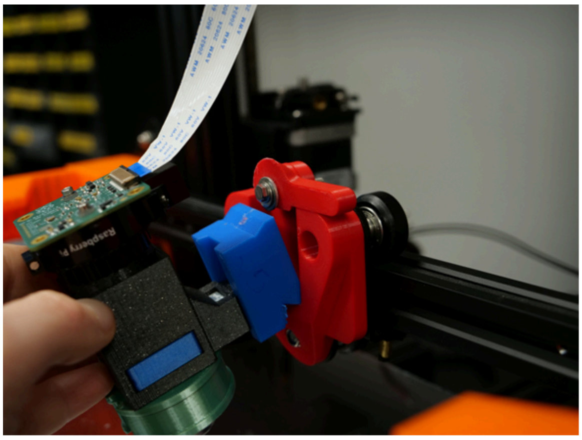
Assembly Instructions

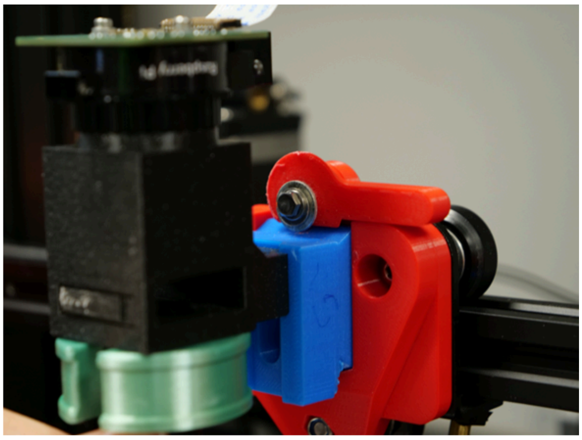
These instructions assume you have already assembled your Ender 3 Pro 3D printer.

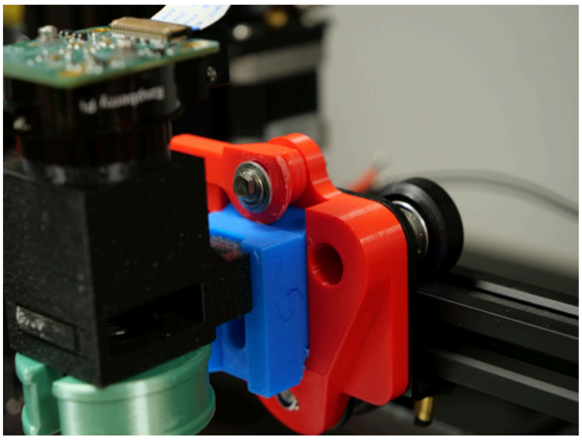
Step 1: Quick Change Mechanism

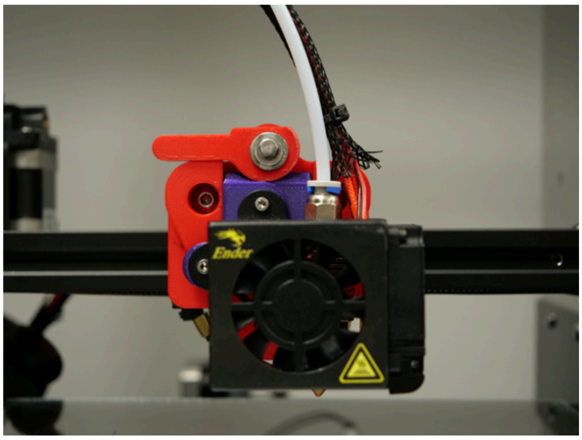
The quick change mechanism is designed to allow you to easily swap back and forth between the original hotend of the 3D printer and the microscope. This means you don't have to sacrifice your 3D printer to have a microscope.

The quick change tool mechanism was adapted from the following design by Jón Schone (ProperPrinting) on Thingiverse “Quick tool change for Creality CR-10 / Ender series” (CC-BY-SA 3.0, https://www.thingiverse.com/thing:3369444/files). There are more details including a youtube video showing how to mount the base onto the X-axis of the printer at the above link.



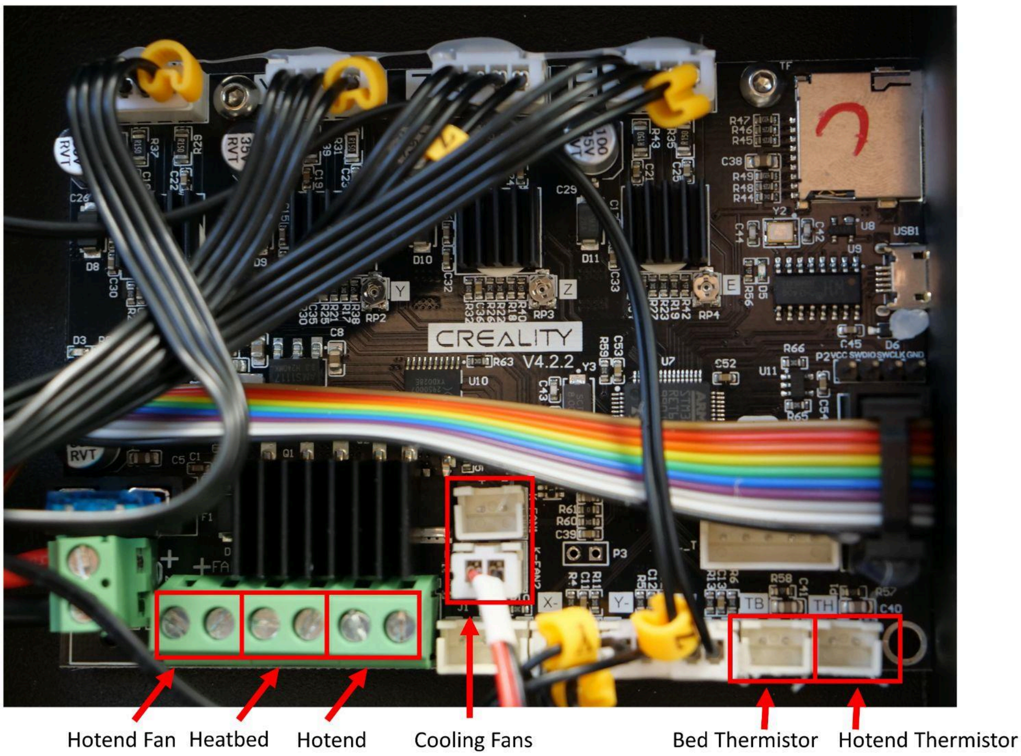




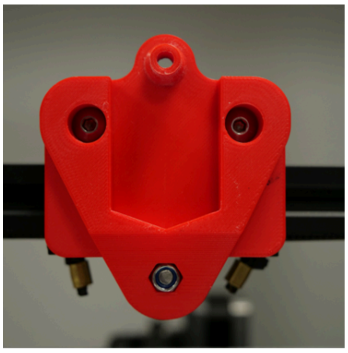


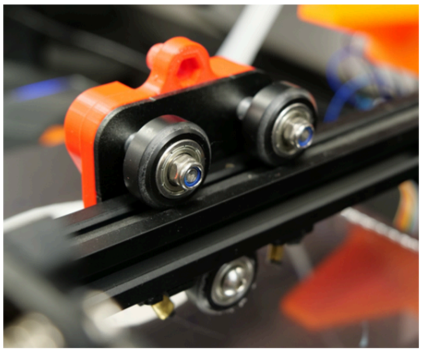
- Print all printed parts (PLA, 0.2 mm layer height, 0.4 mm brass nozzle that comes with printer, no supports (except for optional filter changer)). Best to do this before dismantling the hotend. In particular, make sure 'ToolHolderBase', ‘Lever’ and ‘EnderHotendToolHolder’ are printed before dismantling the hotend of the printer as these are essential to allow the hotend to be reattached.

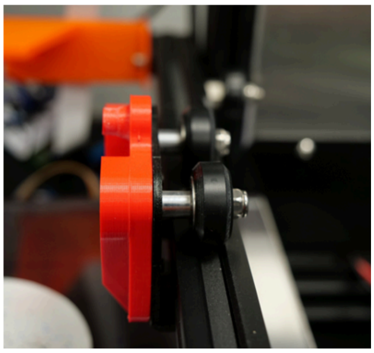
- Remove ender hotend. To do this first unload the filament (***NB*** wait for Ender to fully cool down after this). Next unscrew the two M4 screws mounting the casing covering the hotend. Then unscrew the two M4 screws mounting the hotend on the X-axis carriage.
- Next open up the black casing covering the motherboard. Unplug the nozzle thermistor, bed thermistor, two part cooling fans, hot end fan, heat bed and nozzle.



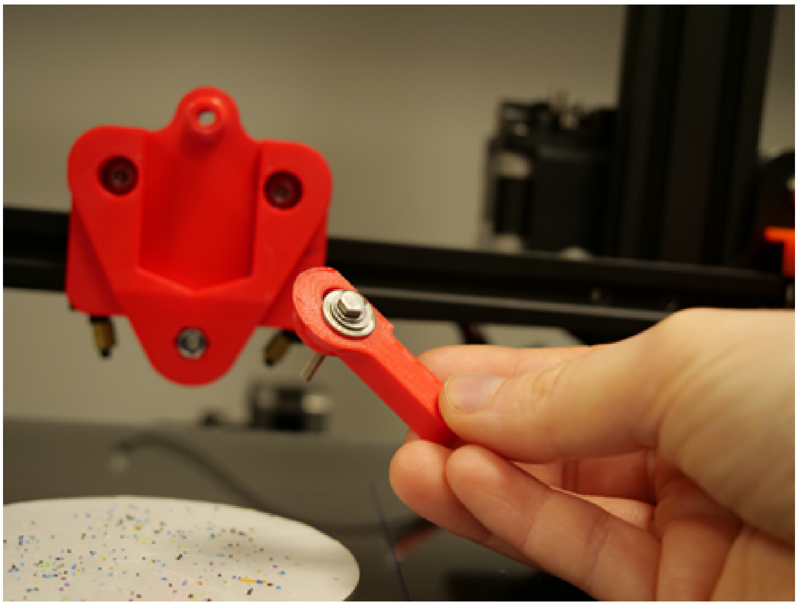
- Attach the printed **'ToolHolderBase'** to the x-axis rollers with the screws and nuts supplied with the Ender. Reattach the wheels exactly as they were originally (i.e. remember that the bolt on the bottom wheel is facing the opposite direction to the top wheels). Use a wrench to hold the nuts in place when tightening.



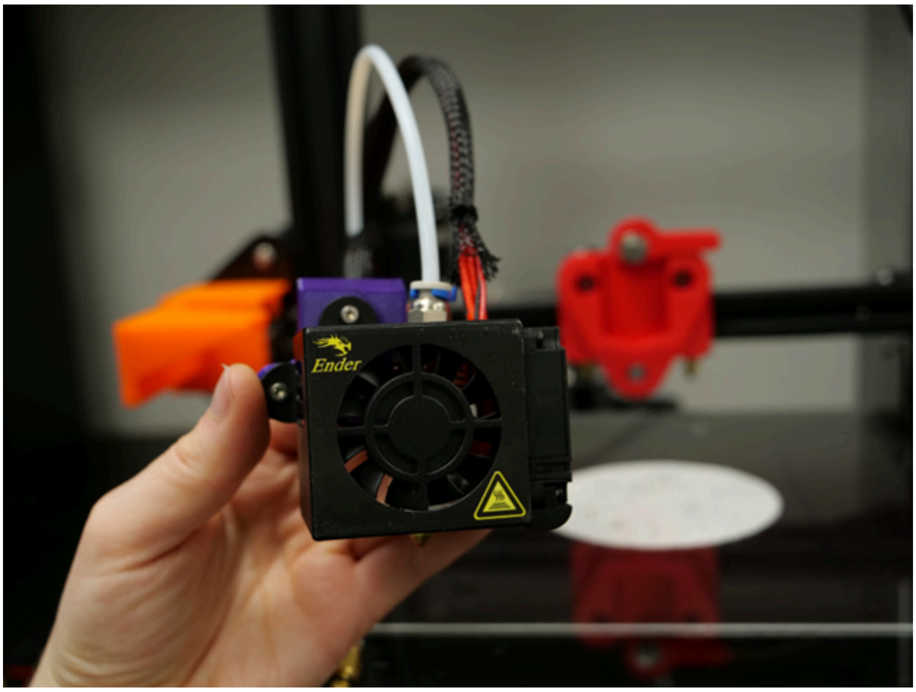




- Attach the printed **'Lever'** with an M4 screw (16 or 20mm) and M4 washer (you can also use an M3 screw with two different sized M3 washers, one thicker (larger outer diameter) than the other. This prevents you from having to buy another type of screw as you may have already purchased M3 screws as they are needed for the illumination mount). This is intended to be a tight fit and you may need to widen the hole with a drill. Ensure the **'Lever'** is tightly secured. When attaching the optics module later, if the **'Lever'** is not securely attached the optics module may wobble during imaging (this is particularly important if you want to do stitching).

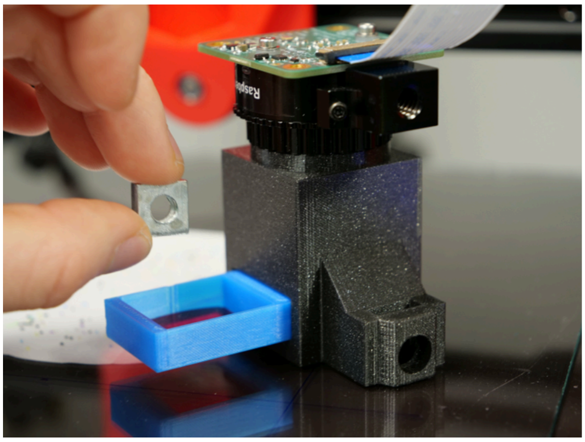
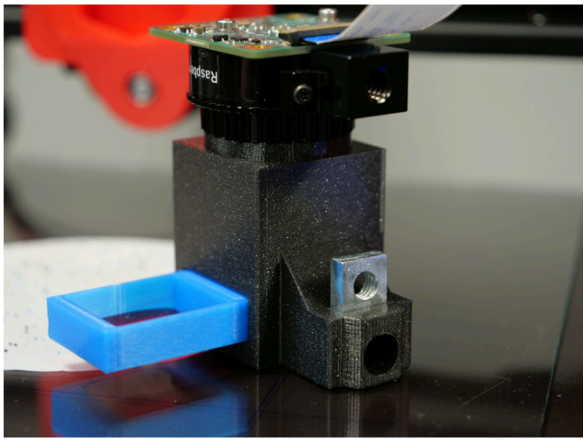


- Attach the Ender hotend to the printed **'ToolHolder'** using two M2.5 screws.



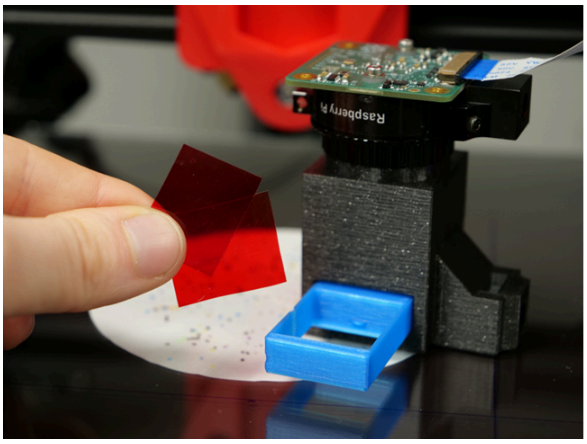
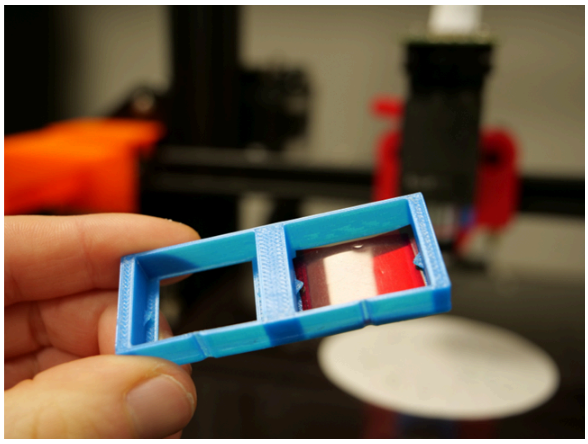
Step 2: Lens Tube

- Print **‘LensTube’** if you haven’t already. There may be some stringing on the inside of the print. You can remove this by hand or with a wire cutter. Insert square M6 nut into the print.

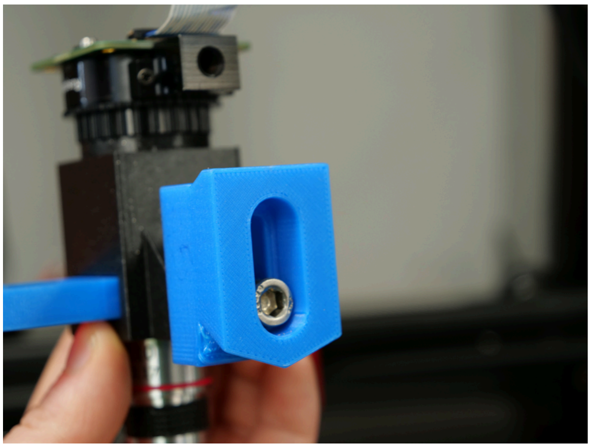
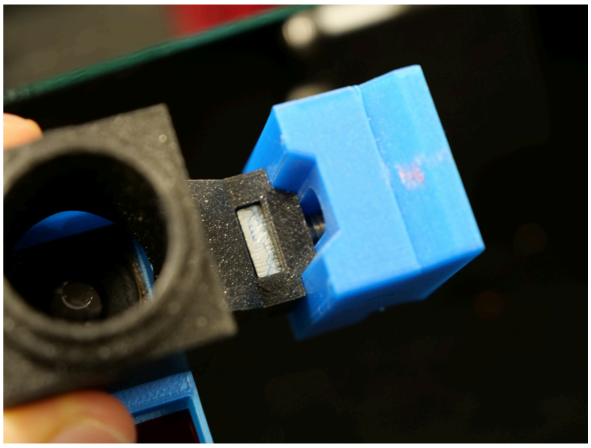
 

Step 3: Filter Slider

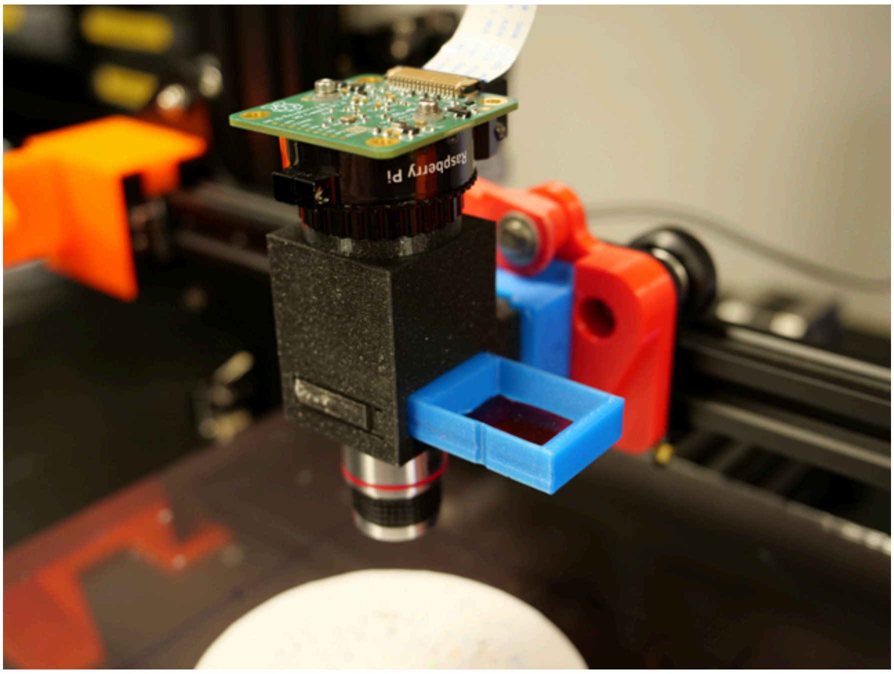
- Print **‘FilterSlider’** if you haven’t already. Cut out a small square from the lighting gel that is big enough to cover one of the holes in the **‘FilterSlider’** print and clip it in under the triangle notches in the print.

- Attach Raspberry Pi HQ Camera to C-mount threads on the **‘Lens Tube’.**
- Attach **‘EnderScopeToolChanger’** to the **‘Lens Tube’** by placing an M6 (12mm, cap head screw will work nicely) screw through the back of the tool changer (with head of screw sitting into hole) and screwing this through to the square nut embedded in the ‘**Lens Tube’** print. The ridge of the ‘**EnderScopeToolChanger’** should line up with the ridge on the **‘Lens tube’**.

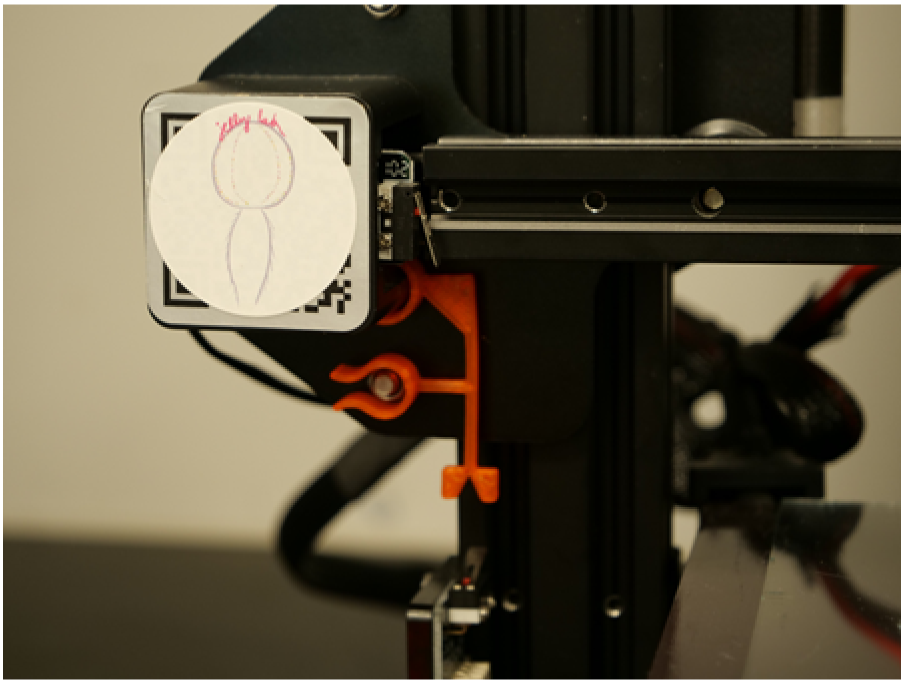



- The entire **‘EnderScopeToolChanger’** and ‘**Lens Tube’** assembly can now be mounted on the **‘ToolHolderBase’** on the x-axis of the printer.
- Screw a 4x finite conjugate RMS threaded objective lens into bottom of the **‘Lens Tube’**. We use a low-cost 4x finite conjugate objective lens (4x 185 Biological Microscope, Standard Plan Achromatic Objective Lens). Available for ~ €8 at the time of writing.



Step 3: Limit Switch Extender:

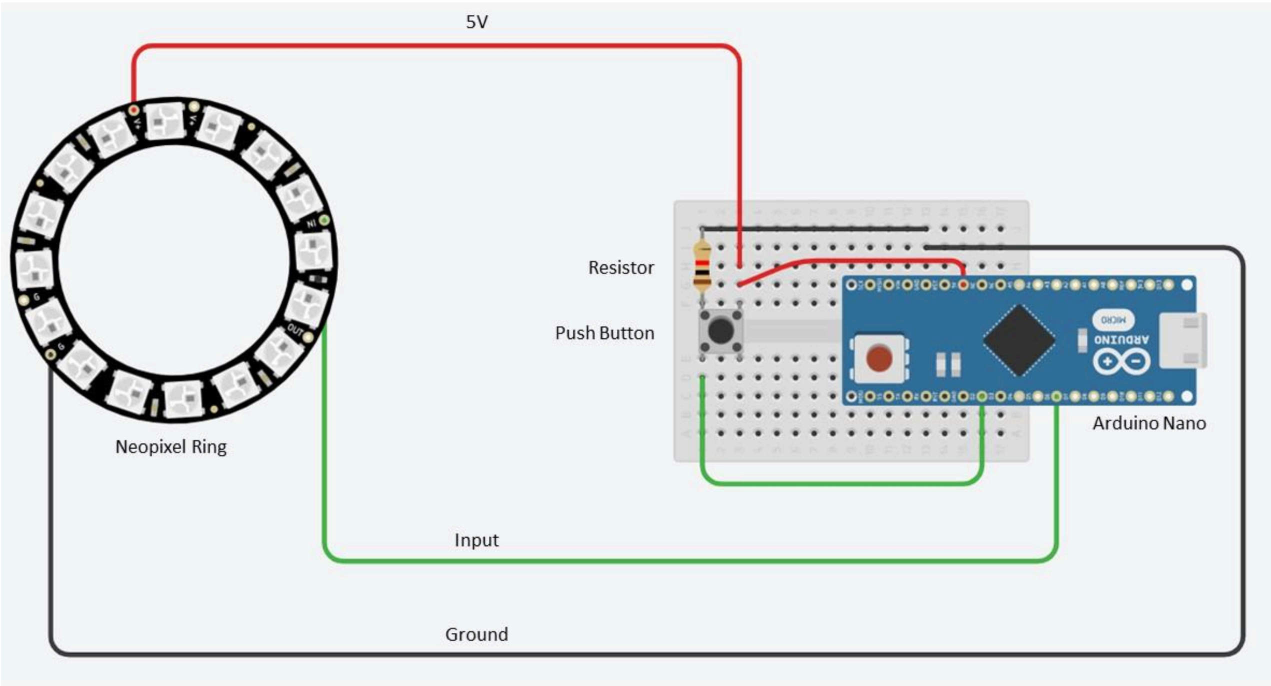
- If using the microscope module, first attach the **'LimitSwitchExtender'** onto the plastic covering of the x-axis limit switch. This will trigger the z-axis limit switch from a higher position so the objective lens will not crash into the bed during homing.



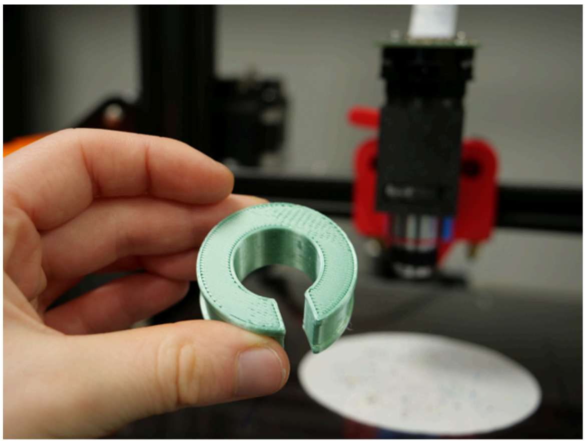
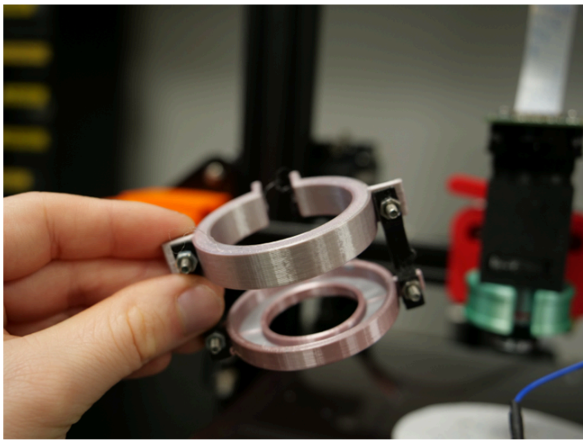
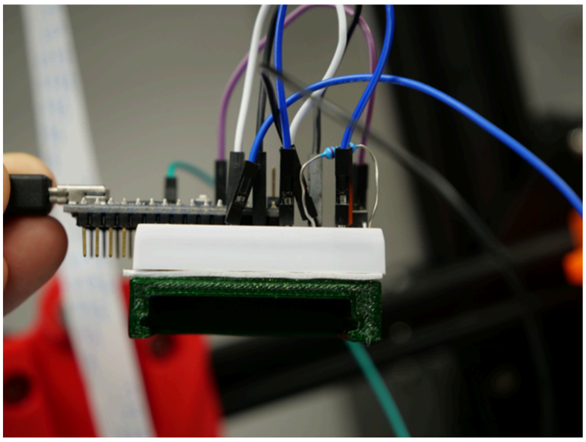
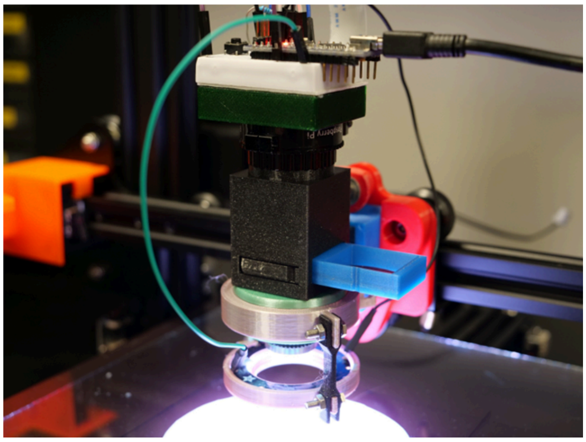
Step 4: Illumination

- Print **‘NeopixelClamp’**, **‘NeopixelMount’** and **‘NeopixelClamp_GenericInsert’**.

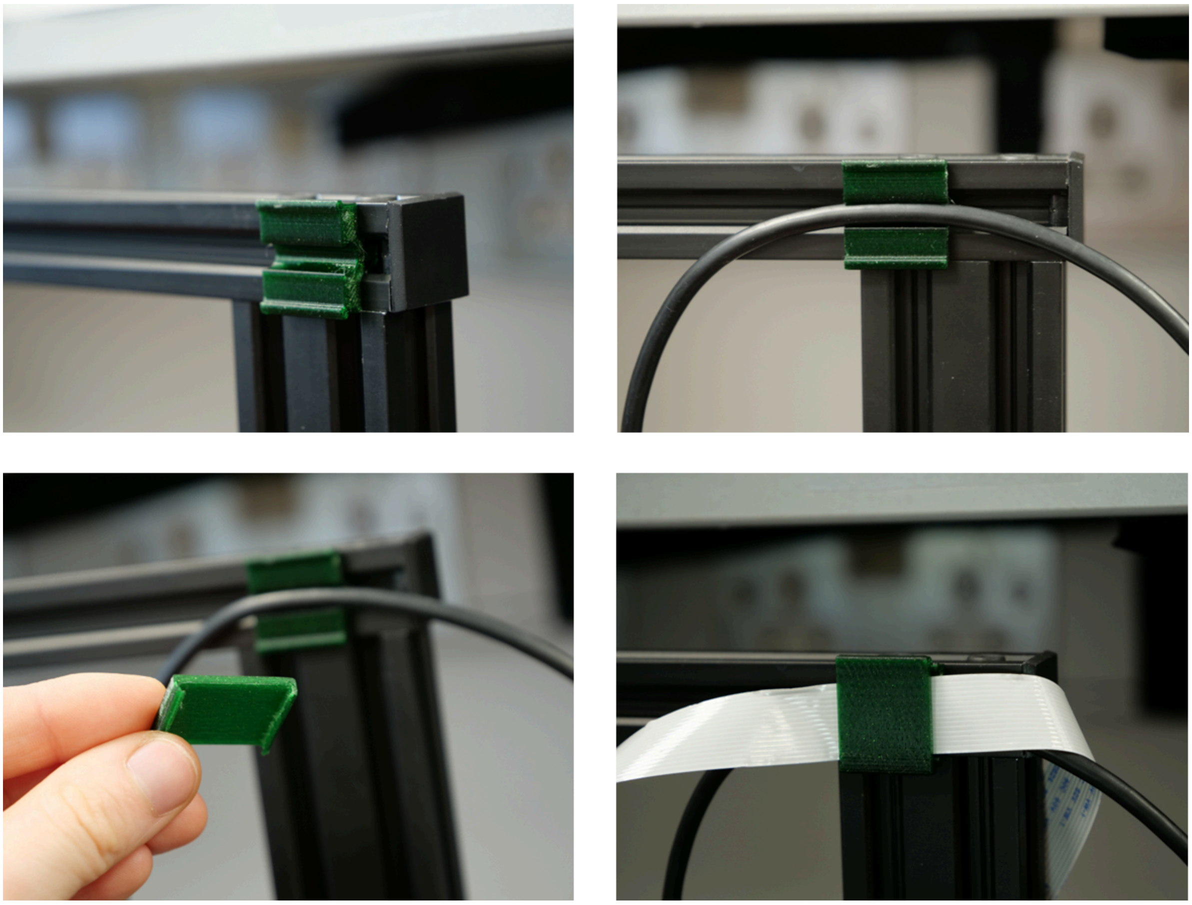
- Place the **‘NeopixelClamp_GenericInsert’** around the objective lens, then clamp the **‘NeopixelClamp’** around it.
- Attach two **‘NeopixelConnecter’** to the **‘NeopixelClamp’** with M3 screws and nuts.
- Attach the **‘NeopixelMount’** to the **‘NeopixelConnecter’** with M3 screws and nuts.
- Solder dupont cables to 5v, ground and data input solder pads of the Neopixel 16 LED ring. To reduce the need for soldering you can also purchase a wired version of this Neopixel ring.
- Wire up the neopixel as shown below:



- Remove the adhesive backing on the 170 pin breadboard and place this on the **'ArduinoNanoHolder'**. Slide the **'ArduinoNanoHolder'**, with the breadboard, push button and Arduino Nano attached, onto the Raspberry Pi Camera.

- Attach the **'CableTidy'** to the horizontal cross piece on the Ender frame. Clip the USB cable and Raspberry Pi CSI cable into the **'CableTidy'**. The T slot part of the **‘Cable Tidy’** design was adapted from the following design by Jarda Linek on Printables “Cable Holder (V) Slot” (CC-BY-SA 4.0, https://www.printables.com/model/339499-cable-holder-v-slot).

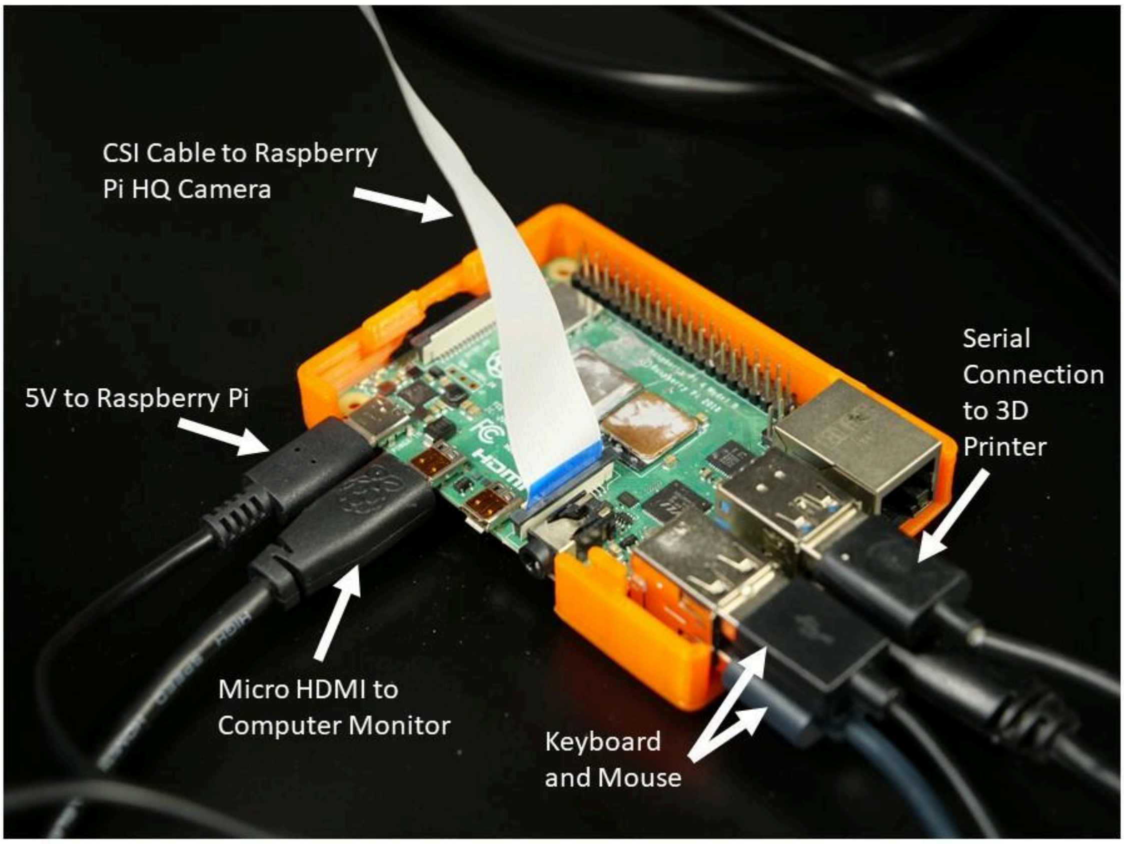


- Upload the relevant Arduino sketch (**‘EnderScopeNeopixelLED.ino’**) to the Arduino nano.

Step 5: EnderScope Setup

- Connect the Raspberry Pi HQ Camera to the ‘camera’ port on the Raspberry Pi using the long CSI ribbon cable.

*** If you have received the Raspberry Pi touchscreen with your EnderScope, connect the touchscreen monitor to the ‘display’ port on the Raspberry Pi using the short ribbon cable. If you would like to view your images on a larger computer screen display, connect the Raspberry Pi to monitor with HDMI-micro HDMI cable.**



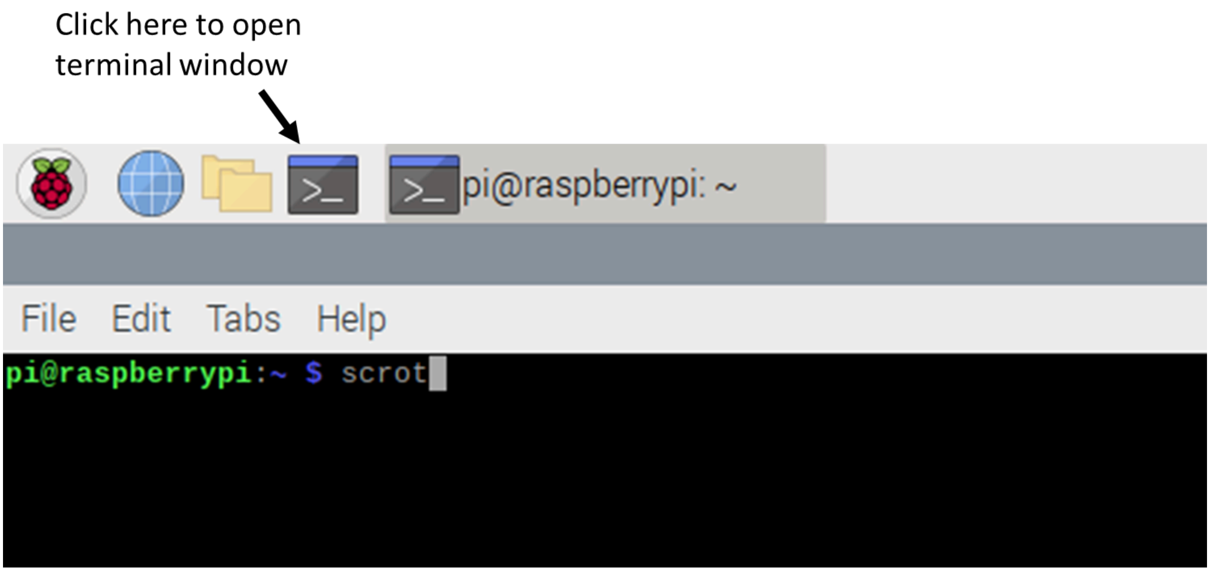
- Turn Ender on.
- Power up Raspberry pi and then connect Ender to pi using the USB to microUSB cable.

NB It is really important to switch the Ender printer on **FIRST** before attaching it to the Raspberry Pi. Otherwise the Pi will try to power the Ender itself.

- Remove the build surface for the Ender and place the glass frame on the print bed.
- Clip in place with binder clips (be careful where you position the binder clips. Make sure they will not crash into the aluminium frame of the printer when it homes. **Otherwise the glass frame may break!!!!**).

Software Installation Instructions

- Open terminal window on the Raspberry Pi.



- Check if python is already installed on your Raspberry Pi by running the following command in the terminal window:

```
Unset
python --version
```

- If python is not already installed, python using Raspberry Pi terminal window:

```
Unset
    sudo apt-get update
    sudo apt-get install python3
```

- To check if pip is installed run:

```
Unset
pip3 --version
```

- If pip is not installed, install pip by running:

```
Unset
sudo apt-get install python3-pip
```

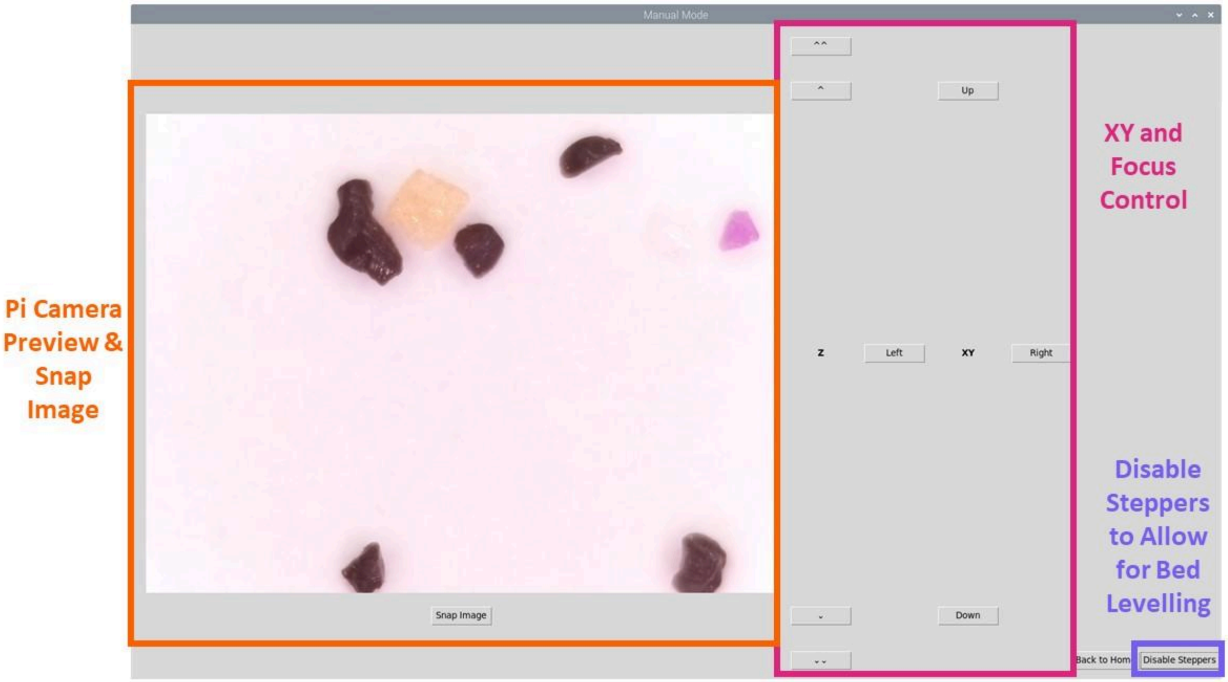
- Use pip to install the following python packages:

```
Unset
pip install matplotlib
pip install numpy
pip install pyserial
pip install picamera
pip install picamera[array]
pip install pillow
```

Bed Levelling Protocol

- Print out the A4 paper **'BedLevelSheet'**, cut to size to fit the Ender print bed and tape it to the Ender print bed. Make sure it is entirely flat. Place glass frame on top of the paper and clip into place on the Ender print bed.
- Open ManualMode.py. You can open this script using a python IDE such as Thonny or Geany (should come preinstalled with Raspberry Pi) or you can run this script straight from the terminal window as shown below:

```
Unset
python /directory/ManualMode.py
```



- Click disable steppers button in the bottom left corner of the Graphical User Interface (GUI).
- Open the GUI and move to the centre cross. Focus the EnderScope on the centre cross using the lead screw. Then move the Z axis approximately one quarter turn of the lead screw below the plane of focus.
- Next move the EnderScope optics module to each corner of the bed and adjust the bed level screws until the crosses in all four corners of the 'BedLevelSheet' are in focus **without** changing the Z position of the microscope.

Run Snake Scan

- Open SnakeScan.py in IDE. The following **bolded and underlined** lines in the script will need to be adjusted:

```python
import matplotlib.pyplot as plt
import numpy as np
import serial
from picamera import PiCamera
import time


if __name__ == "__main__":
  folder = "/home/pi/" #Replace this with folder you would like your images saved in
  #Opens serial port
```

```
  ser = serial.Serial('/dev/ttyUSB0', baudrate=115200)# Enter name of serial
port
  xBase = "G1X"
  yBase = "Y"
  endTerm = "\n"
  speed = 18000  #max_speed
##     Uncoment to home printer at beginning of script - ***WARNING*** You
must make sure objective lens will not crash in to bed before homing. Attach
limit switch extender or remove optics module during homing.
#     home = "G28" + "F" + str(speed) + "\n"
#     home = home.encode('utf-8')
#     ser.write(home)
  x_mid = 0
  y_mid = 0
  startPos = xBase + str(x_mid) + yBase + str(y_mid) + "F" + str(speed) + endTerm
  startPos = startPos.encode('utf-8')
  ser.write(startPos) # go to half max
  # open camera (image acquisition)
  camera = PiCamera()

def snake(x_win, y_win, x_step, y_step):
  a = [] # List of X positions
  b = [] # List of Y positions
  x = 0
  y = 0 # Start point at the bottom left corner

  for j in range (y_win - 1):

    if j == 0 :

      for i in range(x_win): #increment x step

        if i % x_win != 0:
          dx = x_step
          dy = 0

        elif i % x_win == 0:
          dx = 0
          dy = 0

        x = x + dx
        y = y + dy

        a.append(x) #take list a and stick x into it, so 0 is first number in
list
        b.append(y) #take list b and stick y into it, so 0 is first number in
list
```

```
    if j % 2 != 0:

      for i in range(x_win):

        if i % x_win != 0:
          dx = x_step
          dy = 0

        elif i % x_win == 0:
          dx = 0
          dy = y_step

        x = x + dx
        y = y + dy

        a.append(x)
        b.append(y)

    elif j % 2 ==0:

      for i in range(x_win):

        if i % x_win != 0:
          dx = -x_step
          dy = 0

        elif i % x_win == 0:
          dx = 0
          dy = y_step

        x = x + dx
        y = y + dy

        # Round the x-value to 2 decimal places
        x = round(x, 2)

        a.append(x)
        b.append(y)

  x_vals = np.array(a)
  y_vals = np.array(b)
  print (x_vals)
  print (y_vals)
  return a, b

x_win = 3 # Number of steps in the x direction
y_win = 3 # Number of steps in the y direction
x_step = 5 # Step size in the x direction
y_step = 5 # Step size in the y direction
```

```
x_mm = x_win*x_step # to calculate area scan will cover in mm
y_mm = y_win*y_step


x, y = snake(x_win, y_win, x_step, y_step)


tune = "M300 S440 P200" + endTerm
ser.write(tune.encode('utf-8')) # play tune to signal start of scan
pause = "M18" + endTerm # disable stepper - allows user to position optics
module in bottom left corner of sample
ser.write(pause.encode('utf-8'))
camera.start_preview(fullscreen=False, window=(100, 100, 640, 480))
print('Steppers have been disabled')
print('Ensure sample is in focus. Then move optics module to bottom left
corner of sample')
input("Press the Enter key to proceed")
camera.stop_preview()
resume = "M108" + endTerm
ser.write(resume.encode('utf-8'))

print('This is the path of the snake scan. Close graph to proceed.')
plt.plot(x, y, marker = 'x') # plots path of scan
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Snake Scan')
plt.grid(True)
plt.show()


print(f"This will now start a snake scan, scanning over an area {x_mm} x {y_mm}
starting from the bottom left corner of the sample, going right and up")
input("Press the Enter key to proceed")
resume = "M108" + endTerm
ser.write(resume.encode('utf-8'))

camera.start_preview(resolution=(640, 480))
for i, (x_val, y_val) in enumerate(zip(x, y), start=1):
    # Move the printer here (adjust the timing as needed)
  xPos = x_mid + x_val
  yPos = y_mid + y_val
  Mot = xBase + str(xPos) + yBase + str(yPos) + "F" + str(speed) + endTerm
  Mot = Mot.encode('utf-8')
  ser.write(Mot)
  time.sleep(1)
  # Use string formatting to create the file name with leading zeros
  num_digits = len(str(len(x))) # Determine the number of digits needed
  file_name = f"{folder}_{i:0{num_digits}d}.bmp"
  camera.exposure_mode = 'off'
```

```
  camera.awb_mode = 'off'
  camera.resolution = (1014, 760)
  # Configure white balance and exposure settings for each image capture
  camera.awb_gains = (1, 1) # R and B gains, both set to 1.0 for neutral white
balance
  camera.shutter_speed = 8000 # exposure setting - adjust as needed
particularly for fluorescence imaging

  # Capture the image with the constructed file name
  camera.capture(file_name)
  time.sleep(1)
# Close the camera preview when the scan is finished
camera.stop_preview()
```