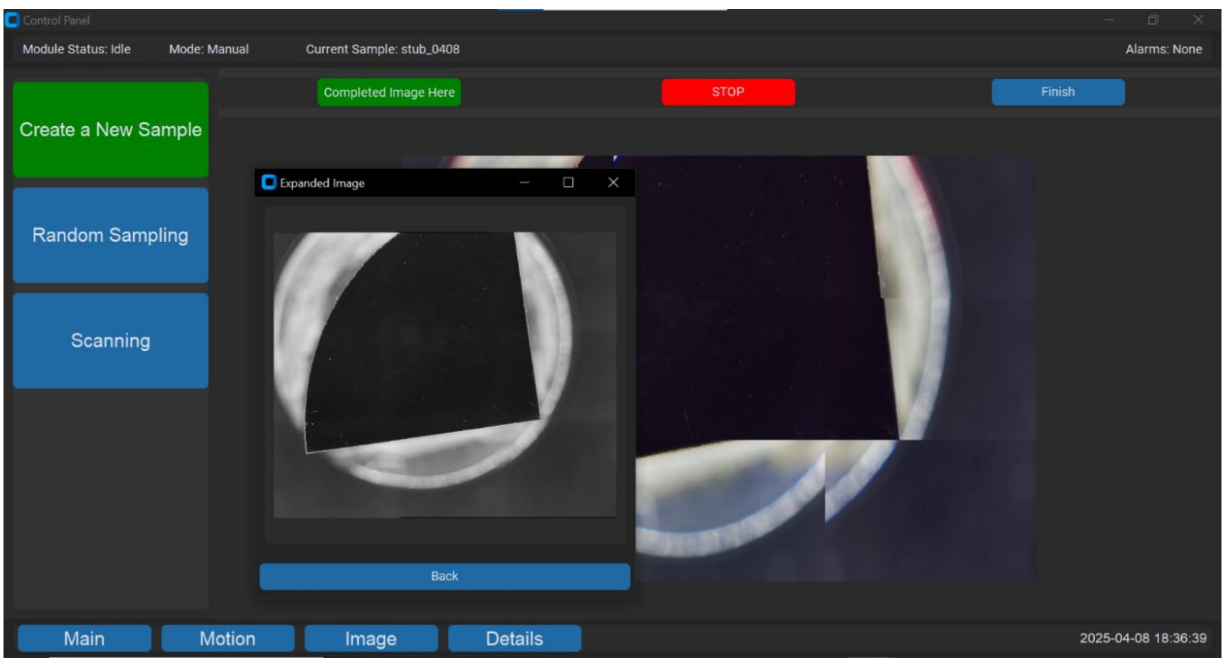


Figure 10 - Scanning frame with an image expanded.

5. Once the software had finished stitching the images together, a "Complete Image Here" button will appear at the top left of the screen. It will display the complete stitched image in a separate window that can be expanded. You can also access the stitched image in the "stitching" folder as specified in the "Details" tab of the GUI.

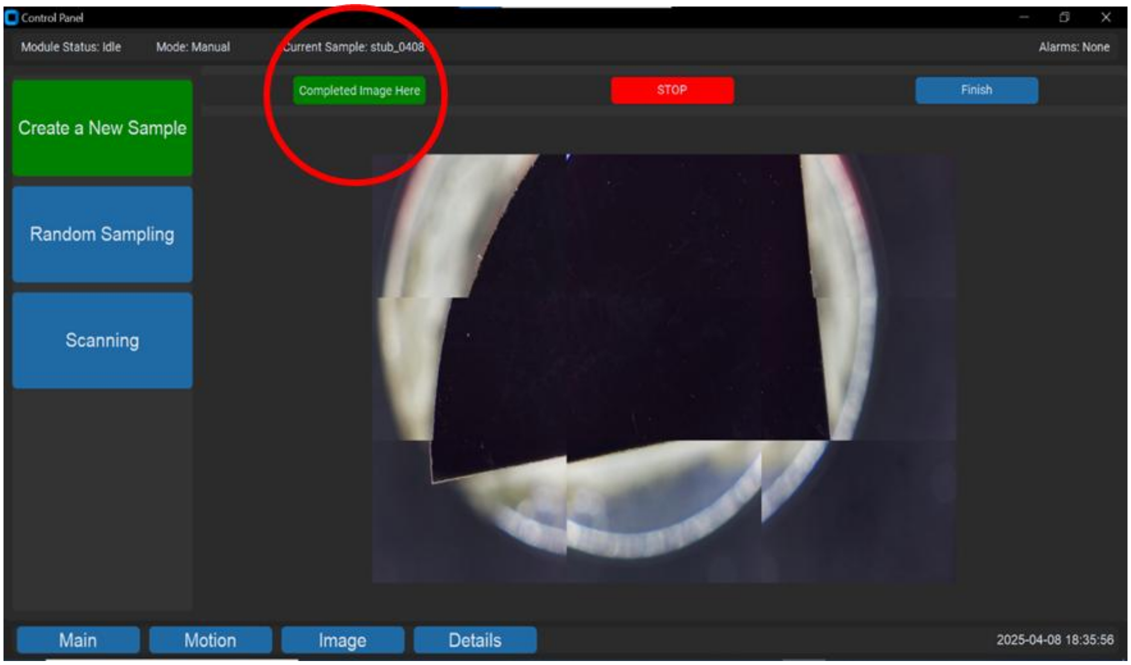


Figure 11 - Scanning completed, stitched image location circled in red.

6. Click "Finish" to close the "Scanning" window. Similar to Random Sampling, when the "Finish" button is pressed, the PC will relocate the files to it's permanent location as specified in the "Details" tab of the GUI, labelled with the sample id and time stamp.

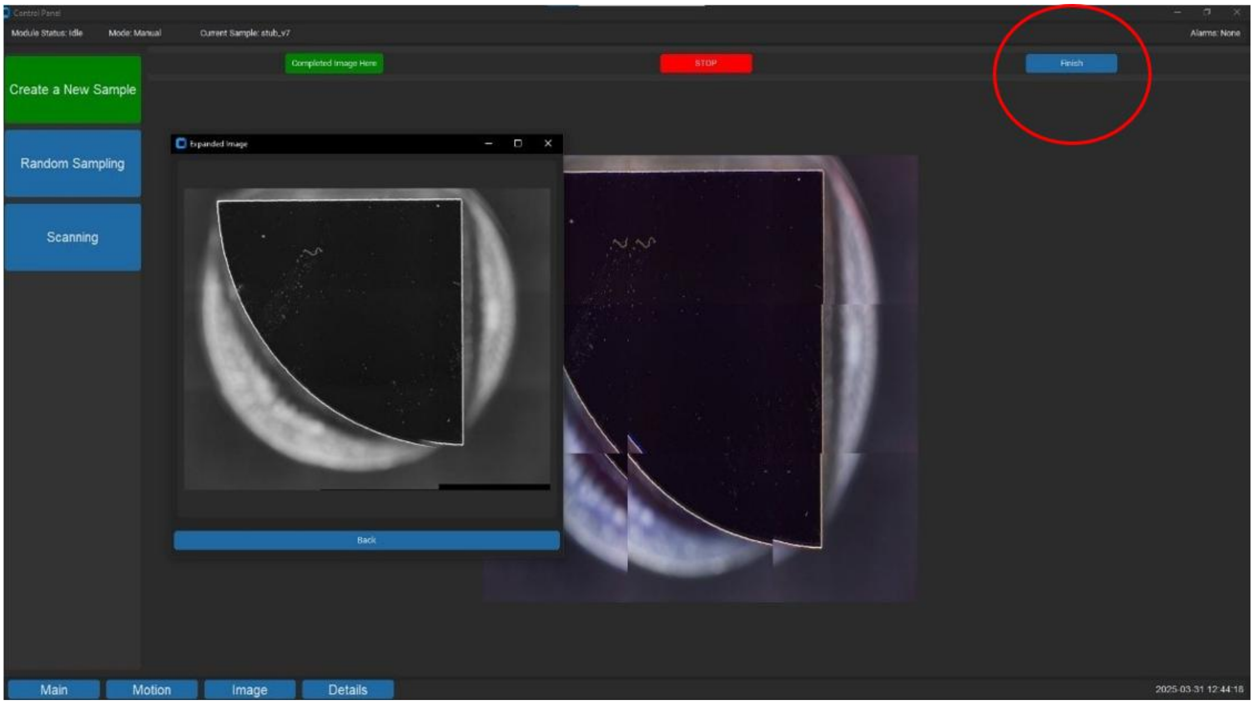


Figure 12 - Completed scanning process, with stitched image expanded.

2.4 Platform Calibration Routine

In order to ensure that all regions of the captured image remain consistently in focus, it is essential that the platform is properly levelled. Variations in height across the surface can lead to blurred or unfocused areas in the final image, especially at the corners.

Manual Platform Leveling:

The platform is supported at each of its four corners by height-adjustable knobs. These knobs allow users to manually tune the z-height of each corner to achieve a level imaging plane. The process involves incrementally adjusting each knob while observing the focus at the corresponding corner.

Autofocus Support via Script:

Although an automatic "Calibrate Platform" button is not yet implemented in the GUI, the foundation for this feature exists in the opticalModule.py script. A skeleton function has been developed that performs the following steps:

- Moves the camera to each corner of the platform.
- Executes an autofocus routine centered on a crosshair marker in the field of view.
- Records the z-height that achieves the best focus score at each corner.

This data can be used to guide manual adjustments. By comparing the optimal focus height at each corner, users can determine which knobs to raise or lower to bring all corners into alignment.

Camera Angle Considerations:

Ensure that the camera is mounted squarely to the platform. A tilted camera can lead to uneven focus across the image, even if the platform itself is level.

Potential Future GUI Integration:

A dedicated "Calibrate Platform" button was added to the GUI and is currently commented out in the gui.py script (see GitHub). This feature was initially planned to run the autofocus routine for each corner automatically and display the corresponding z-heights in the GUI.

2.5 Other GUI Tabs

2.5.1 Motion Tab

Motion Tab provides the user with the capability to adjust the position of the carriage. Before you start any action on the motion tab, make sure that the system has been homed and press “Refresh Coordinates” button. This will display the most recent position of the carriage.

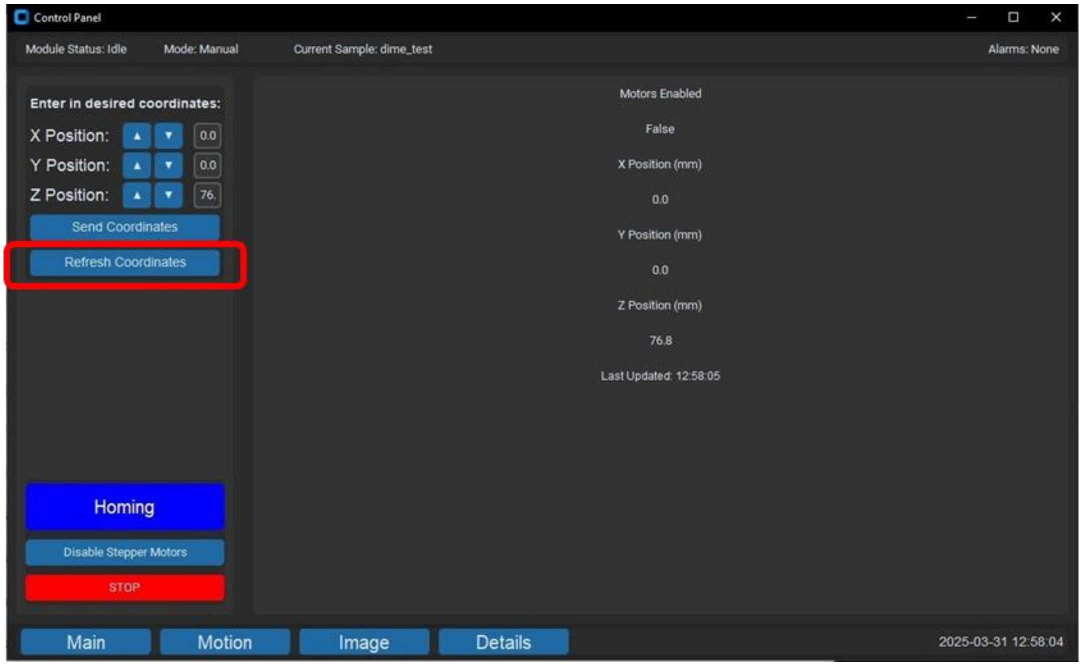


Figure 13 - "Refresh Coordinates" button on the Motion Tab.

- If the module has been reset or if you want to prepare it for future operations, you can home it without initiating any of the available mode by clicking "Homing" button. A pop-up window will appear prompting whether to home all 3 axes or just the X and Y axes.

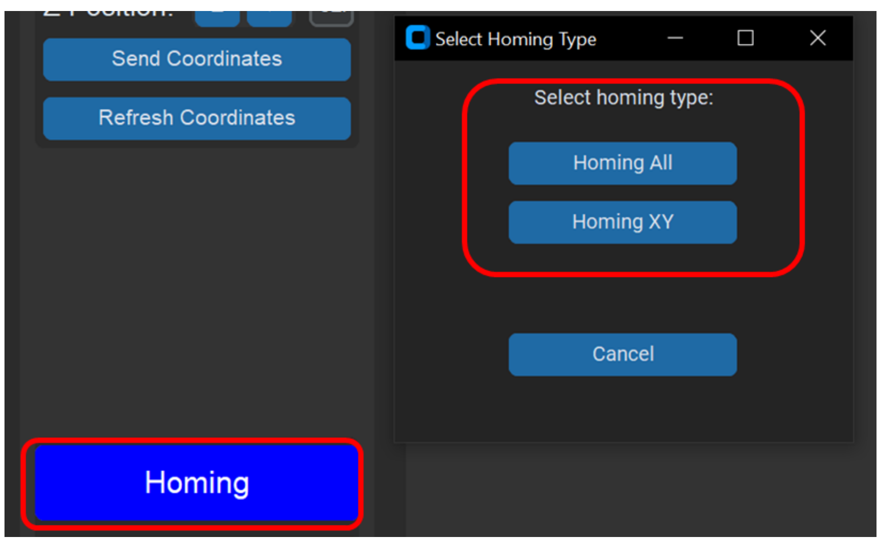


Figure 14 - Homing button and "Select Homing Type" pop-up window.

- You can place the carriage in a certain position by entering the coordinates in the top left entry boxes and pressing "Send Coordinates" button.

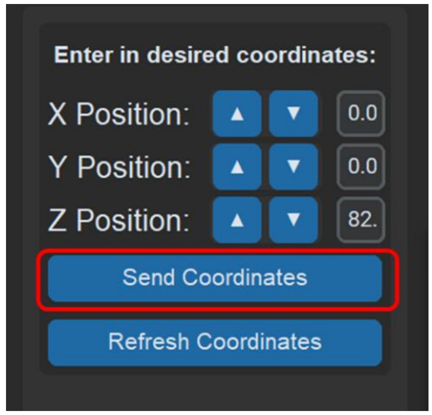


Figure 15 - "Send Coordinates" button on Motion Tab.

- You can also position the carriage in a particular location by pressing the "Disable Stepper Motors" button and manually dragging the carriage to the desired position.
  - *WARNING* At the time of completion of the project, only two out of three motors (X and Y) are disabled by pressing the button. Use CAUTION while moving the carriage manually.

Figure 16 - "Disable Stepper Motors" button on the Motion Tab.

- You can press "Stop" button to interrupt any process at any point. This will require re-homing the system before you can restart the task.

Figure 17 - "Stop" button on the Motion Tab to abort any operations.

2.5.2 Image Tab

Image Tab provides information on the current parameters used for the Raspberry Pi camera and allows the user to take images of selected regions without running any of the two modes of operation. Before you can adjust any camera parameters, please press the “Refresh Data” to extract the most recent values for each of the parameters.

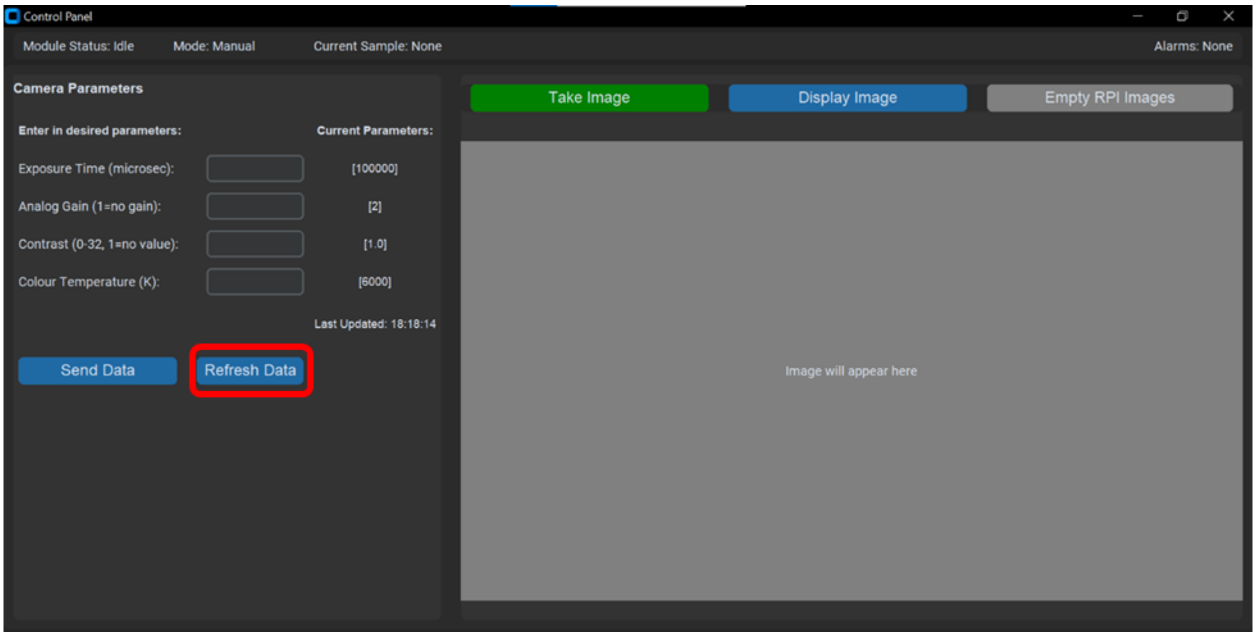


Figure 18 - “Refresh Data” button on the Image Tab without extracted image.

- You can adjust Raspberry Pi camera parameters by entering desired parameters in the associated text box and clicking “Send Data” button. Click “Refresh Data” after 5 seconds to ensure that the updated data has been received by the camera.
  - *WARNING* At the time of completion of this project, the Raspberry Pi receives the new parameters successfully but does not update the settings correctly.

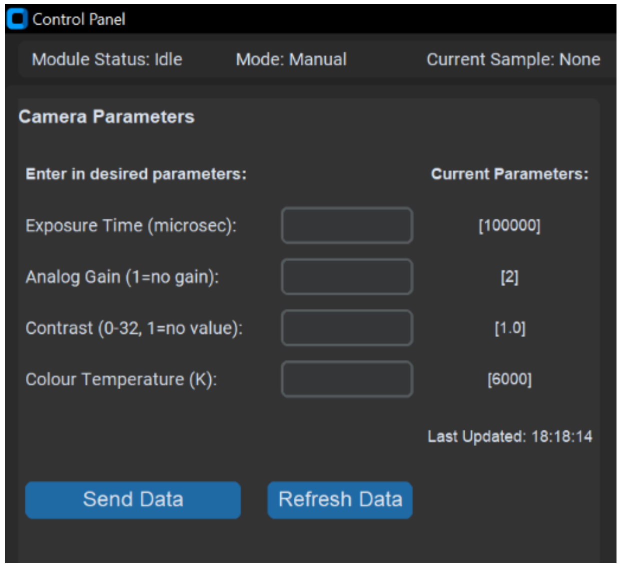


Figure 19 - Camera parameters update section on the Image Tab.

- After you position camera in the desired location, click "Take Image" button to capture the image. Then, click "Display Image" to display the image in the image window on the right side of the image tab.

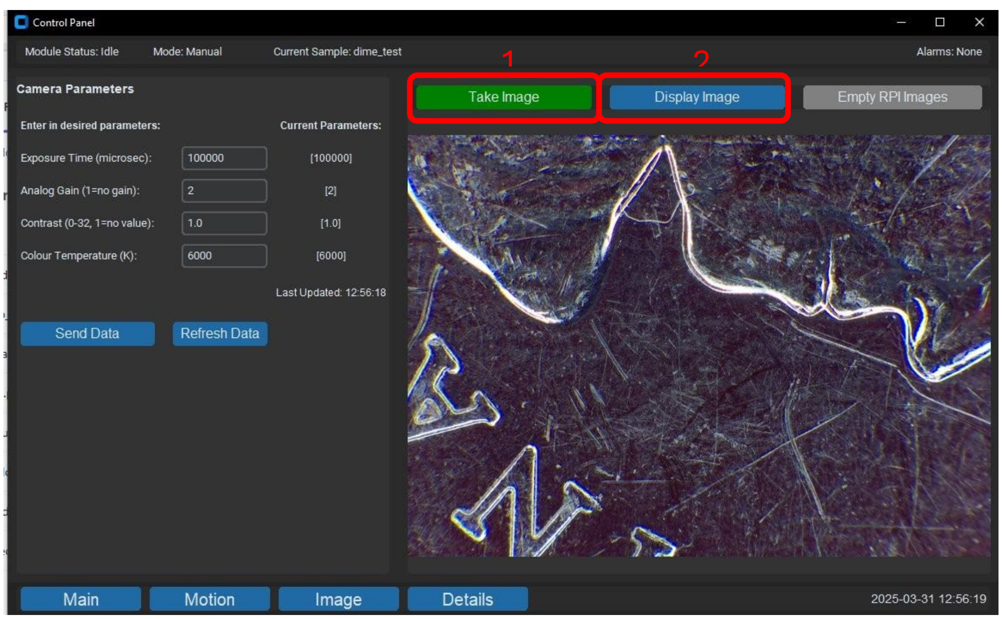


Figure 20 - Take Image" and "Display Image" buttons on the Image Tab with displayed image.

- To empty the folder that stores images on the Raspberry Pi, click "Empty RPI Images" button. This should be done automatically in all current processes; however, it can be used for troubleshooting.

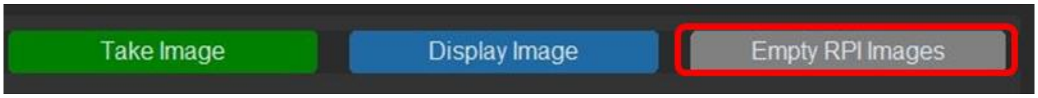


Figure 21 - "Empty RPI Images" button on the Image Tab.

2.5.3 Details Tab

Details Tab (Figure 22) includes some short information on how to use the program as well as the information on the directories used to store files and images. To change any folder paths, you must change them manually in the python file gui.py (link).

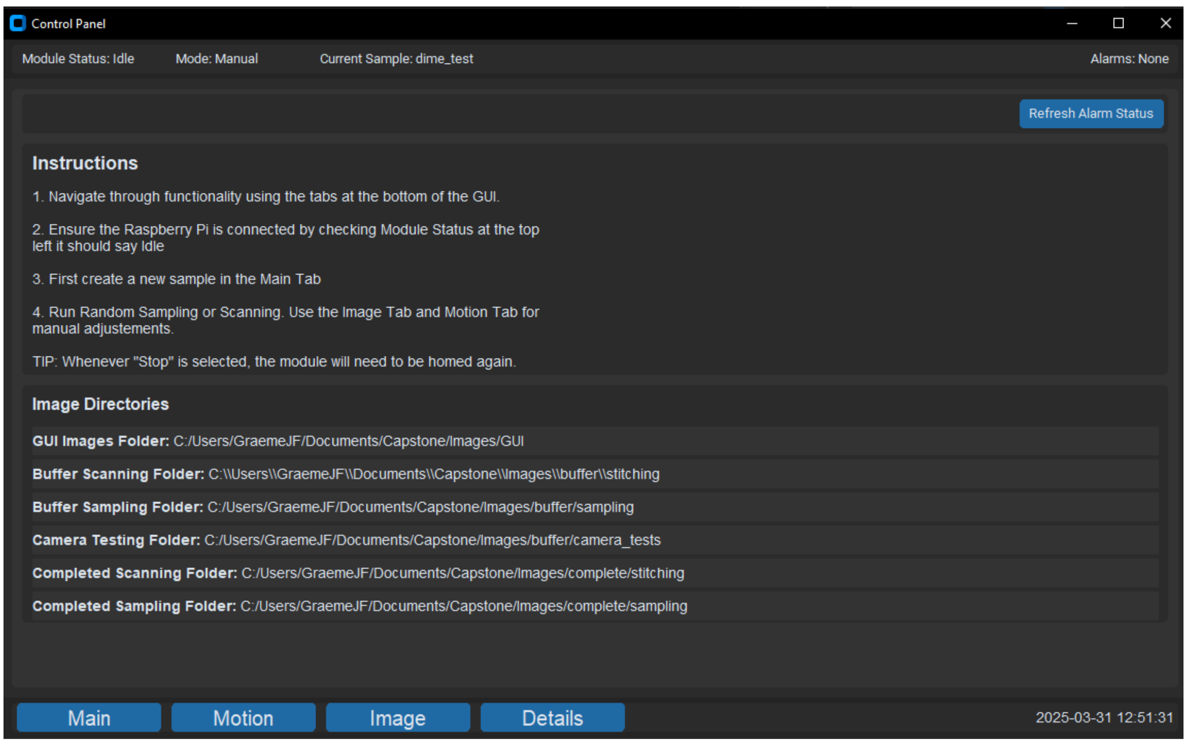


Figure 22 - "Details" tab on GUI.

2.6 Stage to Platform Matrix Transformation

This is an automatic method of establishing scaling and rotation aspects of affine transformation, the method returns the transformation matrix. It is initially intended for Click and Move feature for the microscope, but it is not implemented yet.

The method computes the affine transformation matrix by capturing images at stage coordinates **(49,28)**, **(49,27)**, and **(50,27)**. ORB features are extracted and matched across images to identify a common point, which is used to estimate scaling and rotation. Marked images are saved to the working directory as:

image1_x=49_y=28.jpg

image2_x=49_y=27.jpg

image3_x=50_y=27.jpg

This transformation maps stage movement to image coordinates, allowing spatial calibration of the camera relative to the stage. The method can be found in *opticalmodule.py*, under *matrix_transform(self)*.

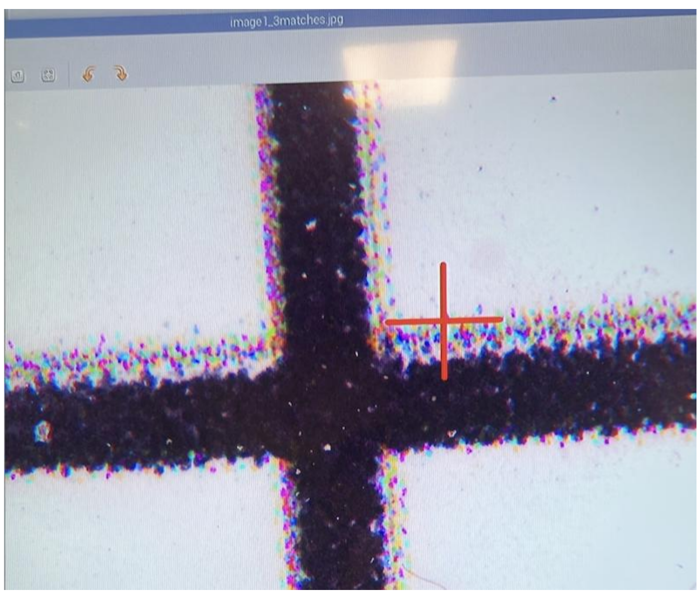


Figure 23 - The crosshair marker indicates the common feature amongst three images (locations).

3. System Overview

3.1 Structural Assembly

Figure 24 and Figure 25 and show the CAD renderings of the system's structural assembly, with key components clearly labeled for user reference. The frame consists of aluminum extrusions that support all functional modules of the system, ensuring rigidity and alignment during operation.

Critical components are highlighted to assist users with system calibration, routine maintenance, and safety awareness:

Users should familiarize themselves with these components prior to operating the system to ensure safe and accurate use.

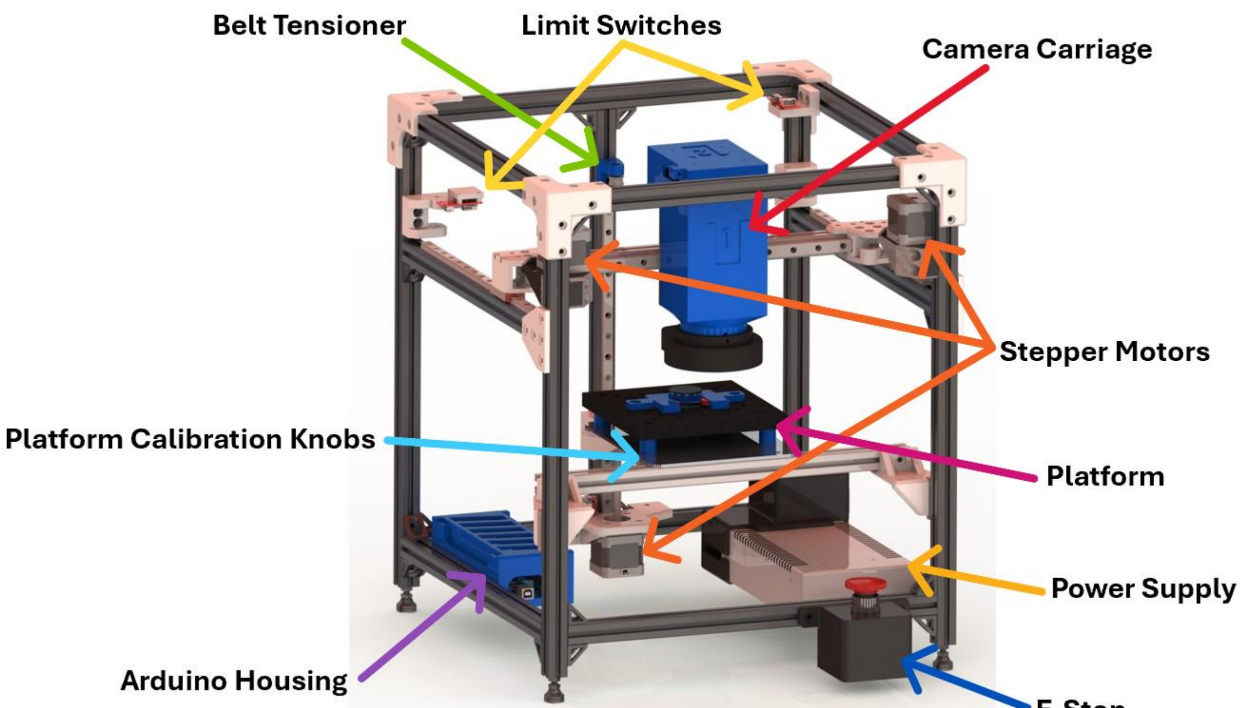


Figure 24 - Front View of the Structural Assembly.

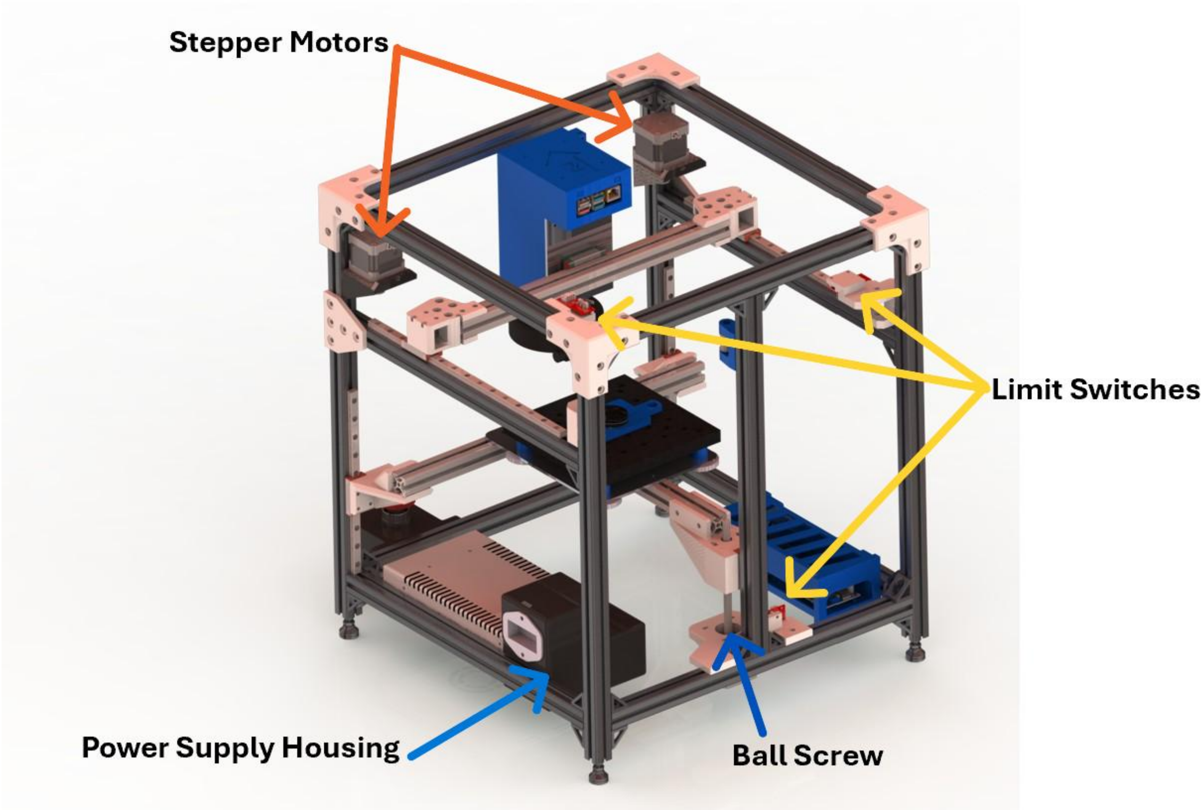


Figure 25 - Back View of the Structural Assembly

3.2 Electrical Assembly

Figure 26 below shows a representation of the electrical components involved in this project. Since the power supply is rated at 24V and 14.6 A (Max), the voltage needs to be converted to 3 levels to accommodate the CNC shield (24V), ring light (12V) and Raspberry Pi (5V), respectively. Furthermore, stepper motors and limit switches (NO) are connected to the CNC shield via their designated pins. An emergency stop is designed to cut off all power of the electrical system at the Live end of the AC power input. Arduino Uno is powered by the Raspberry Pi via a serial connection (USB-A to USB-B). In addition, there is a STOP button on the PC GUI to stop all actions without turning the power off. It is used for quick testing while saving coordinates without the need for homing. Note that the AC generator represents the 110V wall outlet and the power supply is not the exact model, but the same connection regardless.

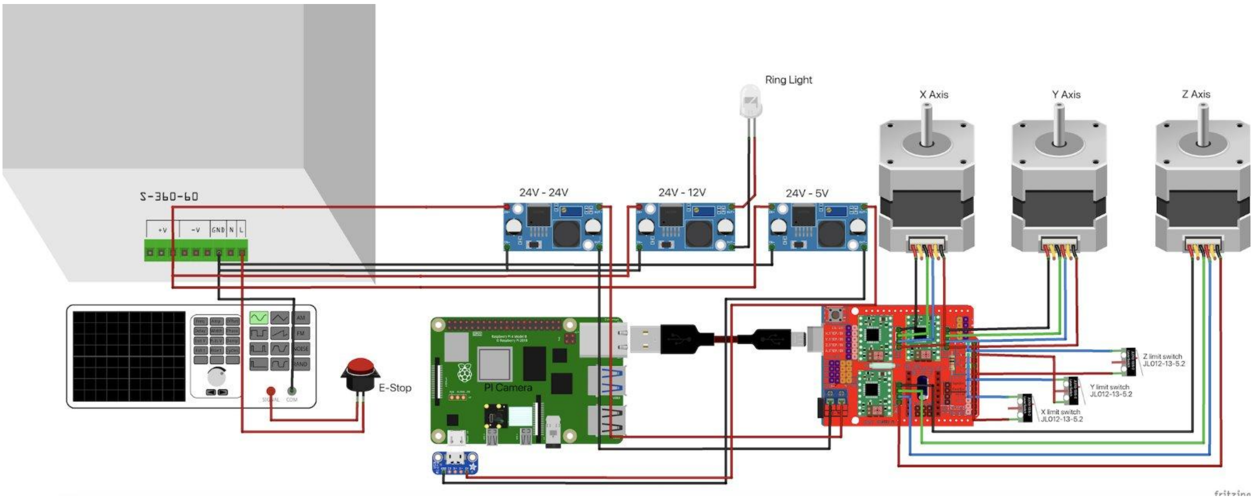


Figure 26 - Electrical component connections.

3.3 Software Startup

All programming files, and explanations can be found in the GitHub here. Before starting up the software, ensure that the ethernet cable is plugged into both the Raspberry Pi and the PC. To get into the Raspberry Pi's interface, either connect the Raspberry Pi to a monitor of it's own or use RealVNC to remote into the Raspberry Pi.

To start the code of the Raspberry Pi, first setup the virtual environment by typing in the following into the terminal:

```
source DIY_Eng_CV/bin/activate
```

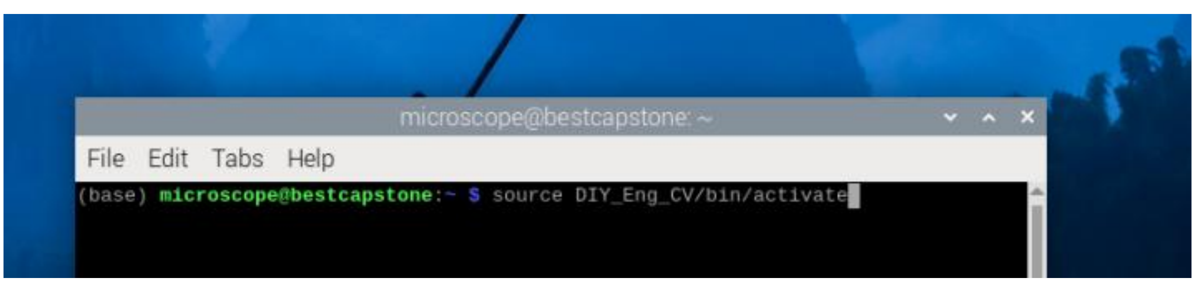


Figure 27 - Screenshot of setting up the virtual environment on the Raspberry Pi.

Then to start the code, start rpmain.py like so:

```
python rpmain.py
```

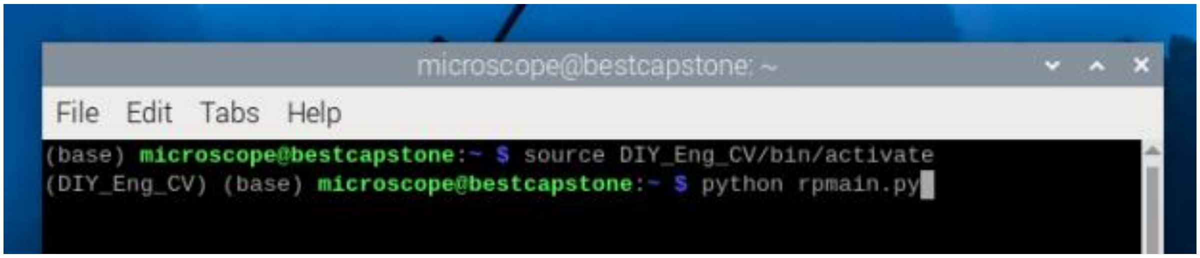


Figure 28 - Running rpmain.py on the Raspberry Pi.

To start the GUI, navigate to where the PC Files are in the PC with cd on the terminal. Then run main.py.

```
python main.py
```

4. Maintenance

4.1 Routine Maintenance

To ensure optimal performance and longevity of the system, users should follow the routine maintenance schedule outlined in Table 1. This table provides a clear breakdown of each component, the maintenance task required, the recommended frequency, and the procedure for completing the task. Regular inspection and servicing of components such as the belt system, linear rails, bearings, fasteners, and electrical connections help prevent unexpected failures and maintain calibration accuracy. Users should refer to this table periodically and document completed tasks as part of their maintenance log.

Table 1 - Routine maintenance schedule.

Component	Task	Frequency	Procedure
Belt System	Check belt tension	Every **30 days**	Inspect for proper tension and adjust if loose.
	Replace belt if damaged	Every **90 days** (or as needed)	Replace if frayed, cracked, or excessively loose.
Linear Rails & Bearings	Clean and lubricate linear rails	Every **60 days**	Wipe debris, clean with isopropyl alcohol, apply PTFE dry lubricant.
	Clean and lubricate ball screw assembly	Every **90 days**	Remove old lubricant, apply lithium/PTFE grease.
	Inspect bearings for wear	Every **180 days**	Check for excessive play or grinding noise.
Fasteners & Structural Integrity	Check witness marks and re-tighten bolts	Every **60 days**	Verify and re-tighten bolts in frame, optical module, platform, and rail assemblies.
Electrical Components	Inspect wiring connections	Every **90 days**	Ensure stepper motor, limit switches, and power cables are securely connected.
	Check power supply & voltage regulators	Every **180 days**	Confirm stable voltage output from the DC-DC converters.
	Test emergency stop functionality	Every **30 days**	Press and release the E-stop switch to verify response.

4.2 Calibration Routines

To maintain system accuracy and imaging quality, Table 2 outlines the recommended calibration routines for critical components. Each row details the specific calibration task, how often it should be performed, and a general procedure. Regular calibration ensures that the platform remains level, the optical system produces sharp and properly aligned images, and motion components such as the CoreXY system and Z-axis motor maintain precise positioning. Users should refer to this table before critical operations or if performance issues arise and follow each procedure carefully to ensure reliable system function.

Table 2 - Calibration routines schedule.

Component	Task	Frequency	Procedure
Platform Leveling	Re-level the platform	Every **60 days** (or before critical use)	1. Disable the motors, then manually move the camera carriage so it is positioned over the crosshair sticker in the bottom-left corner of the breadboard stage. 2. Capture an image using the Image tab. -If the image appears out of focus, manually adjust the height of the platform by rotating the lead screw. -Repeat the image capture and lead screw adjustment until the image is in clear focus. 3. Move on to the next corner in the counterclockwise direction and capture an image using the Image tab. -If the image appears out of focus, manually adjust the knob for that corner. -Repeat the image capture and knob adjustment until the image is in clear focus. 4. Repeat step 3 for two remaining corners, moving in a counterclockwise direction. 5. Ensure that all four corners are in focus by manually moving the camera over the cross hair and

			capturing the image. If at least one of the corners appears to be out of focus – repeat steps 3 and 4.
Optical Calibration	Check microscope focus & alignment	Every **30 days**	Capture an image of the sample or breadboard using the Image tab. -If one side of the image appears sharp while the opposite side is blurry, this may indicate a perpendicularity issue between the lens and the stage. -To correct this, adjust the knob screws supporting the bottom of the lens until the entire image is uniformly in focus. Note: Slight blurriness in the image corners is normal (spherical aberration) and does not necessarily indicate a perpendicularity issue.
	Clean the lens	Every **30 days**	Use compressed air, microfiber cloth, and lens cleaning solution if necessary.
Motion System Calibration	Check CoreXY motion drift	Every **90 days**	Move the camera carriage to a predefined position on the breadboard stage and verify the movement accuracy using a caliper: -For example, move the carriage 50 mm in the X-direction. -Measure the initial distance between the side of the carriage and the aluminum extrusion. -After the move, measure the final distance—it should reflect the expected 50 mm shift. If there is a discrepancy between the expected and actual movement, perform maintenance on the Belt System to correct the issue.

	Z-axis DC Motor drift check	Every **90 days** (or if accuracy issues arise)	Move the platform to a predefined position along the Z-axis and verify the movement accuracy using a caliper: -For example, move the platform 20 mm upward. -Measure the initial distance between the platform and the lead screw bracket. - After the move, measure the final distance, it should reflect the expected 20 mm shift. If there is a discrepancy between the expected and actual movement, perform maintenance on the Linear Rails and Bearing system to correct the issue.

4.3 Equipment

Table 3 lists parts and materials that will need to be purchased somewhat regularly to upkeep and maintain the optical module.

Table 3 - List of parts that will be regularly purchased.

Part	Specifications/Notes	Approximate Cost (CAD)
Linear Rail Lubricant	PTFE Dry lubricant or Superlube	$15 - $25
Ball Screw Lubricant	Super Lube 92003 Silicone Lubricating Grease with PTFE or Lucas Oil 10533 White Lithium Grease	$15 - $35
Belt	GT2, 6mm wide timing belt. ~1m long.	$15 - $35
Isopropyl Alcohol		$5 - $20

5. Safety

5.1 Motors

There are three stepper motors in use for the optical module. The motors get hot after some time. Avoid touching the motors when enabled, or when the device has been on for more than 10 minutes.

Furthermore, avoid entering within the confines of the frame when the optical module is in use to avoid any pinch points. Use the E-Stop immediately if any complications occur.

Lastly, don’t have any loose hair, clothing, or jewelry near stepper motors and belts when the device is enables.

5.2 Electrical Wiring

5.2.1 Power Supply

If making any modifications to the power supply housing, ensure that the optical module is unplugged from the outlet, and that the power supply has been off for about five seconds at least. The power supply stores energy in very large capacitors, so the energy does *not* dissipate immediately.

5.2.2 Arduino and Converter Housing

Do not make any modifications to the housing for the converters or Arduino when the system is running or has power. This could cause adverse affects to the motors, Raspberry Pi, and overall software. Furthermore, although the electric current and voltage is low, it is inadvisable to make any modifications when the system is running due to electrical safety concerns.

5.2.3 E-STOP

The optical module is equipped with an E-STOP with is wired between the AC outlet and DC power supply. When the E-STOP is pressed, any electrical current is mechanically stopped from flowing to the power supply. **Warning**: the DC power supply is equipped with large capacitors, meaning that even after the E-STOP is pressed, the energy won't fully dissipate for a few seconds.

6. FAQ

Why won't an old alarm go away?

Alarms are not automatically refreshed. To erase an old alarm, navigate to the details tab and select "Refresh Alarm".

Where do the images I take go?

During the process for both sampling and scanning, images are saved to a "Buffer" folder. Then when the "Finish" button is selected, all the files are placed in a new folder under the "complete" folder, with the folder labelled with the sample id and time stamp. For specific folder paths, reference the "Details" tab on the GUI.

To change the folder path directories, they most be modified in the gui.py python file (link).

Why is the Z-axis leadscrew making inconsistent noise?

Z-axis leadscrew may be misaligned. By visual inspection, one can determine the deviation when turning the leadscrew (when power is off). Alternatively, the leadscrew may be more difficult to turn at the very top. The solution would be to adjust the stepper motor screws within the sliding screw holes on the motor bracket and continue the test until the leadscrew is aligned with consistent torque throughout its length.

7. Troubleshooting

Why is the program crashing as soon as I send data?

If the program is crashing when trying to use GoTo, Create a New Sample, or Take Image right after opening the GUI, this is an indication that the Raspberry Pi is not receiving information.

First check the Module State in the top left corner. If the status is “Raspberry Pi is Not Connected” this confirms the Raspberry Pi is not communicating with the Gui.

Second, ensure that the Raspberry Pi has power. Third, close the GUI, and then try opening it again. Check to see if the Module State is still “Raspberry Pi is Not Connected” or updated to “Idle”. Lastly, cut power to the optical module, and then power it on again.

Why is my image blurry?

For the scanning function, the entered in “Layer Height” must be set to 0.

For the sampling function, ensure that the initial layer height is correct.

If neither of these solutions work, consider re-leveling the platform using the knobs underneath and the three screws aligning the lens within the camera carriage.

Why is the GUI crashing when I’m stitching an image?

The program that runs image stitching, Fiji, requires a lot of RAM. If there are too many images to stitch, this could cause the PC to max out it’s RAM and crash the system (e.g. 50+ images). Please enter a smaller width and height in the “Create New Sample” dialog if this occurs.

Raspberry Pi is not starting up.

It could be the voltage supply of the buck converters. While the rated voltage for raspberry pi is 5.1 V, sometimes it spikes to above 5V triggering built in protection. It is advisable to turn the converter to less than 5V. If adjusting the converter fails, try powering the Raspberry Pi with a separate USB-C charger.

Why is the Raspberry Pi not connecting to Wifi?

The Raspberry Pi will not be able to connect to a wifi network if it is currently connected via ethernet to the PC. In order to access a wifi network, you must unplug the Ethernet cable.