# Syringe Control System V3.1 3D-printing Instruction

## 1. The File

Click the GrabCAD link down below to access the STL files that need to be 3D-printed. In the same link, you'll also find the complete CAD assembly in SolidWorks as well as in STEP format:

https://grabcad.com/library/linear-syringe-system-for-controlling-soft-robot-actuators-1/details?folder_id=8379164

This project only requires a standard FDM 3D-printer like the Prusa i3 MK3S: https://www.prusa3d.com/. If you happen to have the exact same printer, you can use the .3mf files and the G-code files I provided in the same GrabCAD link to print the parts directly and skip part 2 of this instruction.

## 2. The Slicer

For one syringe unit, you need:

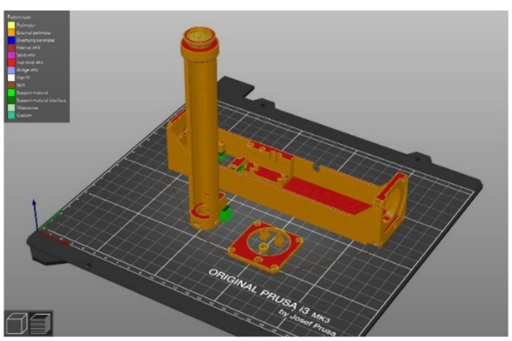
- 2 x button end
- 1 x housing bottom
- 1 x plunge
- 1 x syringe clamp
- 1 x tube clamp

For the master unit, you need:

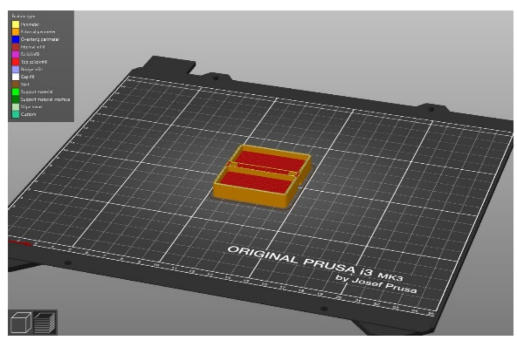
- 1 x master stand

In the slicer software of your choice, load the STL files. With the Prusa printer, I used 0.2mm layer height and 15% infill density for PLA/PHA filament. If you are using standard PLA, it is recommended that you increase the infill density to about 30% for better part strength.

The parts for the syringe unit are slightly more challenging to print, so we'll focus on that in the instruction. In your slicer, try to orient the parts similar to the pictures below. For support, select the option "on build plate only" if possible. Brims and rafts were not used for the Prusa printer, but if the print keeps failing on your 3D-printer, consider adding either one of them to the G-code to see if they help.

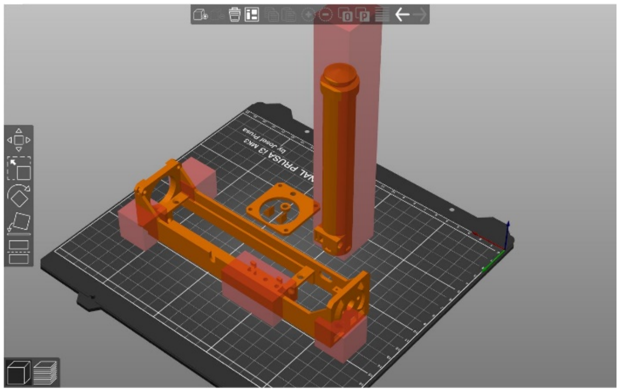


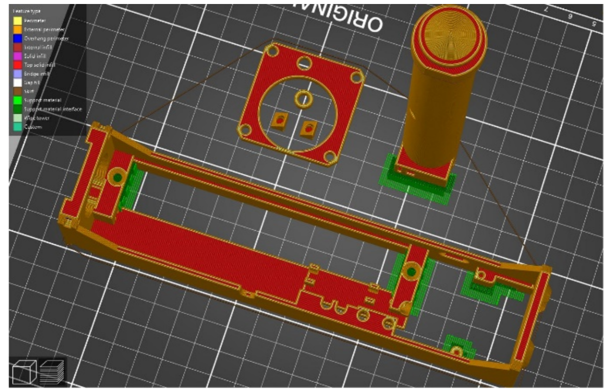

Syringe Unit




Master Unit

If your slicer software has the option, adding support blockers to the following regions will save your time and effort in removing supports after the prints are done. For the Prusa i3 MK3S, the final G-code only contains support on the 5 essential regions shown in the pictures below.





3. **During the Print**

Because of the geometric complexity of the 3D models, the first few layers are crucial for a successful print. Ideally you should stay with the printer to make sure that it has finished layer #1 without a problem. Hair spray and heated print bed will significantly improve bed adhesion.

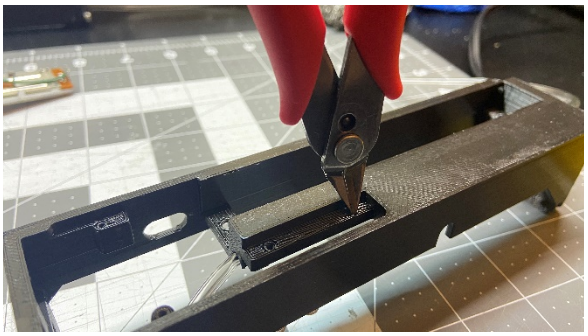
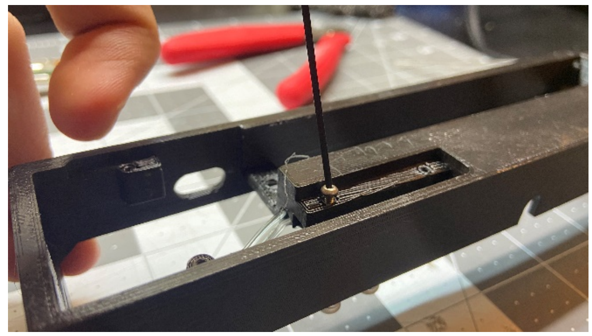
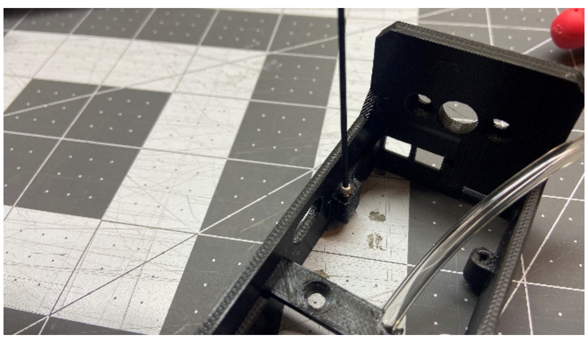
4. **Taking off Supports**

Use the precision plier and the flush cutter to remove support from the print. You can also use a piece of rough sandpaper (120 grit – 320 grit) or a small file to remove any remaining supports. Pay close attention to the supports on the base where screw holes might be blocked by support materials and become inaccessibly during hardware assembly.

5. **Checking Print Quality**

After removing supports from your print, the final check before moving to hardware assembly is to check if the print quality. Specifically, you'll need to check the holes at the bottom of "housing bottom" and see if you can "tap" an M2 machine screw in. You might use the flush cutter to create a "countersink" on the holes will make it easier to put in the screws. When you're done, do the same thing with the M3 holes in "button end".

Remember not to over-tighten the screws! Turning the screws too much could end up breaking the threads in the PLA parts. Hopefully in the next version I will replace the self-tapping holes with nuts and bolts.


# Syringe Control System V3.1 Custom PCB Instruction

1. **Choosing your Manufacturer**

   The custom PCB was made with the purpose of improving the wire management of the system. The design is so simple that if you happened to have a PCB milling machine like the Othermill from Bantam Tools, you could probably make it yourself.

   That said, for a large quantity, it is recommended to use one of those online PCB manufacturing services to do the job for you at a low price. For faster service I recommend OSH Park (https://oshpark.com/).

   Download the GitHub repository and navigate to the "PCB" folder, in which you will find a .zip compressed folder with the name similar to "05_04_2020_gerber.zip".

   When you are on their website, click "Browse for Files" and select the .zip folder. Follow the prompt on their website and select the type of service you'd like your board to be manufactured. The standard PCB option is $22.70 for 3 pieces without shipping.

   Alternatively, you can use any online PCB manufacturing service.

2. **The KiCAD Files**

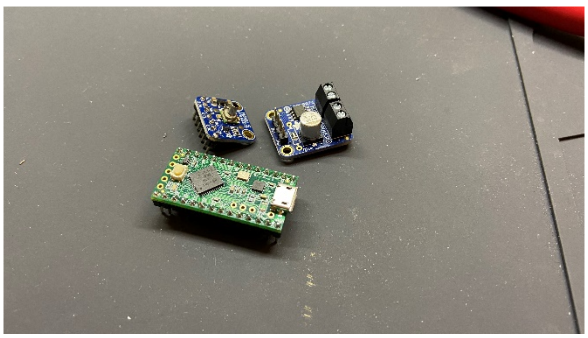
   If you would like to make any custom changes to the PCB design, I have also included the original KiCAD files within the "PCB" folder. All the symbol and footprint libraries should also be in the same folder.

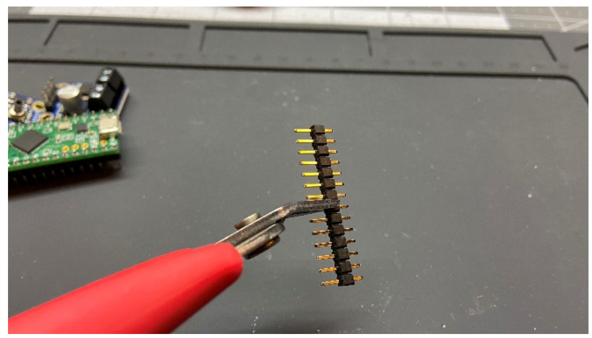
## Syringe Control System V3.1 Syringe Unit Assembly Instruction

Bill of Materials: https://docs.google.com/spreadsheets/d/1QlE3OmNmio2WvQ-pDKSaTucrADuNShoU5-5D75yimro/edit?usp=sharing
**Note:** The pictures were from an earlier version, and therefore the components might look slightly different from what you have. If you notice any inconsistency, please always refer to the most up-to-date CAD assembly and Gerber files for reference.

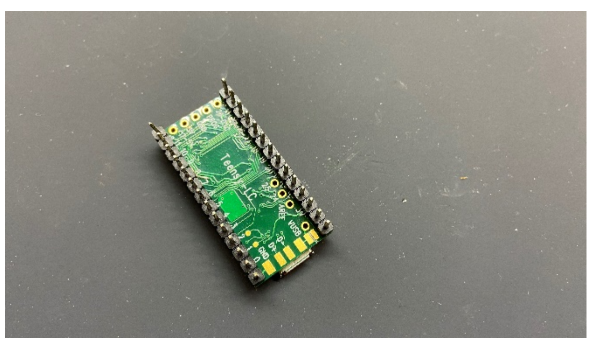
### 1. PCB Assembly

[1] Solder male headers onto the Teensy LC, MPRLS pressure sensor, and the DRV8871 motor driver. Remember to use the short headers and not the standard ones that came with the sensors. You will need to use the flush cutter to cut the pin headers to length. If possible, use the sandpaper to sand the edges of the pressure sensor and the motor driver- this ensures a better fit in later steps.
*MAKE SURE to solder the motor driver at the opposite side (see left pic)
*Tip: I use a combination of breadboard and poster putty as "helping hands" when soldering.

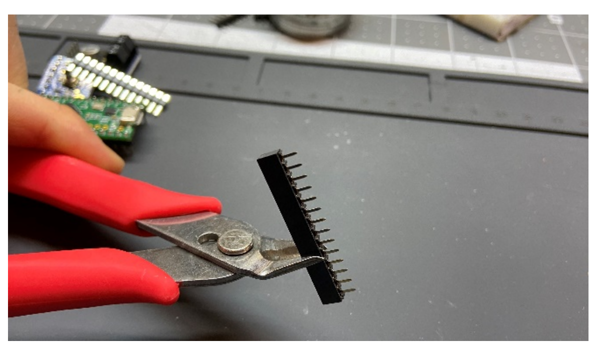




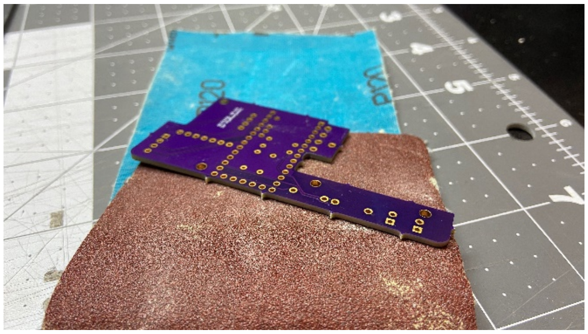
[2] Cut the 5v trace on the Teensy using an X-Acto knife. This makes sure that you can plug in the Teensy to your computer while it's getting power from the daisy-chained wires. <u>If you plan to use the unit by its own and not with a master unit, you can skip this step.</u> Refer to the PJRC website for details: https://www.pjrc.com/teensy/external_power.html



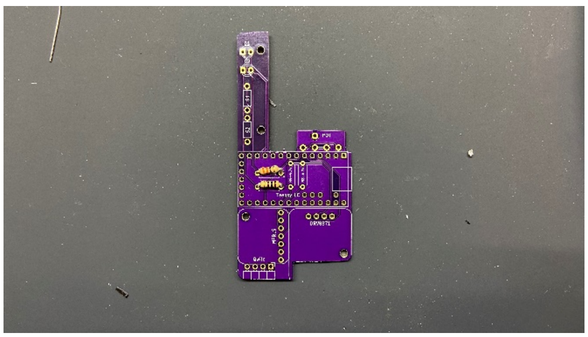
[3] Solder short female headers to the Qwiic adapter. Again, you'll need to use the flush cutter to trim the female headers to the correct length. You can then use the sandpaper to lightly deburr the header.

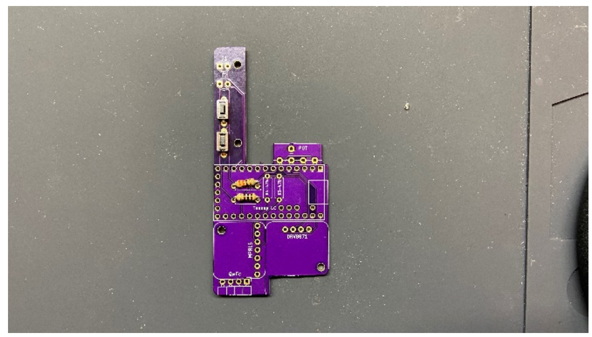


[4] Use the sandpaper to get rid of tabs on the custom PCB. Test to make sure that the board fits onto the 3D-printed base.

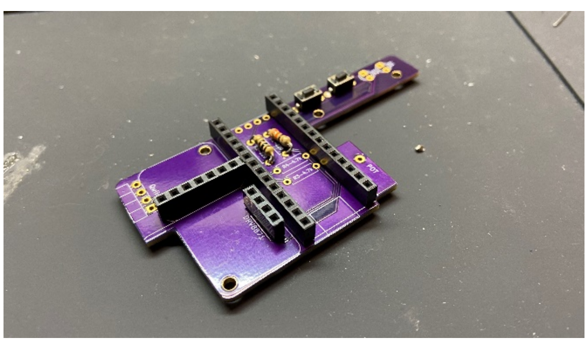


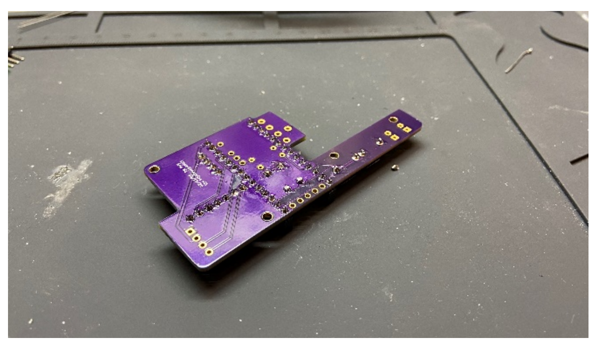
[5] Solder resistors to the custom PCB. I used 300Ω for Red LED, 100 Ω for Green LED. Solder the 300K resistor to the upper position (see pic). Solder the buttons to the Custom PCB.

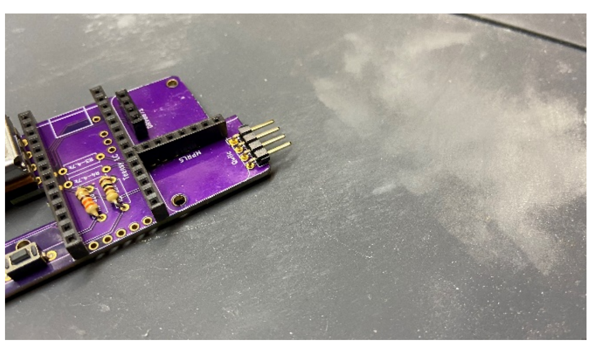




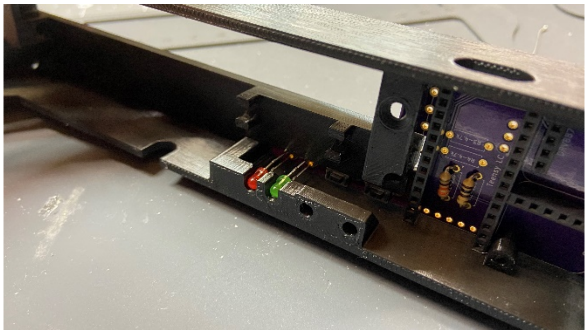
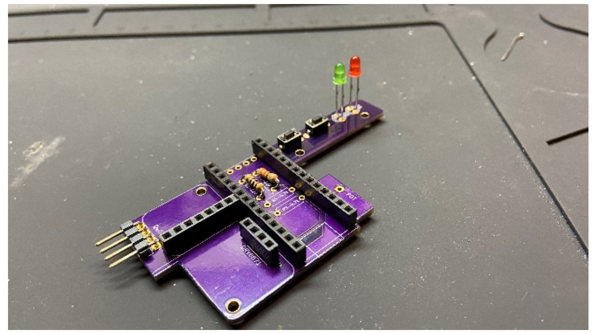
[6] Solder the female headers to the custom PCB. Use the components in place to help alignment if necessary. For longer rows, solder the ends first to ensure alignment. Solder the right-angle pin header for the Qwiic adapter.



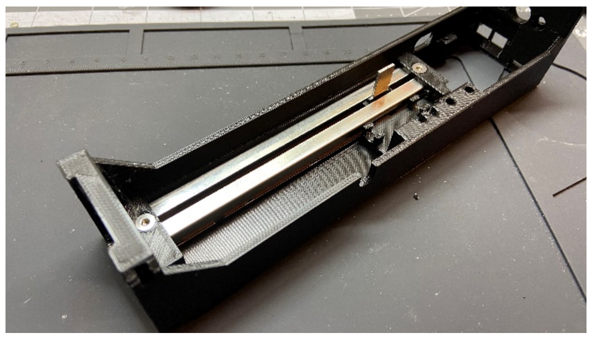
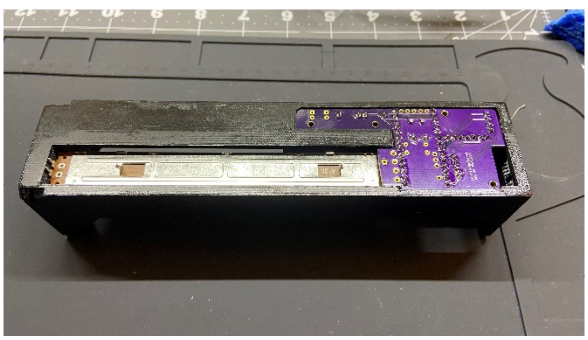
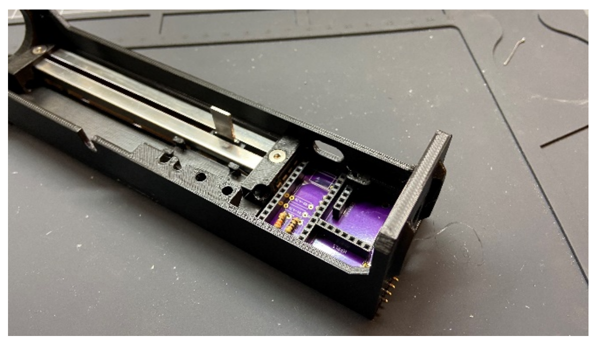




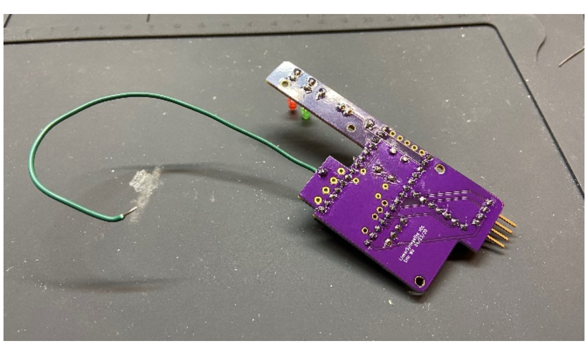
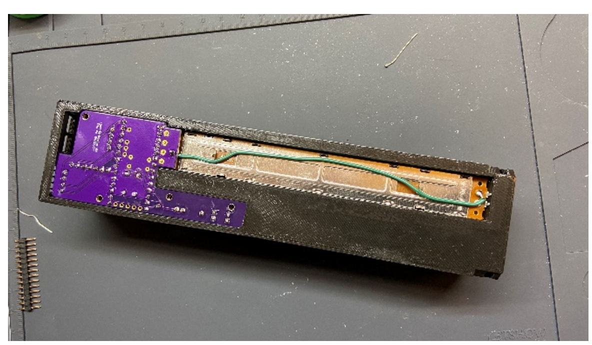
[7] Solder the LED in position. Do not solder them directly against the PCB; instead, solder them against the slot in the 3D-printed base (see pic).

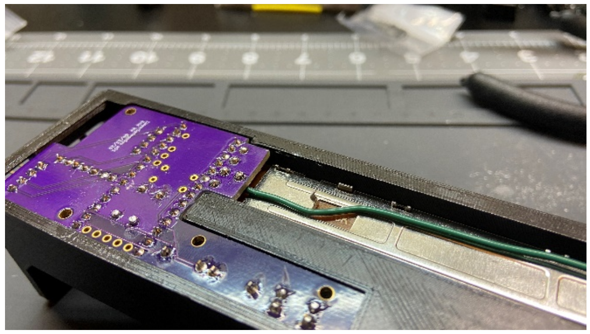
[8] Fixate the linear potentiometer to the base with two M3x5mm flat heat screws. Test fit again with Teensy LC attached (do not attach MPRLS and H-bridge at this time yet). Adjust if Necessary. DO NOT solder the potentiometer in position yet.

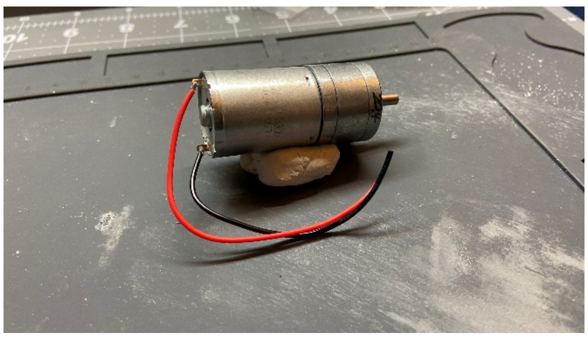
[9] Keep the potentiometer in place and take out the PCB. Prepare a 130mm-long solid core wire. Use the wire stripper to take out about 5mm on both ends of the wire and solder one end of the wire to the potentiometer hole on the PCB. Attach the PCB back to the base and solder the other end to Position 3 on the linear potentiometer.

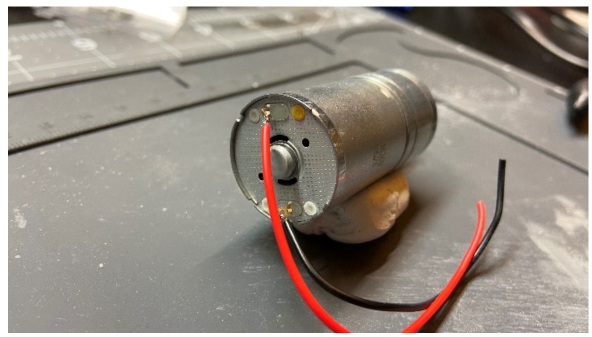



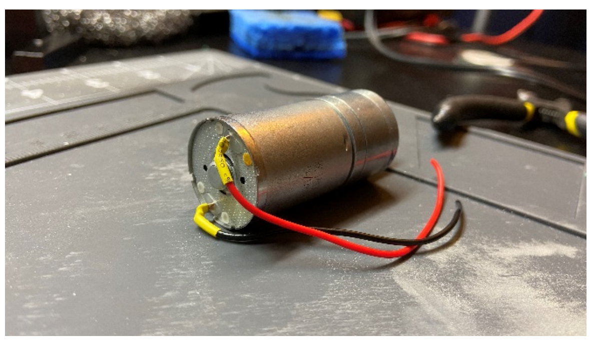
[10] Keep the custom PCB inside the base, and solder the rest of the potentiometer connections



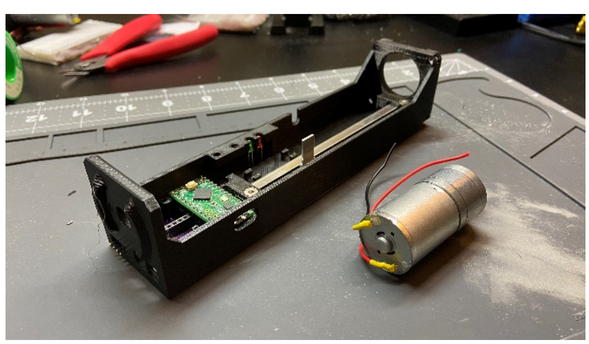
[11] Solder the wires onto the motor (Red wire – 100cm, Black Wire – 90cm). Mind the orientation of the motor and the color placement – the red wire should be with the yellowish mark on the motor. If your motors do not have the same mark, just make sure you are consistent with the orientation across all units. Finally, cover the connection with heat shrink.





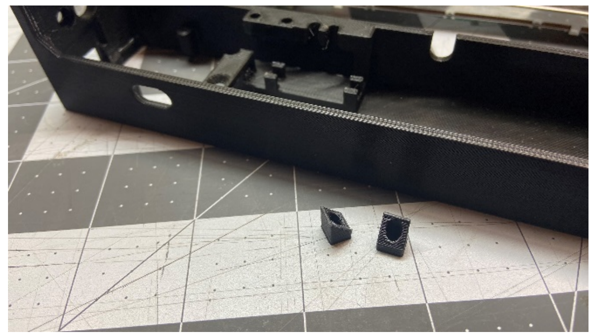


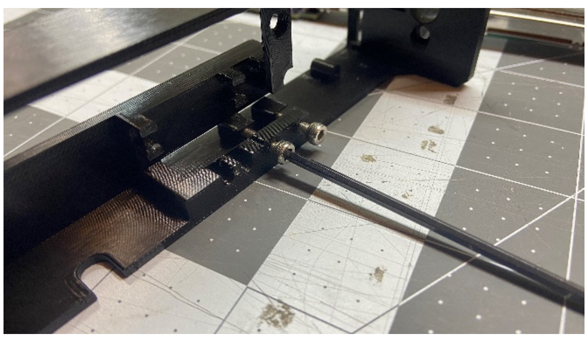
You are about halfway there! Don't worry about fixating the PCB on the 3D-printed base yet. We will need to install one more thing in the next section before we can secure the board. You are now done with the soldering iron – set it aside to make room for other components.

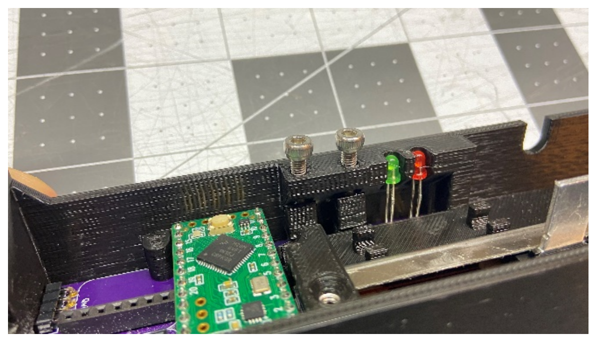


**2. Mechanical Assembly**

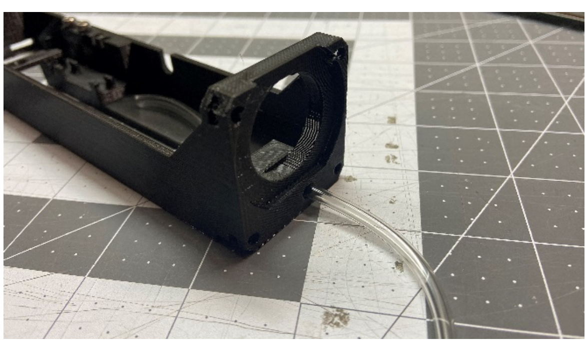
[1] Carefully take the finished PCB assembly out of the 3D-printed base. From the bottom side of the base, insert the button extension pieces. Each of the two pieces is secured from the top side using a M3x16mm socket head screw. The screws should tap themselves inside the extension pieces with some resistance.
*Remember not to over-tighten the screws – you only need a few threads to secure the parts in position. If you over-tighten them, you might risk breaking the "threads" on the PLA. See picture below for reference. You can put back in the PCB assembly to check if the button extension works and adjust the height buy fine-tuning the screws accordingly.



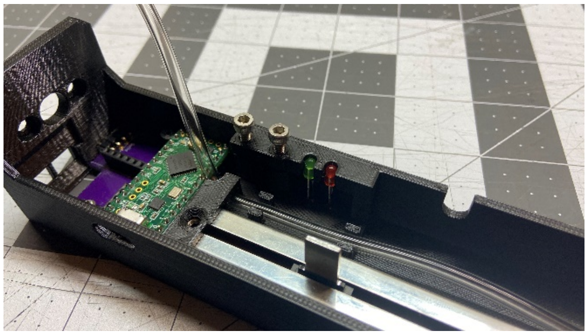
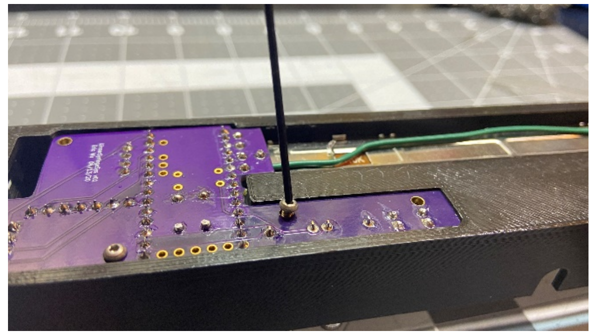
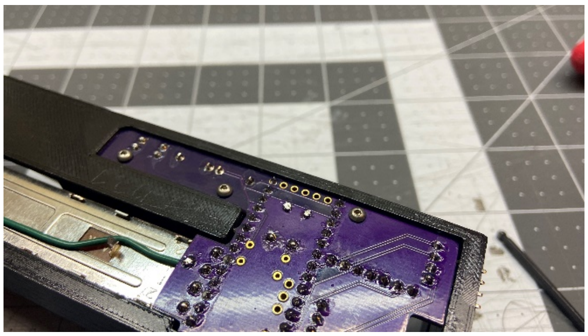
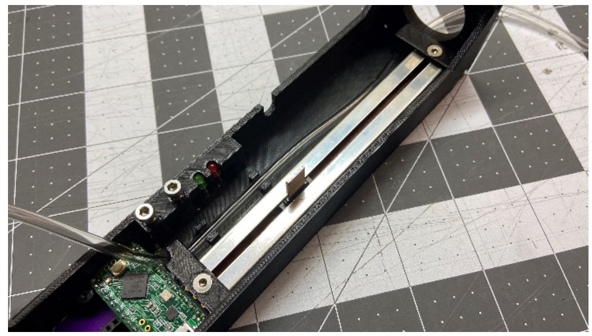




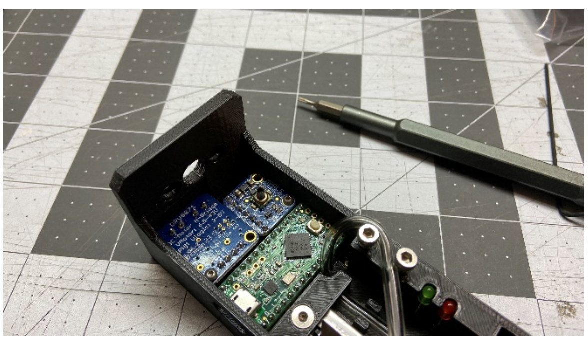
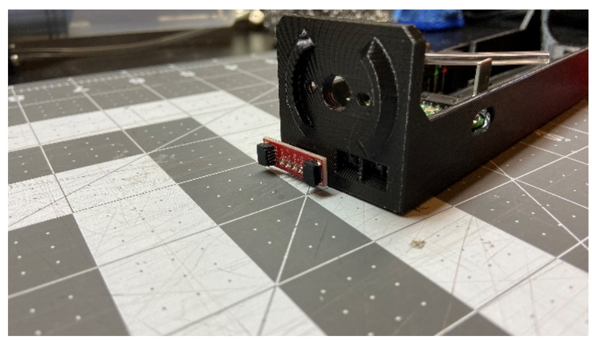
[2] Take a piece of 3/32'' ID air tube that's approximately 14-inch long. From the front of the 3D-printed base, feed in the tube through the guide slots. At the other side, leave about 2 inches out of the last slot (see pic).



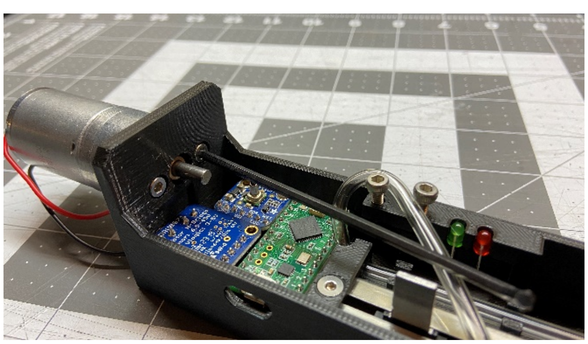
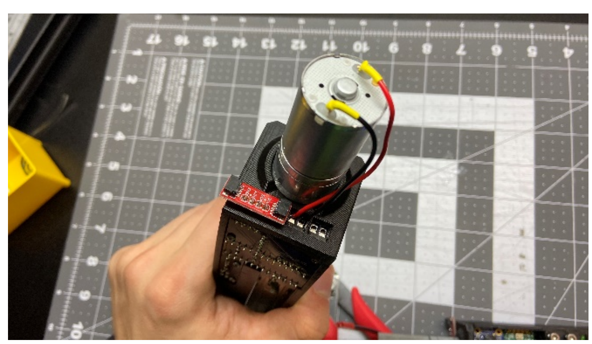
[3] Attach back the PCB and secure the board from the bottom using the M2x5mm button head screws. Secure the potentiometer with the M3 flat head screws.

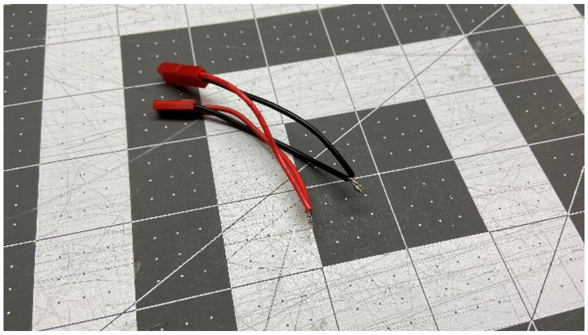
[4] Attach the pressure sensor and the motor driver and secure them to the base with M2 screws from the top. After that, attach the Qwiic adapter from the outside.

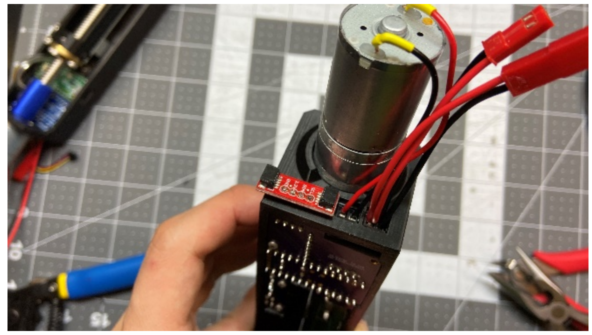



[5] Attach the motor with two M3x6mm socket head screws and insert the motor wires to the screw terminals on the Motor driver. Double-check the orientation of the motor – the red wire should be on top. Do not cut the wire to length until you've made sure your connection is correct.

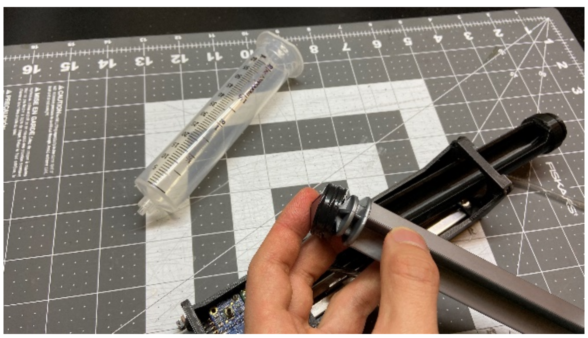



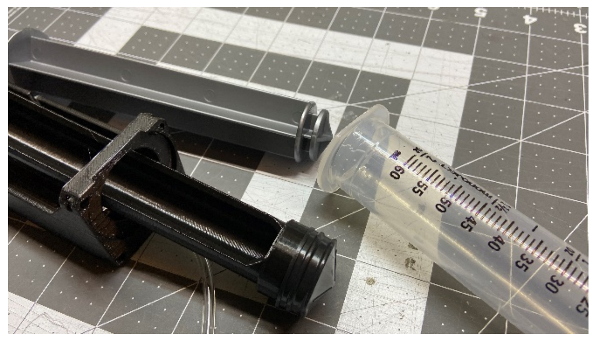
[6] Take the JST connector pair and trim the wires to about 2.5 inches long. Strip the ends of the wires and twist the positive and negative terminals together. Attach the wires to the screw terminals. This could be tricky, as the wires are a little too thick for the size of the screw terminal. If you are having difficulty, consider stripping the wires longer, using solder to fuse the wires, or using plier and cutter to rework the front of the wires.



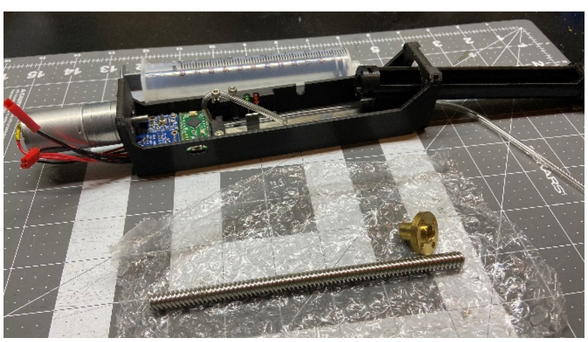


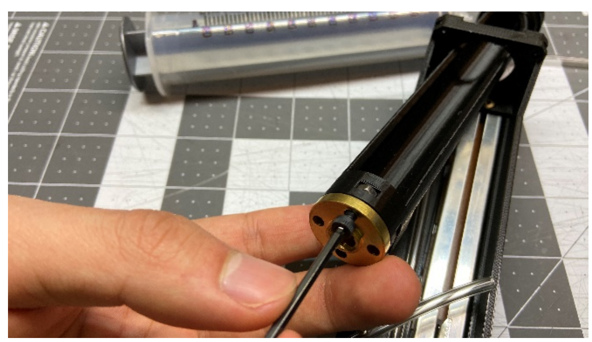
[7] Slide in the 3D-printed, custom plunger from the inside of the base; take off the rubber seal from the actual plunger and attach it to the front of the 3D-printed one.



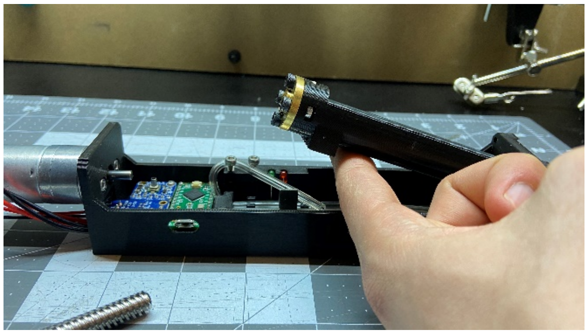


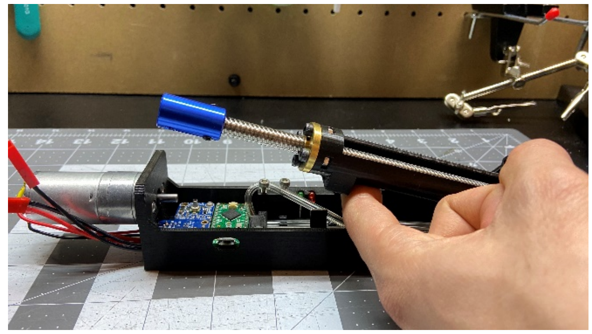
[8] Attach the lead screw nut to inner piston with M3x10mm socket head screws. Attach the lead screw to the shaft coupler. Roll the lead screw by the edge of a table to make sure the coupler is aligned and does not wobble.



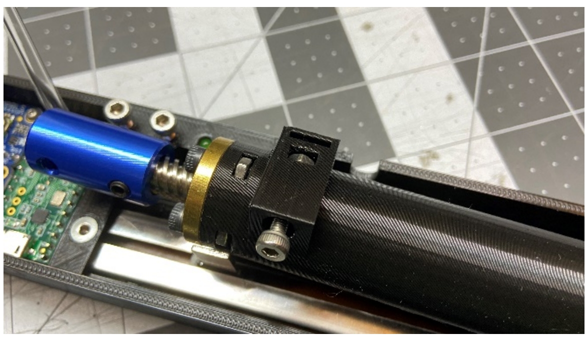


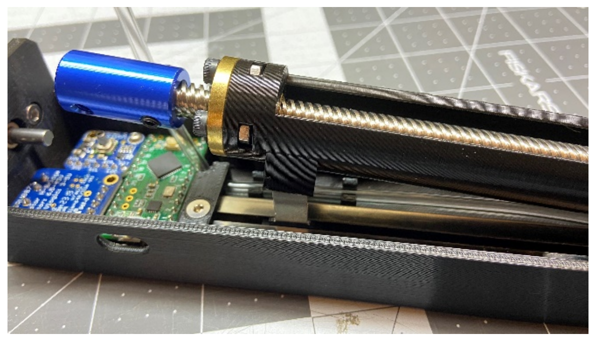
[9] Slightly lift the 3D-printed plunger and carefully feed the lead screw inside the nut.

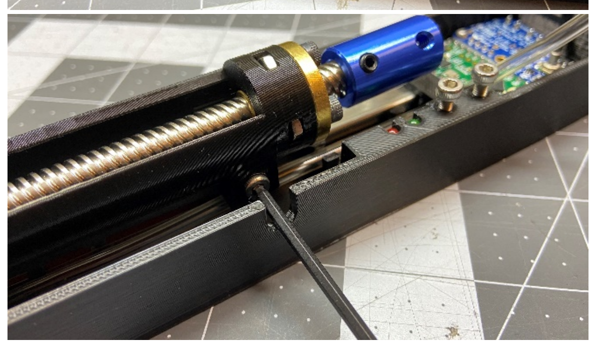




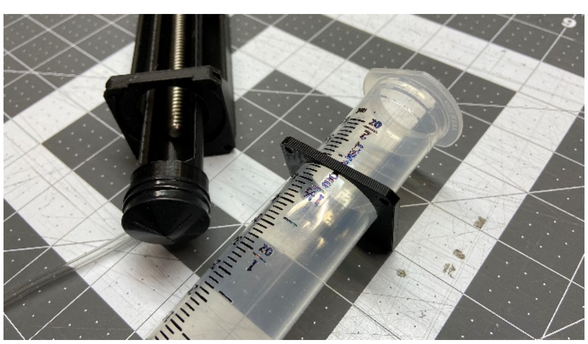
[10] Attach the plunger to the potentiometer slider. Make sure the setup could slide back-and-forth smoothly.
*As shown in the pictures below, you can put in an M3 nut and a M3x16mm socket head screw to reinforce the connection between the plunger and the slider. This is optional since I later found out that not having the screw does not affect the performance – and it saves assembly time, too. However, if your 3D-printed piece has different tolerance value, the screw might end up being quite useful.



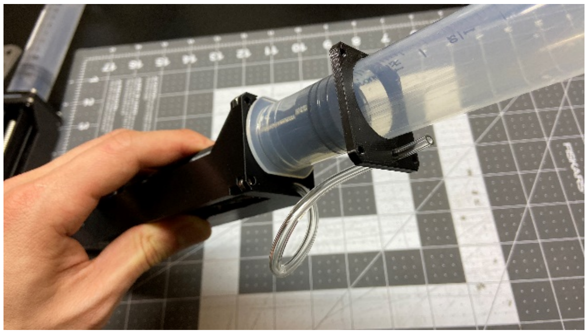
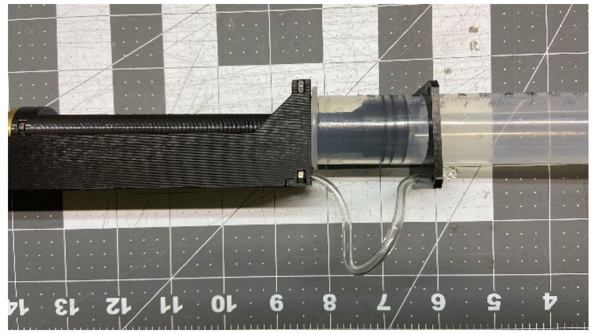
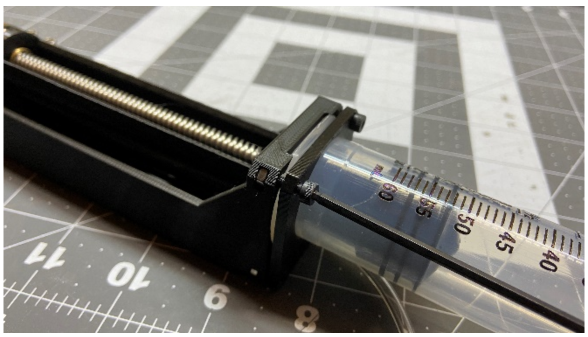
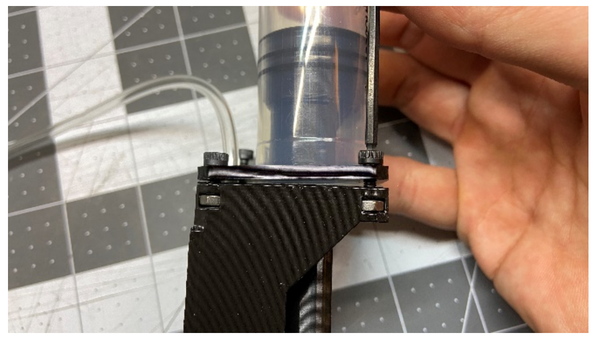




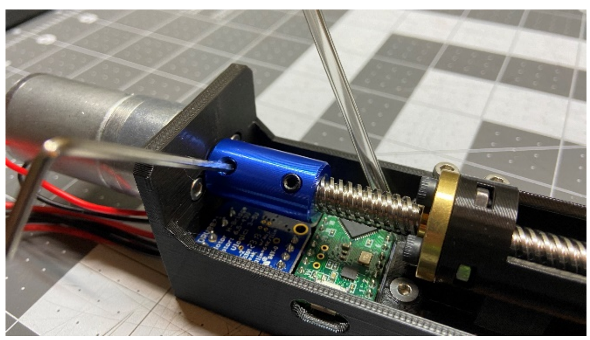
[11] Now take the syringe clamp ring and attach it to the syringe barrel from the front. Put M3 nuts inside the pre-made slots on the 3D-printed base.



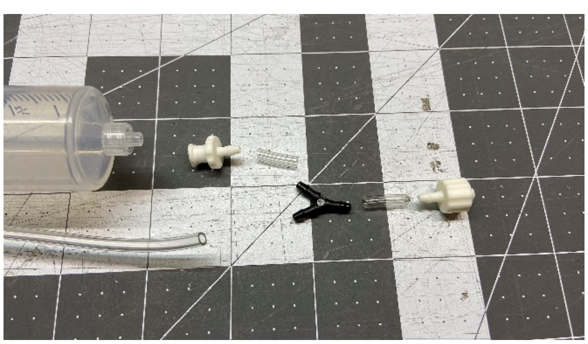
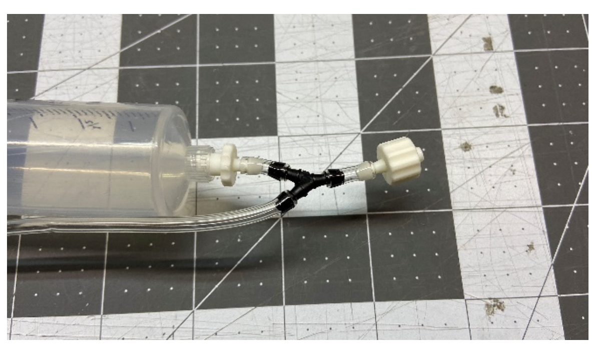
[12] Attach the syringe barrel to the main assembly and secure it to the base with four M3x10mm socket heat screws. Remember to feed the air tube through the hole on the clamp piece before securing it to the base.
*Instead of turning one screw fully in at a time, you may find it easier to alternate between all of the 4 screws to keep the clamp parallel to the base.

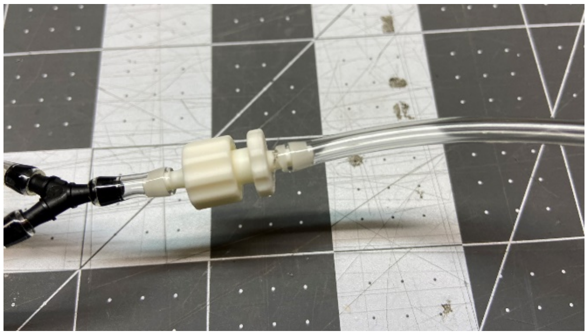
[13] Attach the other end of the shaft coupler to the motor shaft.

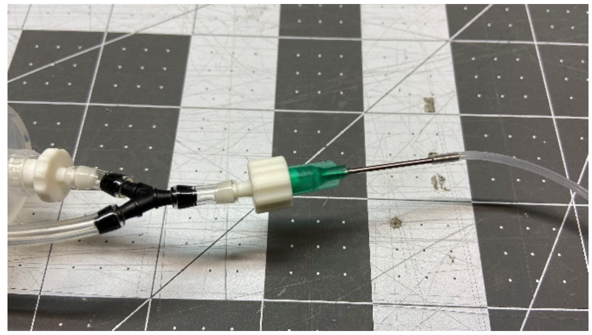


[14] Prepare two short pieces of 3/32’’ ID air tube, each about 0.5 inch long. At the tip of the syringe, attach a 3/32’’ ID luer male connector, a 3/32’’ ID luer female connector, and a 3/32’’ ID Y-connector by following the pictures shown below:

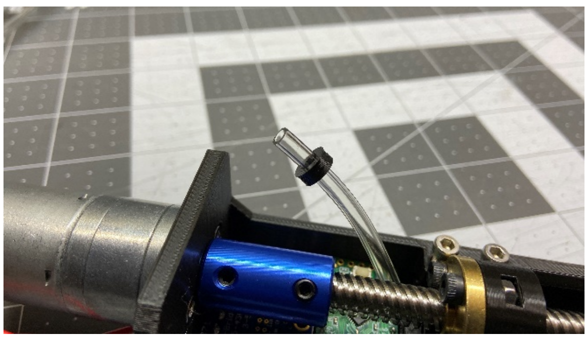



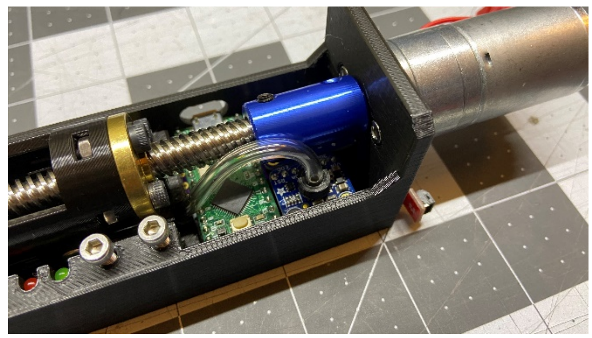
*The female luer lock goes to your actuator. Depending on the tubing on your existing system, you can either purchase a male luer lock adapter to the tube size of your choice, or simply twist on a luer dispensing tip to connect the system to a much thinner air tube.





[15] Attach the 3D-printed ring to the air tube. After you've made sure the tube is routed correctly, cut the excess tubing from the pressure sensor side. Attach the tube to the MPRLS and secure it with the 3D-printed ring.





Congratulations! You've just finished assembling a syringe unit. Now you can either go ahead and finish up making a master unit or jump straight into uploading the code and testing the unit by itself. Refer to the Software Setup Instruction for how to setup the code to run the system.

## Syringe Control System V3.1 Master Unit Assembly Instruction

Right now, this part of the instruction only contains text. Please refer to the CAD files and bill of materials.

[1] Solder male headers onto the Arduino Nano. You'll have to use the flush cutter to trim down the long headers to 15-pin long.

[2] Solder female headers onto the breadboard PCB. Again, use the flush cutter to trim the Female headers down to 15-pin long, and use some sandpaper to lightly deburr the cutting point.

[3] Take the Qwiic jumper adapter cable and solder the jumper pins to the breadboard PCBThe location of the pins are:

Black – GND
Red – 5V
Yellow – A5
Blue – A4

You can also refer to the image below for reference.

[4] Take a Male JST wire (there should be leftovers from building the syringe unit) and solder the 2 wires into the power lines on the breadboard PCB.

[5] Take your DC power supply wires and solder the positive and negative line to the same power lines. Double-check that the wires match with the JST wire you soldered in the previous step.

[6] Insert 2 M3 nuts into the slots in the 3D-printed stand. With all the components soldered in, secure the breadboard PCB to the stand with two M3x10mm socket head screws. Remember not to over-tighten the screws, as they might damage the 3D-printed piece.

[7] Attach your Arduino Nano to the soldered female headers in the correct orientation. Plug in the Arduino to your computer and plug in the Qwiic cable to a syringe unit – you should see the onboard red LED on the Teensy LC turning on, which indicates that the connection is established.

You have now finished all the hardware assembly for the SCS – congratulations! Now you can proceed to the next step, which is uploading the code to have the system up and running.

# Syringe Control System V3.1 Software Setup Instruction

## 1. Syringe Unit

Follow the instruction on the PJRC Website to set up your Teensy LC:
https://www.pjrc.com/teensy/teensyLC.html

Similarly, go to Arduino's Website
https://www.arduino.cc/en/Guide/ArduinoNano
if you are setting up an Arduino Nano as the Master Unit.

Upload "mid_master_TC" to the Teensy LC on each of your units. For every new unit, you will need to change the I2C slave address on line 41. If you are not sure what address values you can use, check out this helpful list from Adafruit: https://learn.adafruit.com/i2c-addresses/the-list

In the code, you can also tune the gain constants for the closed-loop control system in line 32 - 38. They have only been roughly tuned, but for now they seem to work with little to no problem. If you end up finding better constant values, please do not hesitate to share it with me!

```
28 // Constants
29 int count;
30 int pos_raw;
31 volatile float psi = 14.77;
32 float kp_psi = 400;
33 float ki_psi = 200;
34 float kd_psi = 0;
35 float pos = 50.00;
36 float kp_pos = 80;
37 float ki_pos = 5;
38 float kd_pos = 2;
39 volatile uint8_t received;
40 uint8_t target = 0x18; // target Slave address
41 uint8_t self = 0x68; //This should be different for each unit.
```

After you upload the code to a control unit, the unit should be operational in manual mode, i.e. after connecting the unit to 12V motor and 5V logic power, you should be able to control the unit using the built-in buttons. You should also be able to view the current pressure and positional data if you access the Teensy LC using a serial monitor on your computer.

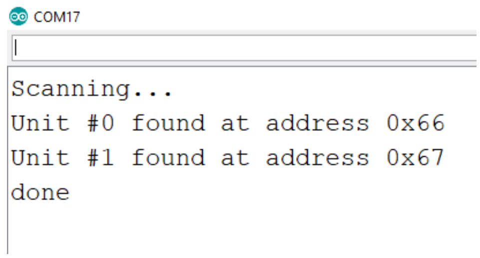
The syringe unit code has a built-in manual control loop which allows you to move the plunge back-and-forth with the build-in buttons. To use this feature without a master unit, simply connect the unit to a 12V power supply from through the JST connector and plug in the Teensy-LC to your computer. If you already cut the traces in step 2 of the PCB assembly, you will also have to supply logic power to the Teensy using a Qwiic-to-jumper connector (found in the master unit section of the BOM).

During the plung's movement, you should see the green LED lighting up; when the red LED lights up, it's either because the plunge has reached the ends of its travel range, or because the pressure of the system is exceeding the operating range of the MPRLS pressure sensor. In addition, you can also see the current pressure and positional data if you access the Teensy LC using a serial monitor on your computer.

## 2. Master Unit

Upload "top_master_nano.ino" to your Arduino Nano as the master unit. With the Arduino connected to your computer via UCB, plug in the Qwiic connector from the master unit to a syringe unit, and you should see the Teensy LC powered up.

In the example code, I have written a simple command system for basic operation and testing the daisy-chained syringe units. If you open up the serial monitor of the Arduino Nano and press the reset button, the Arduino would automatically scan for available units in its I2C line, display the units in the serial monitor, and rename them as #0, #1, #2, etc.




Once the bus is initialized, you can use the serial input to send command and receive data from the unit of your choice. Currently available functions include:

- Get HEX address of unit:

  Type “X-a-” or “X-A-” and replace “X” with the unit number of your choice. In the example above, if I type “0-A-” and hit Enter, I would get “0x66” from the serial monitor output.

- Get data from designated unit:

  Type “X-g-” or “X-G-” and replace “X” with the unit number of your choice. The serial monitor will return the current position of the plunge (millimeters from the back end of the working range where all the electronics are located) and the pressure of the system in PSI.

- Move designated unit to position:

  Type “X-l-Y” or “X-L-Y”. Replace “X” with the unit number of your choice and replace “Y” with any number between 0 and 100. The unit will use its on-board, closed-loop control system to move the unit to the target position Y, which stands for millimeter from the back end of the working range.

- Move designated unit to pressure:

  Type “X-l-Y” or “X-L-Y”. Replace “X” with the unit number of your choice and replace “Y” with any number between 0 and 25. The unit will use its on-board, closed-loop control system to move the unit to the target pressure Y, which has the unit of PSI.

The serial input feature is handy in that you can use any device capable of serial communication (Raspberry Pi, Beagle Bone, Micropython Pyboard, Lego MINDSTORMS EV3, etc.) to control the Linear Syringe System in the same manner.

Keep in mind that using the buttons on the individual unit will break the current closed-loop operation. For now, this was made as a manual override feature, but I’m working on improving the code to make the switch of operation more convenient.

If all (or most of) the above features work on your system, great! But before you proceed to using the system at a day-to-day basis, I suggest that you go ahead and read through the “Final Check & Other Things to Note” section to make sure you understand the current state and limitations of this open-source project.

## Syringe Control System V3.1 Final Check & Other Things to Note

I2C protocol is capable of talking to more than 100 slaves at once. However, for the current version, it is recommended that you do not connect more than 8 units to the same master unit at the same time. This is due to the current rating of the wires used in the design. Future development will focus on how to tackle this issue.

Although the system uses Sparkfun's Qwiic system to communicaate with the units via I2C, it is currently NOT compatible with other boards on the Qwiic system: The SCS uses 5V on its logic line to power the Teensy LC, whereas the real Qwiic system uses 3.3V to power the sensors. Connecting other 3.3V sensors to the SCS system could permanently damage your sensors!

Keep in mind that although the system proves to be safe to operate, the stability of the system has not been so thoroughly tested. This mostly means that certain components might accidentally malfunction or get heated up.

That’s it for now. Thank you four your interest in my project, and I wish you best of luck in working with soft robotics and other things in your life. Good luck!