Tablet 1DOF

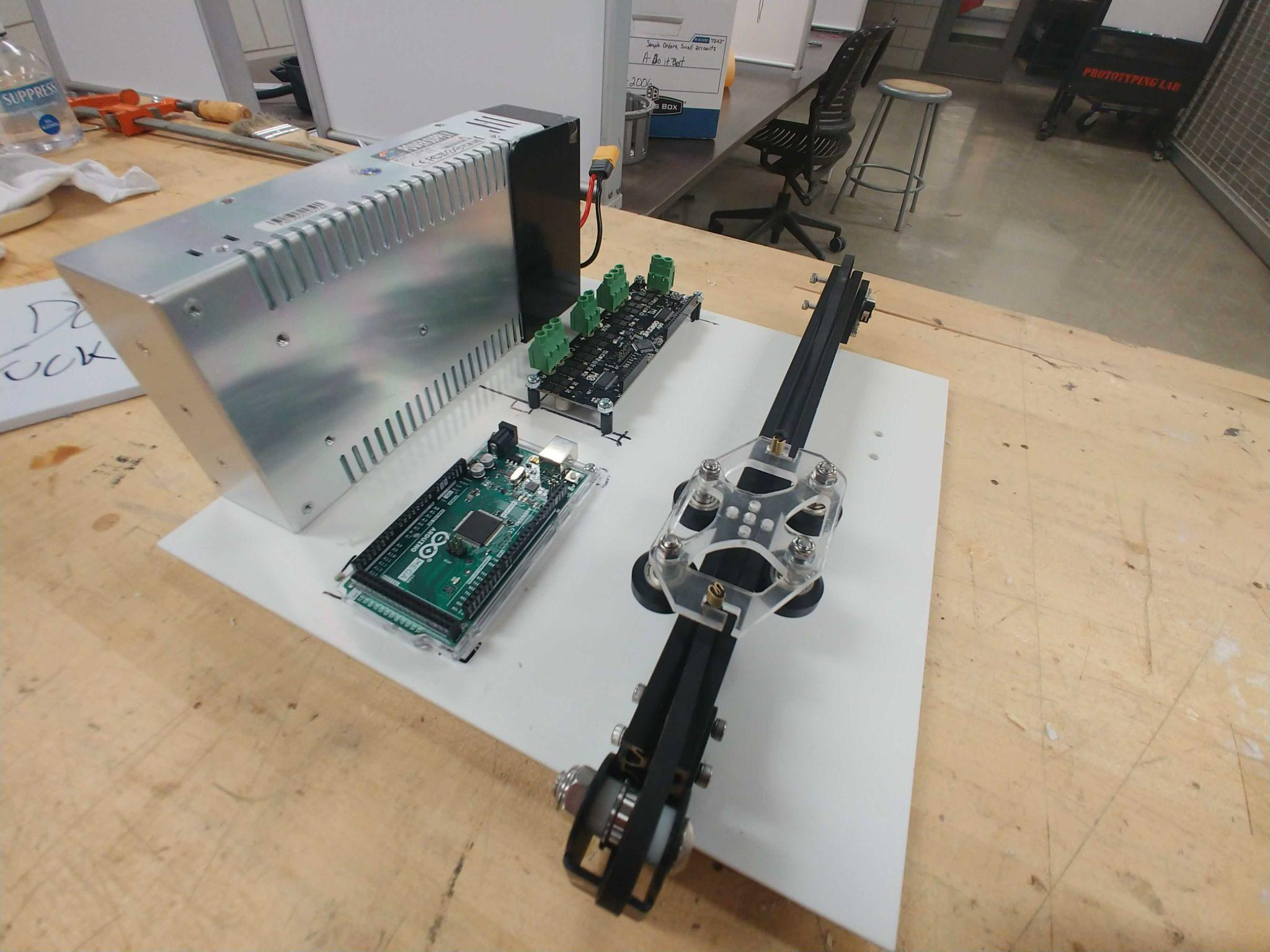
Part 3: Build Minimum Viable Prototype

Lio Liang, Yujui Chen, Cam Mastoras

1. Prototype:

In our 1-DOF prototype, we took the power supply, limit switch, and y-axis gantry from ender3. We use odrive as our motor drive, arduino mega (we might change to raspberry pi 4) as our microcontroller, and BLY172D-24V-4000 motor by Anaheim Automation which combines with AMT102-V encoder.

* Picture:



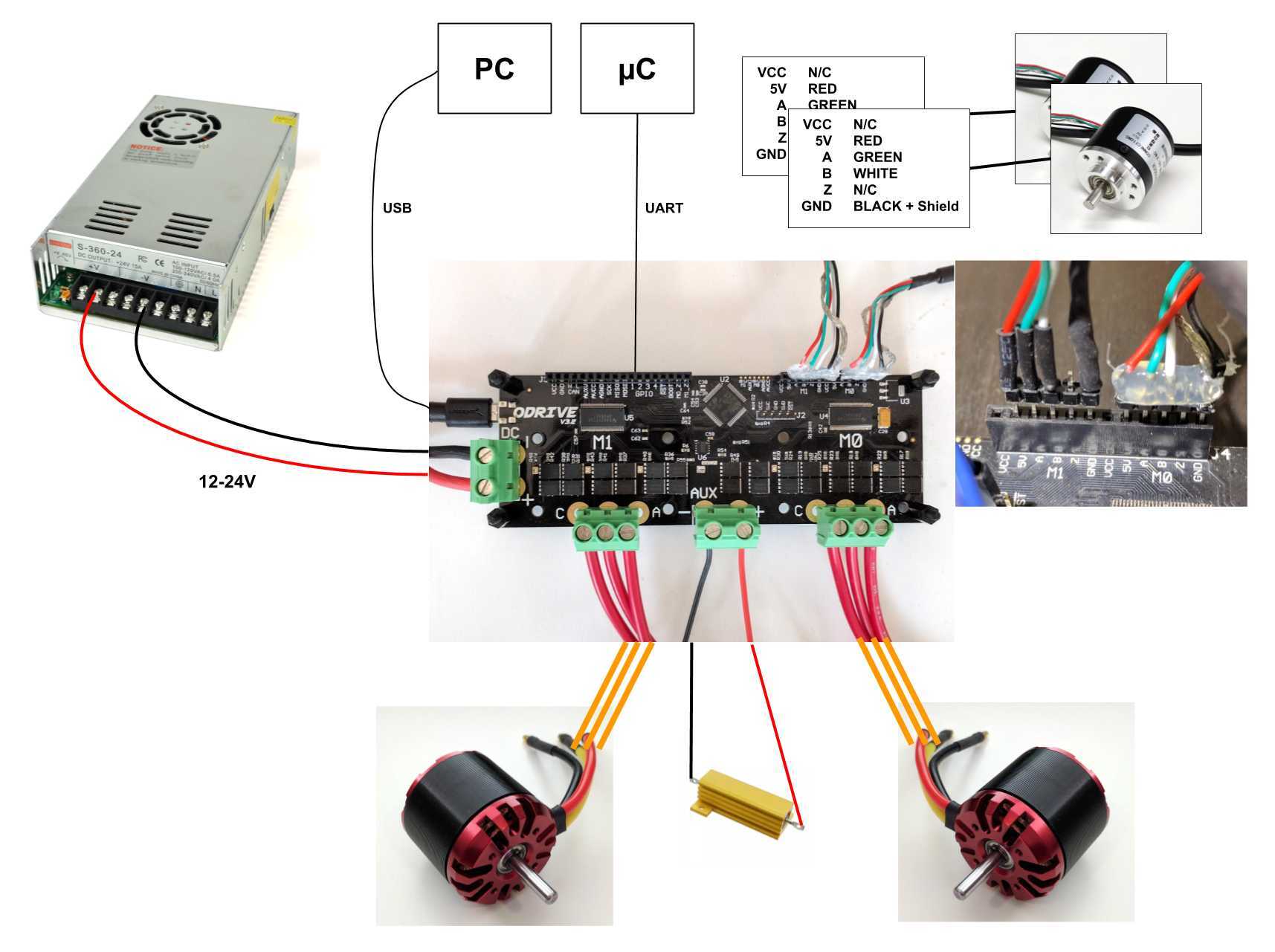
* Demo video link: <https://drive.google.com/file/d/1Rv8hUu46M2jTLNudL9wZwz2g9WoLOVMV/view?usp=sharing>

1. Bill of Materials (BOM):

|  | Amount | Price/each | Subtotal Price |
| --- | --- | --- | --- |
| DC-Powered Electromagnet | 1 | 51.16 | 51.16 |
| Motor from AnaheimAutomationpart number: [BLY172D-24V-40](https://www.anaheimautomation.com/products/brushless/brushless-motor-item.php?sID=143&pt=i&tID=96&cID=22)00 | 2 | 92 | 184 |
| Arduino Mega | 1 | 34.88 | 34.88 |
| uxcell 24V 25N Electric Lifting Magnet Electromagnet | 2 | 11.99 | 23.98 |
| Corrosion-Resistant HTD Timing Belt Pulley | 2 | 6.04 | 12.08 |
| 3D Printer Filament | 1 | 30.99 | 30.99 |
| Ball End Hex Key Wrench Set | 1 | 13 | 13 |
| Tapered Heat-Set Inserts for Plastic pack of 100 | 1 | 15.97 | 15.97 |
| Clear Impact-Resistant Polycarbonate | 1 | 38.61 | 38.61 |
| 3D Printer POM Wheel Plastic Pulley | 1 | 10.99 | 10.99 |
| Total price |  |  | 415.66 |

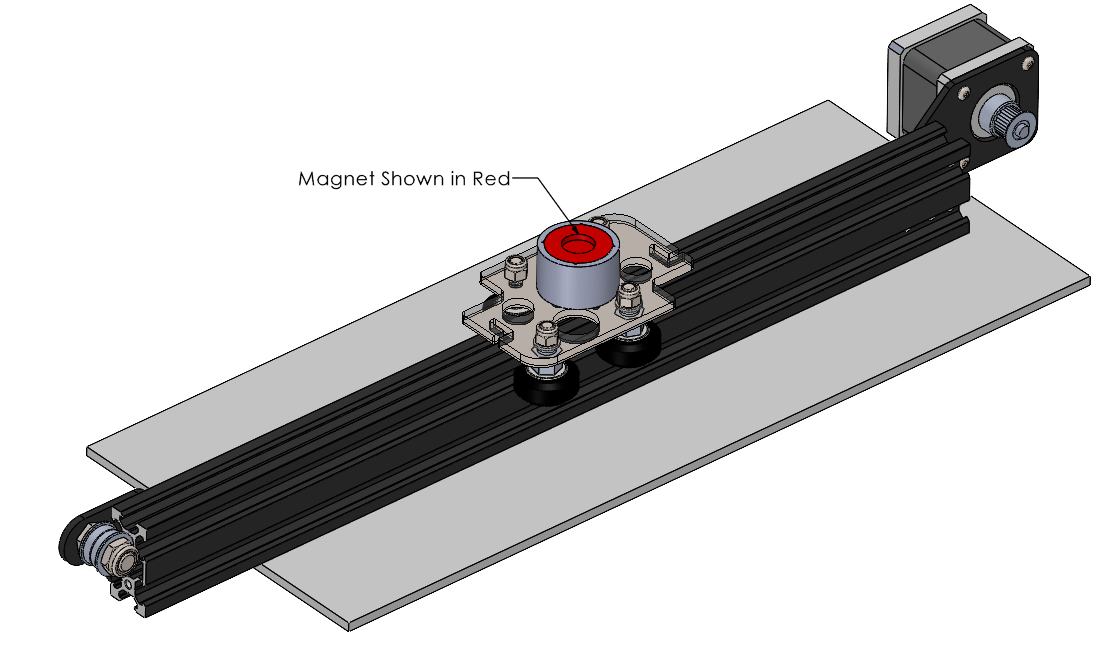
1. Wiring diagram:

For odrive, we connect the odrive with the power supply, motor, and 50W2RJ resistor. The wiring diagram shown in the below. We connect the phase A,B, and C of the motor to odrive M0 and connect the ground, 5V, A, B, and X pin of the encoder to odrive with corresponding holes. We can connect the Arduino with GPIO pins on the odrive which are pin 1 (UART\_A.TX) and pin 2 (UART\_A.RX). Finally, we connect the odrive the PC or connect the microcontroller to PC.

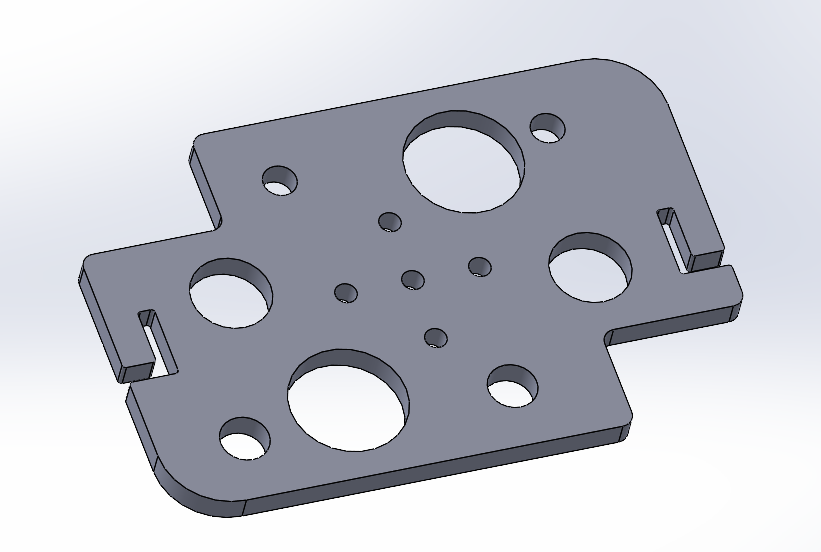


1. Design of platform:

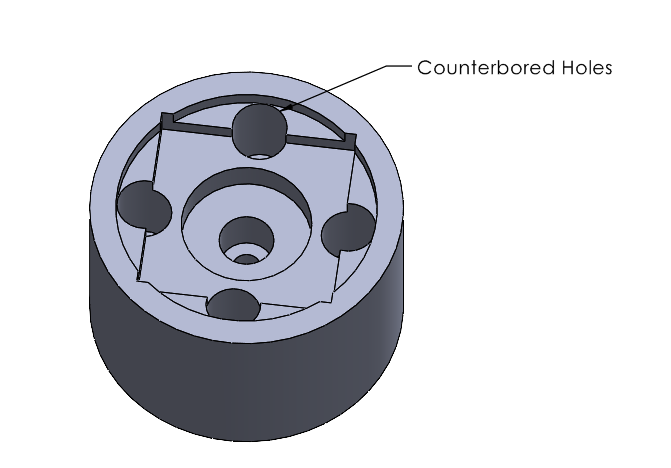
We chose to use the Y gantry from an Ender3 3D printer, which presented us with many advantages. It saved us the work of having to design an entire gantry system, and was also quite easy to source. We swapped out the stock motor with one that was more suited to our needs, which was pretty painless because we were able to find a motor of the same standard mounting size. Additionally, our other main modification was to swap out the gantry plate, which holds the rollers and attaches to both sides of the belt, because the stock aluminum plate was too large and too heavy. We swapped it out for a design consisting of a thin sheet of steel(0.06”) underneath a thicker(0.15”) sheet of acrylic. This was to get close to the geometry of the original plate while decreasing weight



Shown below, is our gantry platform that can fix the belt at two sides, fix the four POM pulleys at the coroner which will attach to the linear rail, and five small holes in the middle which we can install our magnet with a specific height. We install heat-set inserts into the holes in order to attach the magnet carriers to the plate.



Finally, we also created a magnet holder with the capability to fit a number of magnets so that we could experiment with how different sizes felt on the gantry and with the controller



1. Configuration instructions for odrive:

* odrv0.axis0.motor.config.current\_lim = 10
* odrv0.axis0.controller.config.vel\_limit = 2
* odrv0.config.enable\_brake\_resistor = True
* odrv0.config.brake\_resistance = 2.0
* odrv0.config.dc\_max\_negative\_current = -1
* odrv0.axis0.motor.config.pole\_pairs = 4
* odrv0.axis0.motor.config.torque\_constant = 0.035
* odrv0.axis0.controller.config.vel\_gain = 0.05
* odrv0.axis0.controller.config.vel\_integrator\_gain = 0.015
* odrv0.axis0.controller.config.pos\_gain=15 = 20
* odrv0.axis0.motor.config.motor\_type = MOTOR\_TYPE\_HIGH\_CURRENT

1. Comments of source code:

* Odrive commands:

Input mode:

**axis.controller.config.input\_mode = INPUT\_MODE\_INACTIVE**

#Disable inputs. Setpoints retain their last value.

**axis.controller.config.input\_mode = INPUT\_MODE\_PASSTHROUGH**

#Pass input\_xxx through to xxx\_setpoint directly.

#Valid Inputs:

#input\_pos

#input\_vel

#input\_torque

#Valid Control modes:

#CONTROL\_MODE\_VOLTAGE\_CONTROL #CONTROL\_MODE\_TORQUE\_CONTROL #CONTROL\_MODE\_VELOCITY\_CONTROL #CONTROL\_MODE\_POSITION\_CONTROL

Control mode:

**axis.controller.config.control\_mode = 2**

#CONTROL\_MODE\_VELOCITY\_CONTROL

**axis.controller.config.control\_mode = 3**

#CONTROL\_MODE\_POSITION\_CONTROL

* + odrv0.axis0.requested\_state = AXIS\_STATE\_FULL\_CALIBRATION\_SEQUENCE
  + odrv0.axis0.requested\_state = AXIS\_STATE\_CLOSED\_LOOP\_CONTROL
    - odrv0.axis0.controller.input\_pos = 1
* Odrive connect Arduino Mega 2560
  + Change baud rate on the serial monitor to 115200
  + Type the number “0” or “1” to begin motor calibration of motors 0 or 1.
  + Once Calibration is complete the motor is ready to be used in closed-loop control.
  + Serial commands are, “s”: start sinusoidal motion, “t” stop motion, “u” set speed, “a” set amplitude.

1. Sensor data:

The frequency of the control loop is 600Hz.

