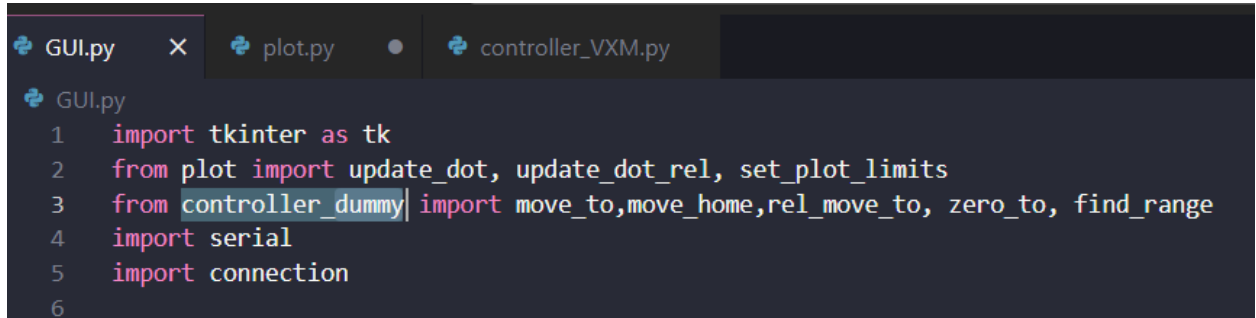


## Directions of the Hodoscope/MCP Linear Stage Program:

### 1. Start Up

Once you have downloaded the .py file and required dependencies. Within the GUI.py file, import the desired controller file (e.g. controller\_VXM”, controller\_VP9000’, or ‘controller\_dummy’).



```
GUI.py x plot.py controller_VXM.py
GUI.py
1 import tkinter as tk
2 from plot import update_dot, update_dot_rel, set_plot_limits
3 from controller_dummy import move_to, move_home, rel_move_to, zero_to, find_range
4 import serial
5 import connection
6
```

Now, unless you are running controller\_dummy, plug in the RS-232 cable into the PC that will be running the program and connect to the motor controller.

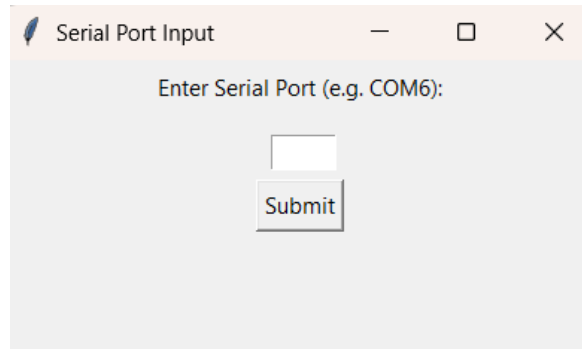
Open ‘Device Manager’ to check which COM port the controller is connected to. I also recommend double checking other connection settings like parity, baudrate, bytesize, etc to ensure connectivity. You can change these in the code block below:



```
GUI.py x plot.py controller_VXM.py
GUI.py > ...
17 def on_submit():
24     print("Using dummy controller.\n Opening serwindow.")
25     serwindow.destroy()
26     else:
27         try:
28             connection.ser = serial.Serial(
29                 port=SERIAL_PORT,
30                 baudrate=9600,
31                 bytesize=serial.SEVENBITS,
32                 parity=serial.PARITY_EVEN,
33                 stopbits=serial.STOPBITS_TWO,
34                 timeout=1
35             )
```

Now run ‘GUI.py’

## 2. Opening the GUI.py

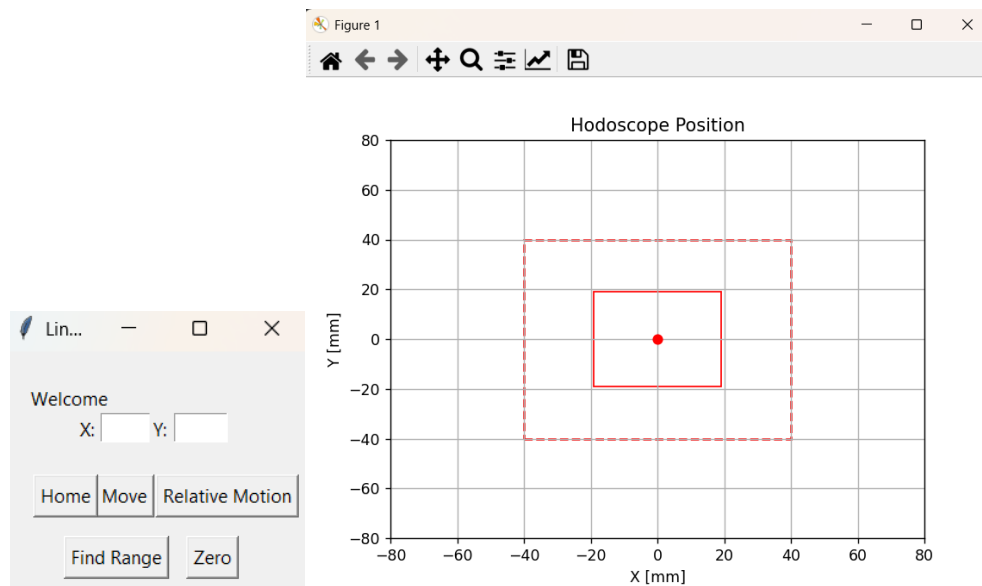


When the serial connection window opens, input the serial port name. If you are running the dummy controller, type 'dummy' and the window will close.

Once connection has been established, the motors will run the `zero_to()` and `find_range()` commands.

## 3. Navigating the GUI

Once the motors are ready to be used, two windows will open: the position graph, and the controller menu



Commands:

- *Home*: This command will move the stage and the plot back to the (0,0) position

- *Move*: This command will take the inputs provided in the text boxes to move the stage to the absolute position.
- *Relative motion*: Like *Move*, this command will take the inputs provided in the text boxes to move the stage to a relative position (e.g. input = (1,2) previous position = (1,0); Output = (2,2))
- *Find Range*: This command will move the stages to find their respective range of motion. This function also updates the position plots limit range
- *Zero*: This command functions like the *Home* command but rather than move to (0,0) it also redefines (0,0) which will fix any bugs with the motors. (e.g. if the motor gets stuck run this command to ensure correct absolute positioning)

#### Hodoscope Position Plot:

- This plot shows the position of the Hodoscope using a plot with the ranges  $X = [-80, 80]$  and  $Y = [-80, 80]$ , however this is not the actual range of the linear stages with the Hodoscope sits on. That range is shown with the dotted red box which confines the center of the Hodoscope. This confinement box can change its range when the *Find Range* command is running (This command is also running at the beginning of the connection stage).