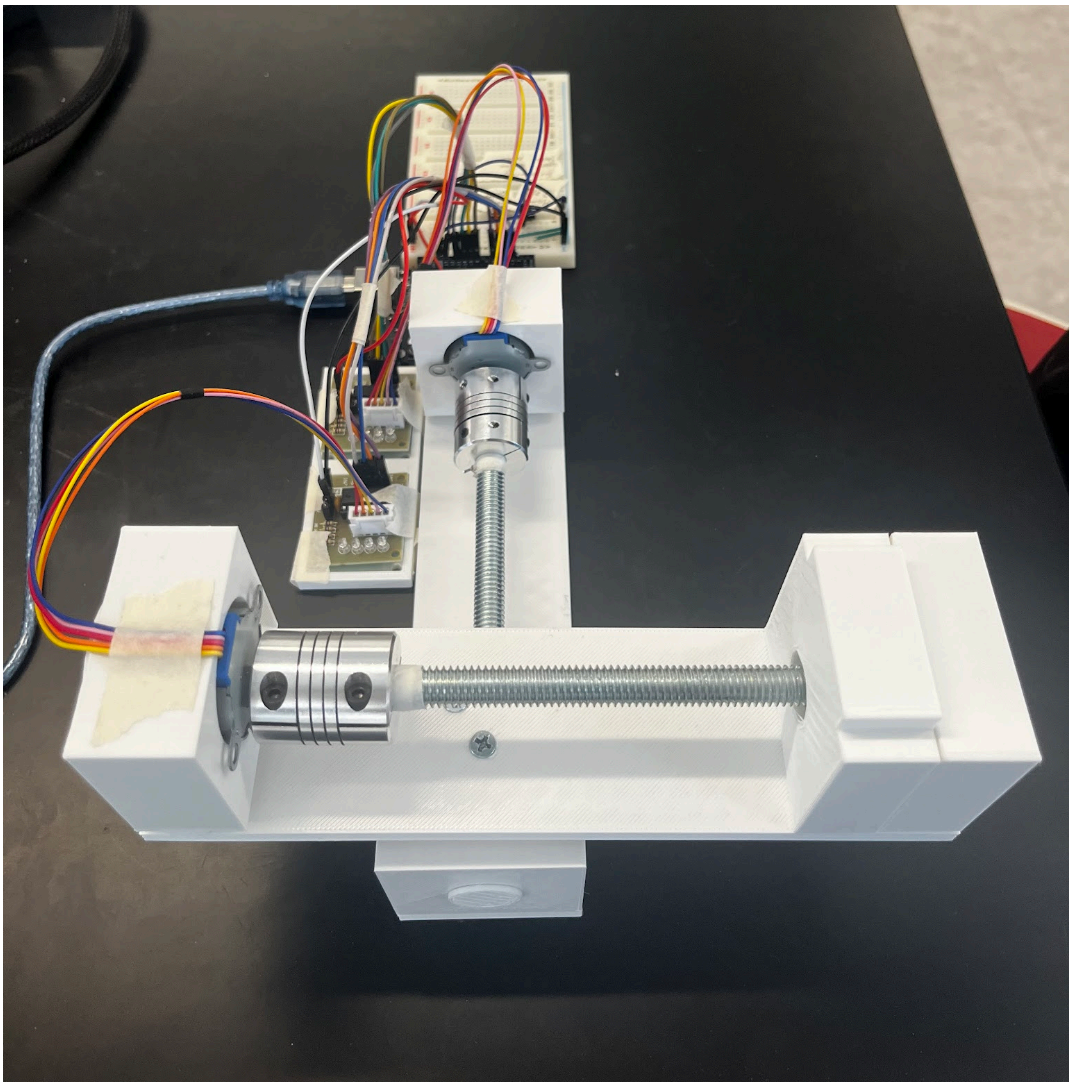
DIY Digital Microscope System

User Manual



CONTENTS

OVERVIEW

Microplastics are a prevalent pollutant that can cause serious health conditions in both humans and animals. This microplastics microscope aims to make research on this environmental threat more accessible by providing a low cost, user-friendly option for citizen scientists.

This design implements easily accessible materials as well as 3D printable parts assembled together to form a cheap solution for integrating platform movement into a digital microscope. By downloading our computer vision program you could also implement image processing and analysis.

SUPPLIES

PLATFORM MOVEMENT

- MECHANICAL PARTS:
- Threaded Rod x 2
- Hex Nuts x 2
- Wood Screws or Hot Glue
- 5mm to 10mm Coupler

- ELECTRICAL PARTS:
- Breadboard
- Arduino Uno ELEGOO
- Simple Buttons x 2
- Wires
- Stepper Motors & Drivers

BODY (3D PRINTABLES)

- Anchor (Motor) x 2
- Anchor (Rod) x 2
- Base x 2
- Connector x 2
- End Cap (Motor) x 2 (OPTIONAL)
- End Cap (Rod) x 2 (OPTIONAL)

MICROSCOPE

- DinoLite Handheld Digital Microscope
- HOWEVER YOU DECIDE TO HOLD UP THE MICROSCOPE

TOTAL COSTS: ~ $192

SOFTWARE INSTALLATION

For the body of the platform movement mechanism you can print out all necessary parts by following this OnShape link:
https://cad.onshape.com/documents/e72b41258d8db29a654f51c0/w/662ee55716d57352caf9fe64/e/9e6fdf051ceebceee81b5664?renderMode=0&uiState=680a4ab988ce27211b37d235

To control the platform movement mechanism, you have to install a couple arduino code files. These will be tuned to our stepper motors and may not be perfect in every scenario. This being said, there are three main files that you can use to properly move the platform: The Movement Scripts, The Manual Movement Script, and The Testing Script. The Movement Scripts will run the necessary movements allowing for a scanning motion to be achieved. The Manual Movement Script is necessary so that the user can manually home the platform if there are any mishaps. The Testing Script can be used if the user is noticing issues with the movement script. This script will test one full revolution of the stepper you are using and you can then see how far the platform moved. This value must be placed in The Movement Script, so that you can get the most accurate movements. Find the code here: https://github.com/JameMake/Platform-Movement-Arduino
There are two options for platform movement, the time based interval movements or the distance based interval movements. The time based will move the platform for some amount of time and is easy to set up with our image acquisition software, while the distance based will move the platform for some distance and then will pause for imaging. This may be better for you if you have your own way to acquire the images or to view your sample. For our cases use the time based movement script.

Next we need to set up the actual program to run microplastics analysis. Go to the https://github.com/ikessler21/Microscope-Pipeline/tree/master on GitHub. To make this process easier, I used Visual Studio Code to run the program, however, you can use a different environment or forgo an environment altogether and run the program on the terminal. If you are newer to coding, I would recommend Visual Studio Code as it is a small application, easy to navigate, and has extensive online resources for help.

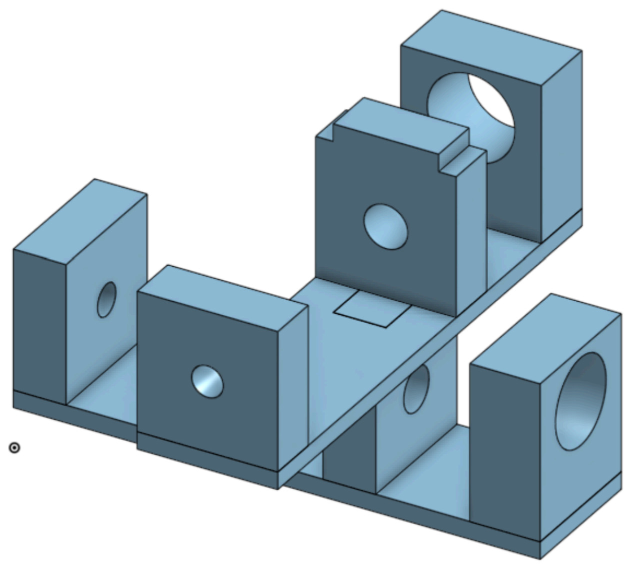
Use the same GitHub repository that the camera holder file is found in, and clone it to your computer. From there, use the README file to set up the application.

NOTE: For the scanning scripts, the stepper motor's speed is set to 15. You may be able to get away with using up to 17 without the motor skipping, but 15 is recommended to get best results as we found it to be most consistent. Also be sure to adjust the timing of the movements and pauses to match that of the image capturing program, so that the pictures are being taken at the proper time. For larger work areas, generally a larger movement time is appropriate and will allow for quicker image acquisition.

ASSEMBLY

MECHANICAL ASSEMBLY

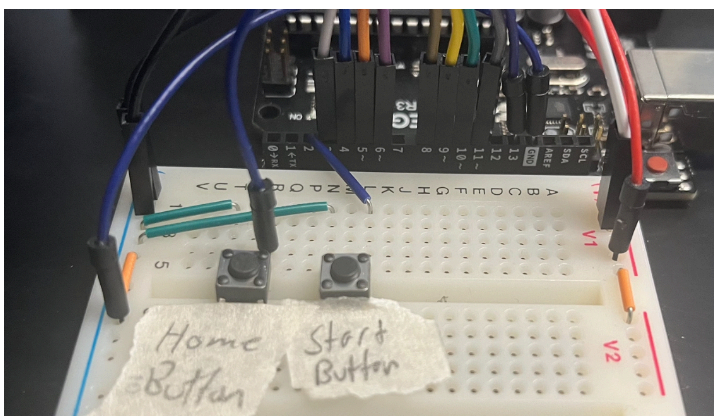
1. After printing out 3D printables and acquiring the rest of the parts you are ready for assembly
2. Use the wood screws or the hot glue to put the parts together as shown below:



3. Now, put together the stepper motor mechanism. Place the 5mm end of the coupler on the shaft of the motor and connect the 10mm end to the threaded rod. Tighten the coupler so that there is minimal slipping.
4. Place your hex nut inside of the connector block and thread it onto the rod enough so that you can fit the stepper motor mechanism between both anchors.
5. Put the stepper motor into the motor anchor with the wires facing up, this is necessary because the orientation of the motor will determine how straight the rod is.
6. Put the rod into the anchor on the other side of the base. If you are using end caps you can now place them on the ends, they should snap into place, if not then you can secure them with hot glue.
7. Repeat this process for the second module.
8. Once you have two stepper motor modules, you will now stack one on top of the other by securing one to the other's connector block. This could be done with wood screws or with hot glue.

ELECTRICAL ASSEMBLY

1. Start with the stepper motor drivers, plug in each motor into the slot on the drivers.
2. In our case we used pins 3-6 (Upper Level) and 8-11 (Lower Level) to wire the drivers to the arduino. This is adjustable in the code if you would like. Wire these so that In1 is connected to the highest pin for each level and it goes down the line as such. For example the lower level would look like this (In1 to 6, In2 to 5, In3 to 4, In4 to 3).
3. We need to power both stepper motors however the arduino only has one 5V pin. Because of this we will use the breadboard to share the voltage amongst the two stepper motors. Bring a wire from the 5V source to the breadboard and then connect it to the positive end on the driver. The same technique could be used for the steppers to share a ground, however is not necessary because the arduino has plenty of ground pins on the board itself.
4. Lastly place the buttons into the breadboard. In our case we used pins 2 (Start Button) and 13 (Home Button), however once again this could be changed in the code if you would like. Wire the button so one end goes to the pin on the arduino and the other goes to a GND.



5. Now you have your finished platform movement mechanism for the digital microscope!

OPERATION

For operation of this system you must ensure that the movement mechanism is working properly and is synchronized with the image acquisition software.

OPERATION OF THE MOVEMENT MECHANISM

The two buttons which you set up in the ELECTRICAL Assembly will be used to control the platform. Depending on which script is actively uploaded to the arduino, the buttons will do different things. For the scanning sequence scripts there is a start button and a home button. The start button will begin the scanning process and the home button will interrupt the scanning process and bring the motors back to their homed position. For the test script, each button is assigned to a particular motor. When a button is pressed it will activate the corresponding motor, to reverse the direction of movement, hit the button again, to stop the motor, hit the button a third time. Lastly, for the distance testing script the button will run the motor for one revolution, allowing for you to measure the distance and record it in the scanning scripts (may not be necessary if using the same set up as our system as we already have this value in the scanning scripts)