Jan 14th, 2020

Rough Plan:

- Step 1: LEGO actuator + small syringe,
  - powered by LEGO Technic block with simple forward/reverse option
  - manual air release with switch
  - goal: explore possibility of using low-cost, easily-accessible components to control soft robot actuators.
- Step 2: Stepper motor + medium/large syringe
  - Powered by Arduino/raspberry pi with ability to hold at exact position
  - Manual/automatic air release
  - Pressure sensor in line & display readings or become part of the control algorithm
  - Goal: to produce more stable and precise pressure output
- Step 3: Compare step 1 & 2 with existing control system (pump + solenoid valves)
- Step 4: using step 1 or step 2 to create a demo with multiple channels (snake, tentacle, articulated fingers etc.)
- Step 5: (if time allows) ultimate strength challenge
  - Fabricate a custom syringe at Bray (aluminum etc.)
  - Use a heavy-duty linear actuator with feedback
  - Test the performance on an artificial air muscle

To do: Look up ferromagnetic materials

Links:

LEGO Linear Actuator: https://www.amazon.com/Lego-Technics-Linear-Actuator-61927C01/dp/B014T9E43G/ref=sr_1_4?crid=8B4FMZXIQ38X&keywords=lego+linear+actuator&qid=1579061360&sprefix=lego+line%2Caps%2C144&sr=8-4

Stepper motors:

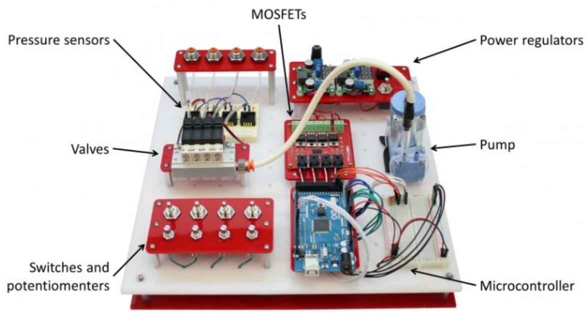
Heavy-duty Linear Act.: https://www.pololu.com/category/127/linear-actuators

Syringes:

Pressure Sensor:

Jan 22nd, 2020

- Performed research on existing control mechanism for soft actuators.
    - In most cases, hard valves and pumps are still widely used for controlling soft actuators:
        - https://softroboticstoolkit.com/book/control-board



This is a traditional control system presented by Soft Robotics Toolkit and is open-source. Similar set up is also used by NASA: https://www.nasa.gov/feature/langley/beyond-the-metal-investigating-soft-robots-at-nasa-langley

    - A research group recently published a paper about making a custom-made, soft, bistable valve to replace the hard solenoid valves: https://robotics.sciencemag.org/content/3/16/eaar7986/tab-pdf
    - Useful paper that summarizes current soft robotics manufacturing and control systems: https://www.nature.com/articles/nature14543.pdf
    - ***Found this useful research paper on using force-feedback and the pressure-feedback controllers to actuate the SRAs reliably. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6545483/pdf/nihms-1027456.pdf

Based on research, it seems a promising direction for my research is to build a syringe pneumatic/hydraulic system that uses force-feedback or pressure-feedback controllers to actuate soft robot actuators.

**Q Has anyone used the linear actuator with EV3?

- Submitted purchase request for some LEGO linear actuators
- Next Steps:
    - build the actuator in LEGO version
    - review the research paper and learn control
    - start designing version one of my syringe system and produce a BOM

Jan 26th, 2020

- Found this 3d-printed linear rail for syringes - https://www.youtube.com/watch?v=Q3A4NqTPOYY
  Might be useful as reference when I design my setup.

Jan 27th, 2020

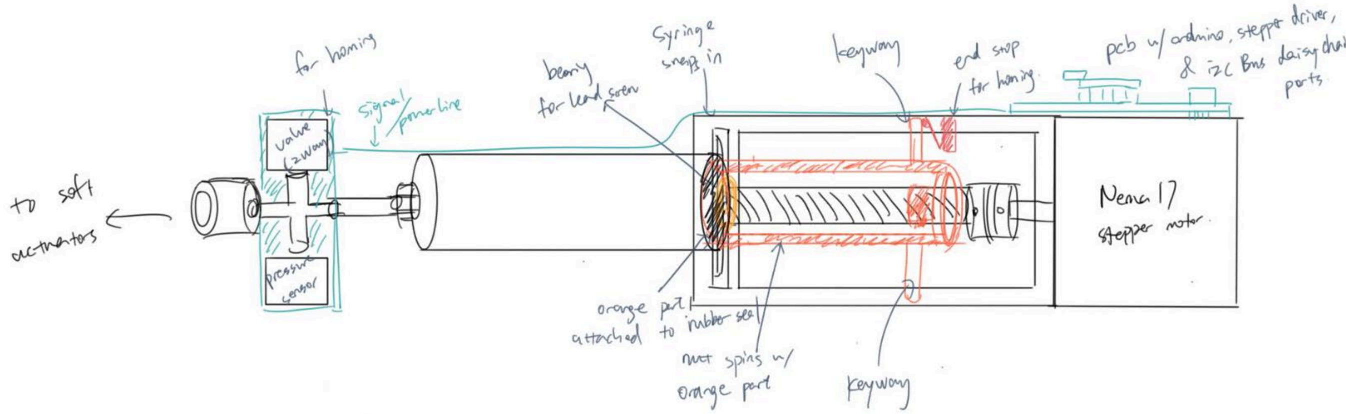
- Used a linear actuator, The MINDSTORMS EV3 kit, and a syringe to build a prototype setup for the LEGO version of the system. The linear actuator is directly mounted to a LEGO medium motor, which is then controlled by the brick with VSC and micro-python.
- The setup was able to control the linear motion of the syringe to some degree, however:
  - The structural integrity of the LEGO build and the torque provided by a single linear actuator is not ideal. The next step could be to mirror the build at the opposite side with another linear actuator to double the force and make the setup Stabler.
  - The LEGO linear actuator does not have a large stroke width. Will need to try with actuators to confirm this.

Jan 30th, 2020

- Found a research journal about developing a soft robot using ferromagnetic materials:
  https://www.researchgate.net/publication/335455581_Ferromagnetic_soft_continuum_robots
- Reason not to focus in this direction:
  - Not realistic with current available resources, time, and scope of the independent studies
  - Focuses more on developing new types of actuation methods, whereas I'm planning to focus on improving the control method for existing, accessible, and easy-to-fabricate materials.
- Finished building the LEGO version of the syringe system with two medium motors and two linear actuators.
  - Able to control the syringe to actuate some actuators in a reasonably accurate manner
  - Proved that LEGO can control soft robotics!
- Next steps:
  - Improve/Document the current LEGO setup
  - Start designing a non-LEGO version, give design&BOM

Feb 5th, 2020

- 80% done with the Instructables on making the LEGO syringe system. Still waiting to cast a successful soft actuator and film a demo of the complete system.
- Preliminary sketch done for the new, non-LEGO syringe linear actuator system:

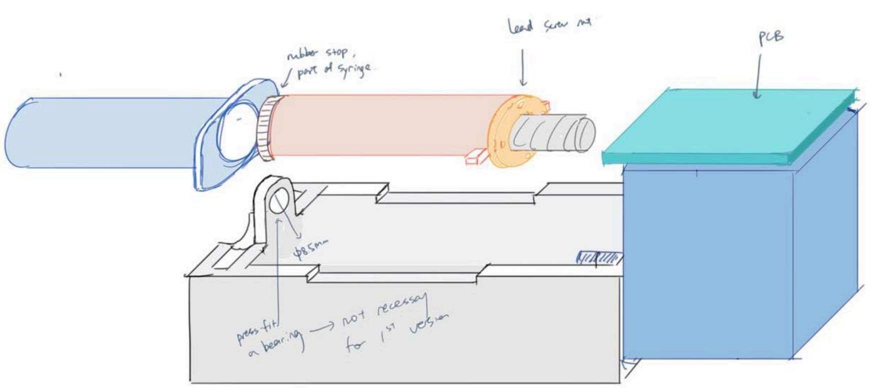


Cost per: $15 + $10 + $2 + $5 + $5 + $20 = $47
(pressure sensor, stepper, syringe, valve, 3D print cost, electronics + pcb)

  - Trying to limit the cost of each unit to be less than $50.
  - Ideally, each unit should have a microcontroller on board, capable of listening to an I2C bus – modularity.

Feb 9th, 2020

- Detailed Sketch



- BOM:

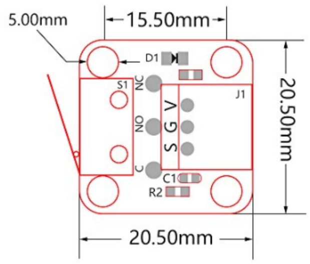
Item Name	Link	Qty	Price Per	Price Tota	Comment:
Stepper Motors	https://www.amazon.com/STE	1	45.99	45.99	
Lead Screw with Nut	https://www.amazon.com/Re	5	8.19	40.95	
Shaft Adapters	https://www.amazon.com/YO	1	7.99	7.99	
JST XH 2.54 connector 6 p	https://www.amazon.com/LA	1	8.99	7.99	

- Preliminary CAD with selected components:
    - Included stepper motor, shaft coupler, lead screw with nut, and syringe components
    - still need to add: end stop, tubing, upper shell, PCB, etc.

Feb 18th, 2020

- First Iteration of the linear actuator model has some issues:
  - It is difficult to put in the screw that are used to attach the stepper motor to the main housing
  - The syringe lip was a little loose when attached
- Updated the CAD model in SolidWorks, and hopefully I can print the next version soon.
  - I was also trying to incorporate end stops to the bottom_housing model. Found this compact end stop module made for 3D printers, but will suit my purpose:

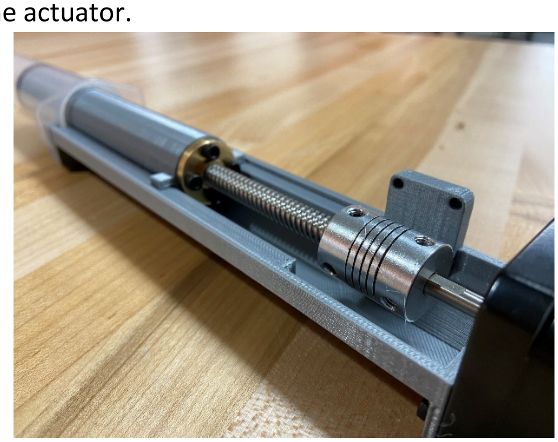
https://www.amazon.com/Printer-Accessories-Mechanical-Endstops-MXRS/dp/B083BZ3VTL/ref=sr_1_12?keywords=endstop+3d+printer&qid=1582095410&sr=8-12



This end stop converts touch sensor data to digital high, so I can directly plug it in a micro-controller, no need for pull-down resistor.

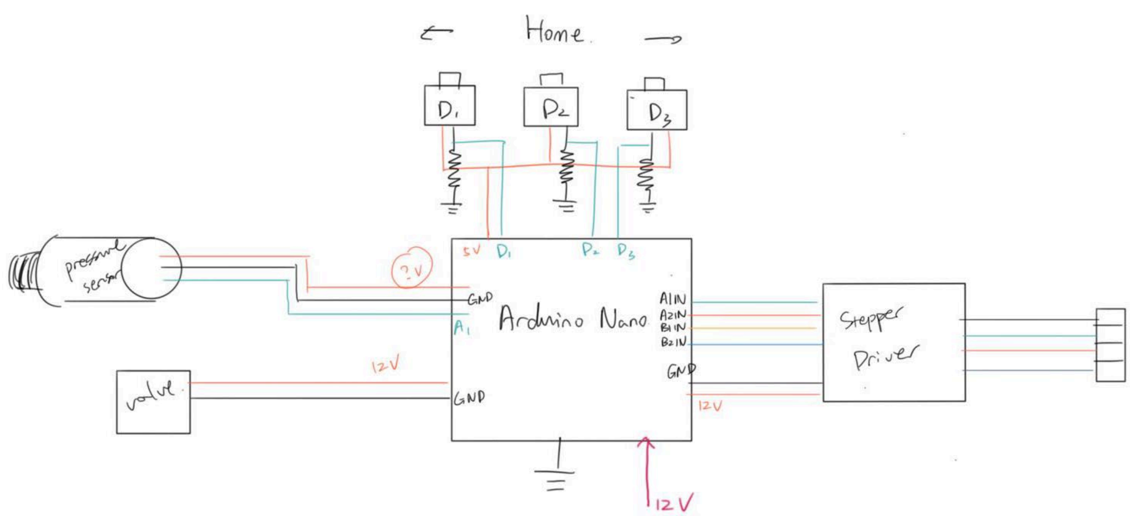
Feb 22nd, 2020

- 3D-printed out and assembled the first version of the syringe system. Assembled the components – except for the end stop and the pressure sensor, which I'm still waiting for to arrive in a few days. Right now, the system looks pretty good. Some following steps:
  - Connect the end stop module, pick a stepper motor driver, and breadboard the electronics for testing the syringe
  - The end support for the lead screw appears to be a little tight. Consider inserting a bearing or simply increase the diameter for the next iteration.
  - Design a 4-way connector that will be attached to the end of the syringe, which allows connection to a valve, a pressure sensor, and the actuator.



Feb 27th, 2020

- The pressure sensor, solenoid valve, and the end stops have arrived. Some issues that are currently preventing me from moving forward:
    - The end stop does not fit in the assembly the way id did in SolidWorks – the solder joints are thicker than my estimation. Will have to make changes in the design, moving the module to another location
    - The pressure sensor and the solenoid valve are a little bigger than I expected. Will see if I can still incorporate them while maintaining the modularity of each units.
- Now that I'm starting to think about controlling the motor, here's a qualitative schematic



March 3rd 2020

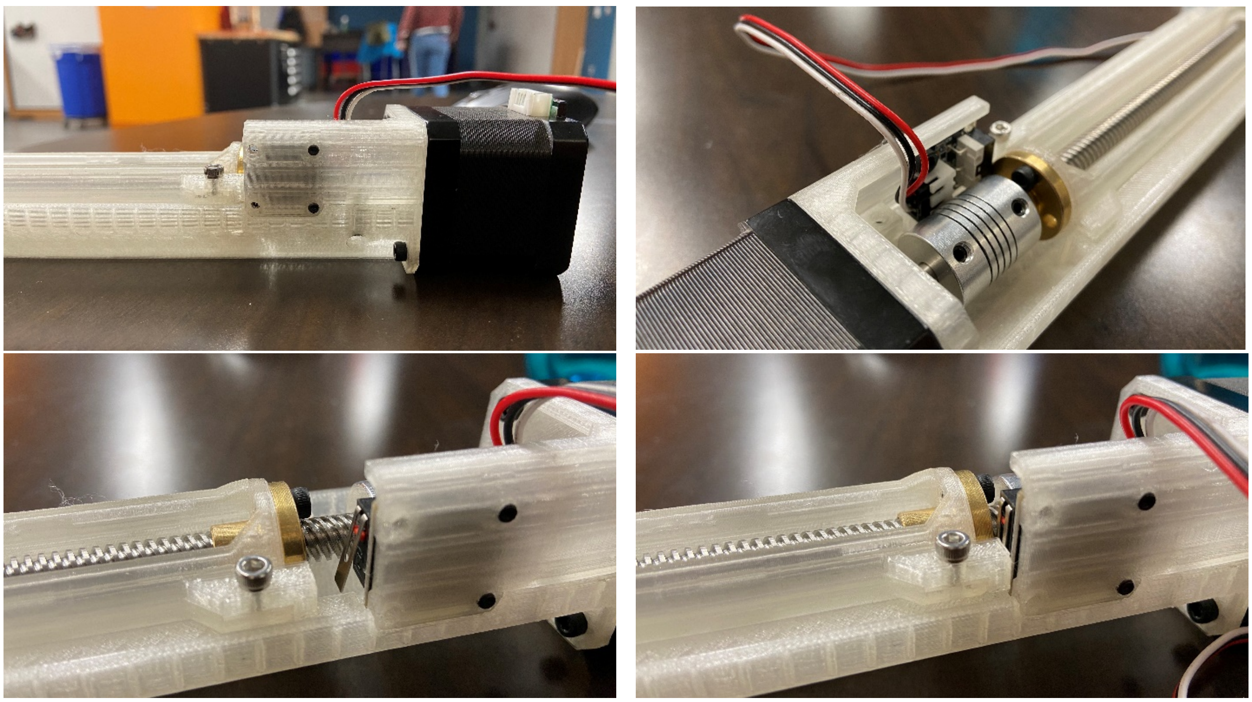
- Second version of base body: End stop fits in, but the flanged nut would hit the main switch body, deforming the touch switch before the switch can be activated. Solution: Modeled in an attachment piece onto the middle piston, so the piston will flush with the flange.
- The top attachment to the stepper motor forms an angle relative to the base-added another screw from bottom of the base to connect the two components together
- My plan for the following weeks:
    - Finish making a single model
    - Reach out to Steve Cogger (through Chris) after having the system tested on a breadboard.
        - Will need to decide between usb serial and I2C. Leaning towards I2C because the need to stack the modules together, and expand the possibility for advanced users to customize and integrate the system to their existing Arduino/hardware projects
    - Develop testing protocols to compare a single syringe module and a valve system
        - Compare the time response of both models
        - Noise level at performing specific tasks (e.g. incrementing pressure steps, holding at a specific pressure level, etc.)

  - Operation to failure of valves vs syringes (for valve – limited lifetime, for the syringe – likely to be much longer)
  - Maximum pressure output (might be more of a challenge to the syringe system
  - …

March 5th, 2020

- 3D-printed and assembled the next iteration of the end stop + base + lead screw. Below is a series of images showing the assembly design:

  Had to print an extra extension piece to make the center piston align with the end of the flanged lead screw nut, so that the end stop will fully engage at the current position.



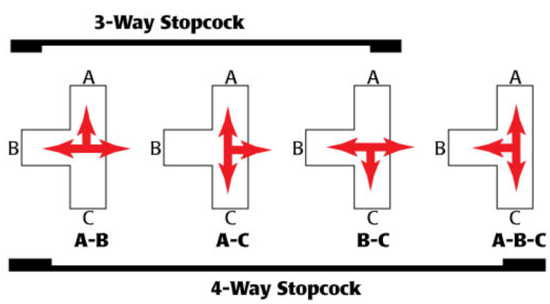
  Chris also suggested that I try a linear potentiometer. I might look into buying one to test it out, but there are a few concerns I have regarding the linear pot:
    - The size of the linear potentiometers is rather fixed – might be a little too big for the current design
    - Length – again, might end up too long/too short
    - One way to solve the last two issues is with a custom-made linear potentiometer, but it might end up not being as reliable and will take up longer time.

March 9th, 2020

- Found a different type of pressure sensor:
    - Link: https://www.digikey.com/product-detail/en/nxp-usa-inc/MPXV7002DP/MPXV7002DP-ND/1168436?utm_adgroup=Pressure%20Sensors%2C%20Transducers&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Sensors%2C%20Transducers_NEW&utm_term=&utm_content=Pressure%20Sensors%2C%20Transducers&gclid=CjwKCAiAzJLzBRAZEiwAmZb0au5x4Fh7pw8ptc1JYEKLhQb6Ahx8LDb0RJvFU8afhjauOeO0J3Nh_hoCql4QAvD_BwE
    - Advantage: compact and relatively cheap. Available on digikey as an SMD part. Might ask Chris to order one to play with and see if this can replace the big, bulky pressure sensor
- As for update regarding the solenoid valve and how it can be integrated:
    - Considering to instead have the 3d-printed piece have a hose nozzle output, so that users can decide whether they want a cheap manual valve or an integrated solenoid valve (will leave a place for the valve control in the pcb.)
    - Next is to perform iteration on 3d printers to find the optimal tolerance for luer lock and other related dimensions

March 10th, 2020

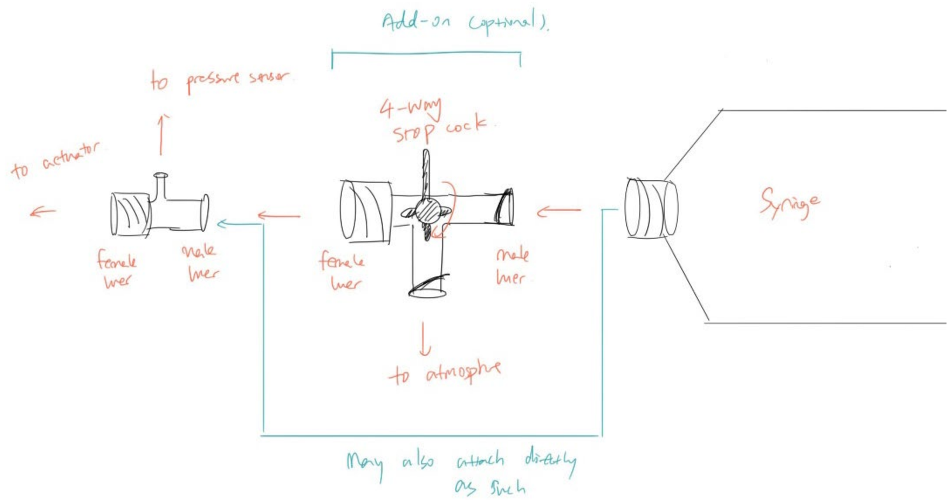
- Rapid prototyping on the Prusa MK3S 3D printer resulted in a working model of low-cost luer lock.
- Found a possible alternative to the solenoid valve system: 4-way stopcock:



Possible vendors:

https://www.vitalitymedical.com/hi-flo-four-way-stopcock.html?network=g&device=c&keyword=&campaign=1014818849&adgroup=pla-343185493395&gclid=Cj0KCQjw0pfzBRCOARIsANi0g0ujdrwSreyTPA7IAFGXs8l-pfPWNsNM-vq_oXd37A3u38b7v8JUY70aAmkpEALw_wcB
https://serfinitymedical.com/products/discofix-stopcock-4-way-b-braun-456020?variant=30738510577739&gclid=Cj0KCQjw0pfzBRCOARIsANi0g0tC74YHgleRaUm4IZAubfhLwfk1pENLHA0Ly4-TqFPpi-5cqYnZZ0EaAge8EALw_wcB

- As seen from the image, I could connect C to the syringe, and A to actuator + pressure sensor. Configuration #2 is the operation mode, and configuration #4 is the calibration mode.
- For the final setup, this could be a simple add-on to the system. The user can choose to directly connect the 3D-printed connector I designed that splits into the actuator and pressure sensor connection; alternatively, they could choose to purchase this and simply add the 4-way stopcock inline to make the calibration process easier to perform.



- Considering bringing back the Adafruit I2C pressure sensor – other options are wither two clumsy, too expensive, or does not have desirable pressure range.
    - Will validate this decision by testing the Adafruit pressure sensor and the NPT pressure sensor with the gripper actuator
- * Top priority is still finishing the open-loop system with the stepper motor and the syringe.

March 11th, 2020

- Campus will shutdown doe to Coronovirus. Will need to come up with a plan

Week of March 9th	Finalize the physical design the best way I can Try and solve the stepper motor vibration issue Breadboard manual control system with buttons
Week of March 16th	Take some rest…? Use the manual control system to test out different pressure sensors. Pick a pressure sensor.
Week of March 23rd	(if needed, figure out soft I2C/ possibility to use an Arduino as both a master and a slave)
Week of March 30th	Close-loop system with pressure sensor in line with syringe and actuator
Week of April 6th	Continue working on single close-loop system Start developing I2C code for master Arduino/Pi and individual Arduinos.
Week of April 13th	Continue developing I2C code for master Arduino/Pi and individual Arduinos.
Week of April 20th	Acquire performance data and compare with air pump.
Week of April 27th	Acquire performance data and compare with air pump. Organize research results and prepare for potential presentations.
Week of May 4th	

Will mostly need (and might be able to get):

A soldering station

Jumper wires, resistors, capacitors, other electronics. Various pre-made boards from Adafruit/Pololu etc.

PCB manufacturing service

Nscope

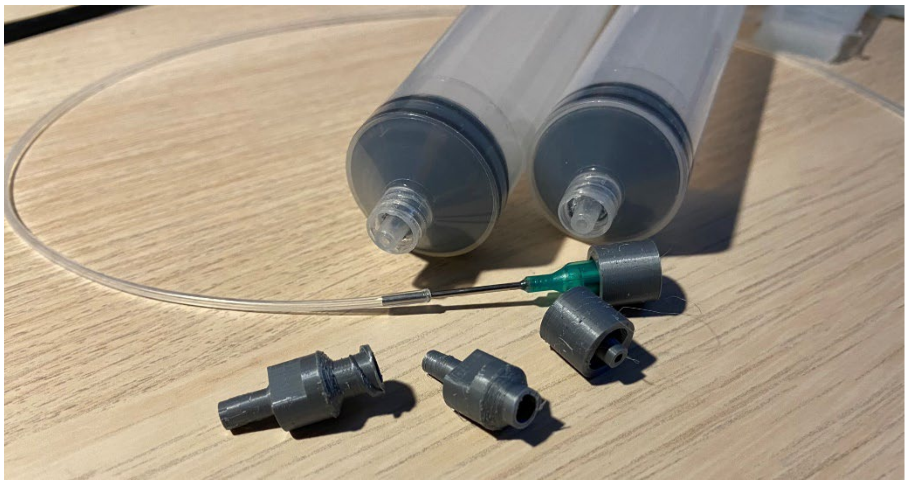
March 21st, 2020

- A little behind the schedule, mostly dealing with COVID-19 related issues. Hope to resume research starting today.
- Revisited previous breadboard setup where the stepper motor is vibrating and creates unpleasant sound.
    - Tried adding different capacitors to reduce current spikes
    - Searched online to try to find solutions
    - Reduced some vibration noises by applying hot glue to loosen parts on the end stop module
    - Considering replacing the flexible shaft coupling to spider-style/normal coupling to reduce noise and size. Some viable options include:
        - https://www.amazon.com/ZYCST-Aluminium-Coupling-Connector-Airplane/dp/B07QPKGM1Q/ref=sr_1_48?crid=10OPTD1S6MFZ7&keywords=5mm+to+8mm+coupler&qid=1584807883&sprefix=5mm+to+8mm+%2Ctools%2C150&sr=8-48
        - https://www.amazon.com/Befenybay-Aluminium-Flexible-Coupling-Connector/dp/B07RMZCLZ3/ref=sr_1_4?crid=10OPTD1S6MFZ7&keywords=5mm%2Bto%2B8mm%2Bcoupler&qid=1584807883&sprefix=5mm%2Bto%2B8mm%2B%2Ctools%2C150&sr=8-4&th=1
- The stepper motor, unfortunately, was not delivering enough torque. It started skipping once out of a few steps as pressure increases. It still pushed the syringe all the way to the end, but skipping steps made it hard to monitor the position. Possible solutions I can think of:
    - Ignore the step-skipping, and use a linear potentiometer to monitor the position
    - Use a stronger stepper motor
    - Use a DC motor with encoder (or without?) and use either two endstops or a linear potentiometer
    - Use a brushless motor…? Maybe see if O-drive has an open-source schematic…
- Motor RPM to linear actuator moving speed:
    - Lead screw: 2mm pitch, 4 starts
    - 1 revolution of motor – 8mm travel of lead screw.
    - Travel speed (mm/sec) = (RPM / 60) * 8mm
    - Current speed with 17HS19-2004S1: approx.. 16mm/s; therefore, to find a DC motor that matches the speed, I should be looking for a motor with RPM > 120. Ideally the higher the better, with enough torque and reasonable rated voltage and current draw.
- Next steps:
    - Prototype a connector with female luer, male luer, and a tube coming out – then hook up a pressure sensor in line with the syringe, and use a scale found in my house to graph a pressure-torque characteristic of the motor.
    - Pick a desirable maximum pressure and a corresponding force. Calculate the torque needed for the motor. Use this as the final criteria in selecting a motor (maybe with a factor of safety of about 1.5?).
        - https://www.pololu.com/category/115/25d-metal-gearmotors
        - https://www.pololu.com/category/116/37d-metal-gearmotors
    - Look for linear potentiometers (100mm travel length).
        - https://www.digikey.com/product-detail/en/tt-electronics-bi/PS100-2B1AR10K/987-1407-ND/2620676

        - Might need a larger syringe since the travel length is now limited: https://www.amazon.com/DEPEPE-Connections-Scientific-Measuring-Filtration/dp/B07PPFLKRD/ref=sr_1_5?keywords=100ml+syringe&qid=1584851175&sr=8-5

March 22nd, 2020

- Prototyping of the Y Luer connector hit a bottleneck – the pla isn't strong enough and does not produce an air-tight environment. The pisture shows the iterations I performed today:

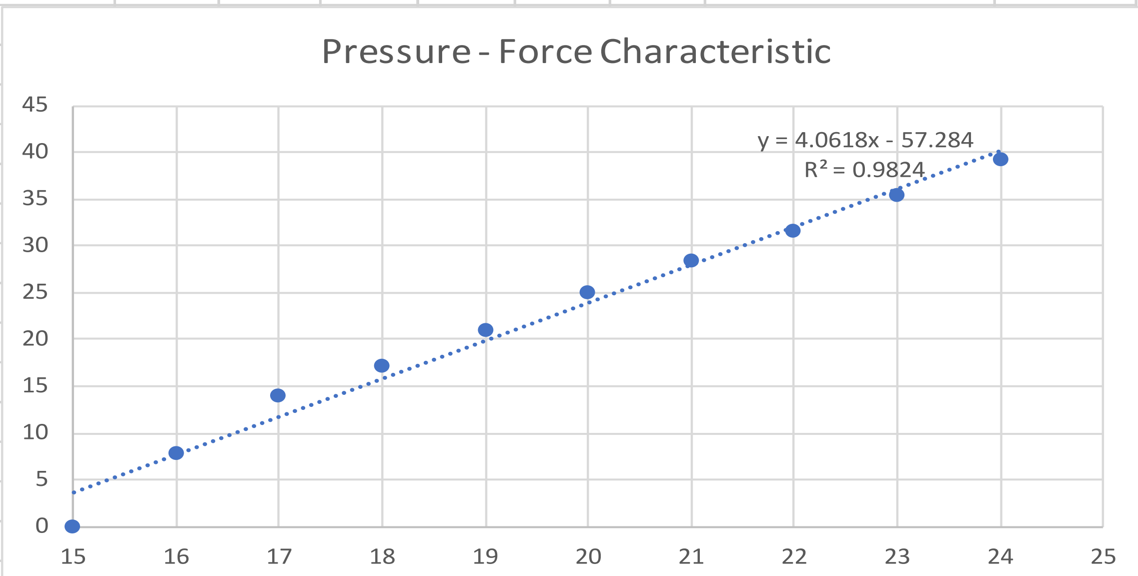
  The PLA wasn't strong enough – as I tried to untwist the connector off from the syringe, the end broke off and stayed inside the luer connector, making the syringe unusable.
  Some options includes:
    - Teflon Tape
    - Use petg/or other PLA substitute
    - Use industrial Luer connectors instead (promising)

  So the current plan would be:
    - Purchase some y-connector, and while waiting for the shipment, I could also purchase some PETG and Teflon Tape. In the best scenario, both methods would work and I can present more alternatives of my design in the final report. I hope that through this process, at least one of the options would work.
    - https://www.qosina.com/y-connector-male-luer-connector-with-spin-lock-two-female-luer-lock-ports-84044
      https://www.mcmaster.com/51525k122
      use 3/32 ID Tubing in the pressure line. Solved the problem!
- To do:
    - Start compiling an order list for CEEO
    - Use the scale to roughly measure the pressure-force relation
    - Use the relation to predict the force needed to actuate the example gripper design
    - Calculate the required force/torque of motor. Pick a new motor and re-design.

March 23rd, 2020

- 60ml Syringe Pressure-Force Characteristic Test. Method: I placed the end of the syringe on a kitchen scale, with the pressure sensor hooked up to the other side. Next to the setup, I have the Arduino Serial monitor opened, which outputs the pressure in the system in PSI.
  YouTube Link (listed only): https://youtu.be/_aCKt8hK5S8

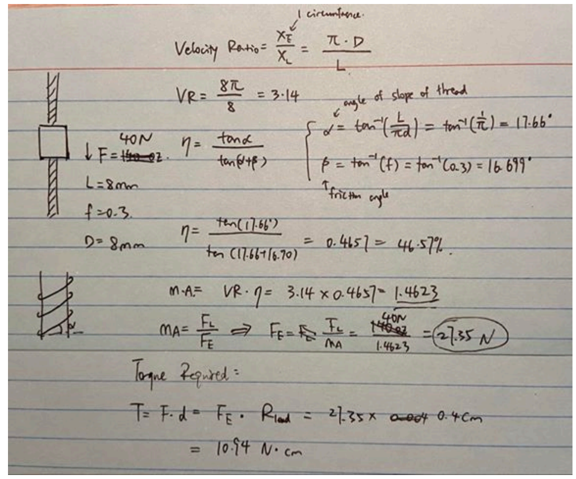
Pressure (psi)	m1 (g)	m2	m3	m4	m5	m6	Force (avg from 4-6)	Gripper (A
15	0	0	0	0	0	0	0	19.77369
16	2340	2031	2246	553	967	867	7.80549	
17	2789	2435	2405	1541	1380	1372	14.03811	
18	2861	2863	2774	1719	1702	1815	17.12172	
19	3120	3302	3359	2068	2262	2057	20.88549	
20	3320	3624	3627	2455	2609	2590	25.02858	
21	3653	3871	4233	2780	2909	2973	28.32474	
22	3887	4062	4457	3042	3298	3293	31.49991	
23	4253	4357	4550	3522	3655	3643	35.3814	
24	4702	4610	4706	3999	3971	3994	39.12228	



  - I chose to only graph results averaged from trial 4-6, since the frictionless syringe gives a much more linear result. The force required to get the pressure to 24 PSI (a little below the limit of the Adafruit pressure sensor) is around 39.12N.
  - I also did a test on the existing soft gripper I have by pressing the syringe to the kitchen scale in similar fashion. I performed 3 trials and ended up with a force output of around 19.77 N. From the characteristic graph, the pressure around 18.97 psi, which is well within the range of the pressure sensor.

March 25th, 2020

- Calculated the required torque based on the following video: https://youtu.be/cdDYhViYO9s

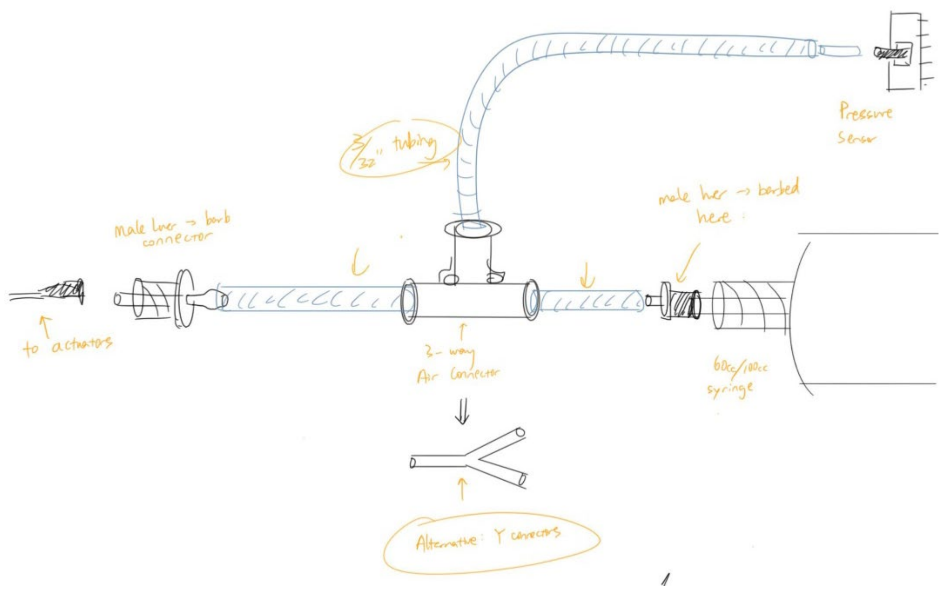
  Interestingly, the calculation implies that only about 10.94 N*cm of Torque is needed, which is well within the “holding torque” of the stepper motor I am using. Therefore, I revisited the stepper motor setup.
  - I noticed that the friction inside the free-running syringe is partially the reason of the poor performance. I applied some cooking spray :) to the inner rubber seal and it worked a little better, but still not as close as the rated performance.
  - I then took a closer look at the driver and whether the driver is actually outputting enough current for the motor. Measuring the current through one coil gives approximately 600mA. From the A4988 Datasheet, it appears that this corresponds to a current draw of ~1A, which is less than the rated 2A of the stepper.
    A4988 Datasheet:
    https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf
    Pololu stepper driver instruction:
    https://www.youtube.com/watch?v=89BHS9hfSUk&feature=youtu.be
  - I then realized that the A4988 board I am using is different from the ones that Pololu sells – could be a knockoff, which prevents me from accurately measuring and adjusting the current limit. I already turned the on-board potentiometer all the way to the max side, so I think I’ve pushed board to its limit. I may need a better stepper driver.
  - Next, I plan to test the torque of the motor by 3D-printing a lever-arm and attach weight to the arm.
- While I am waiting for the result of the torque testing, I am debating whether I should:
  - keep using the stepper motor and hope that the new driver would solve the problem. The advantage is that most of my CAD model would remain the same, and I already have the schematic, testing code etc. set up; the downside is that this has a higher chance of not working at all in the end.
  - Switch to a DC motor based on the calculated torque and a relatively high “factor of safety”. This would take more time to implement but would have a higher chance of success; besides, DC motors are much easier to deal with than stepper motors. I will have the linear pot and pressure feedback, so precise “stepping” is no longer a necessary element to have.
  - In both cases, I need to start integrating the linear potentiometer, replace the flexible coupling and the endstop, and re-print the entire setup.

March 26th, 2020

- Some further investigation regarding the lack of torque produced by the stepper motor.
    - 3D-printed a lever arm and used a water bottle and a kitchen scale to create “standard weights” for the torque test.
    - After a series of testing, I found out that under low RPM (about 30 RPM), the motor was able to produce about 10N*cm of force, which is, coincidentally, the same as the maximum torque required to reach the pressure sensor’s upper limit. The reason I was running into skipping issue before were: a) Speed was too fast, and b) I had a bad control code for the motion (I had increments of 20 steps inside two if statements which listened to the states of two buttons)
    - However, even if The system could work right now, I’m still leaning towards the plan to switch to DC motor. Reason being that:
        - The stepper motor system is far too loud, and it vibrates like crazy…
        - The stepper motor would run at a lower operation speed at desired output torque.
        - DC motors are far easier to control, and with the linear potentiometer feedback, the precision won’t be a problem.
- New Purchase list:

Item Name	Link	Price Per	Qty	Total Price	
12V motor MP #3228	https://www.pololu.com/	19.95	1		
12V motor LP #3251	https://www.pololu.com/	19.95	1		
12V motor MP #3227	https://www.pololu.com/	19.95	4		
12V DC Gearmotor	https://www.robotshop.c	14.95	4		
4mm to 8mm shaft coupli	https://www.amazon.con	4.09	4		
Microcontroller w/ 2 I2C	https://www.digikey.com	14.38	4		
Motor Driver DRV8871	https://www.digikey.com	7.5	4		
Linear Potentiometer	https://www.digikey.com	5.49	4		
PLA filament	https://www.mcmaster.c	45	1		
Syringe	https://www.mcmaster.c	3.13	4		
Male Luer to 3/32"	https://www.mcmaster.c	3.78	1		
Female Luer to 3/32"	https://www.mcmaster.c	4.27	1		
3/32" Y connector	https://www.mcmaster.c	7.96	1		
3/32" Tubing	https://www.mcmaster.c	15	1		
PTFE Sealing Tape	https://www.mcmaster.c	2.58	1		

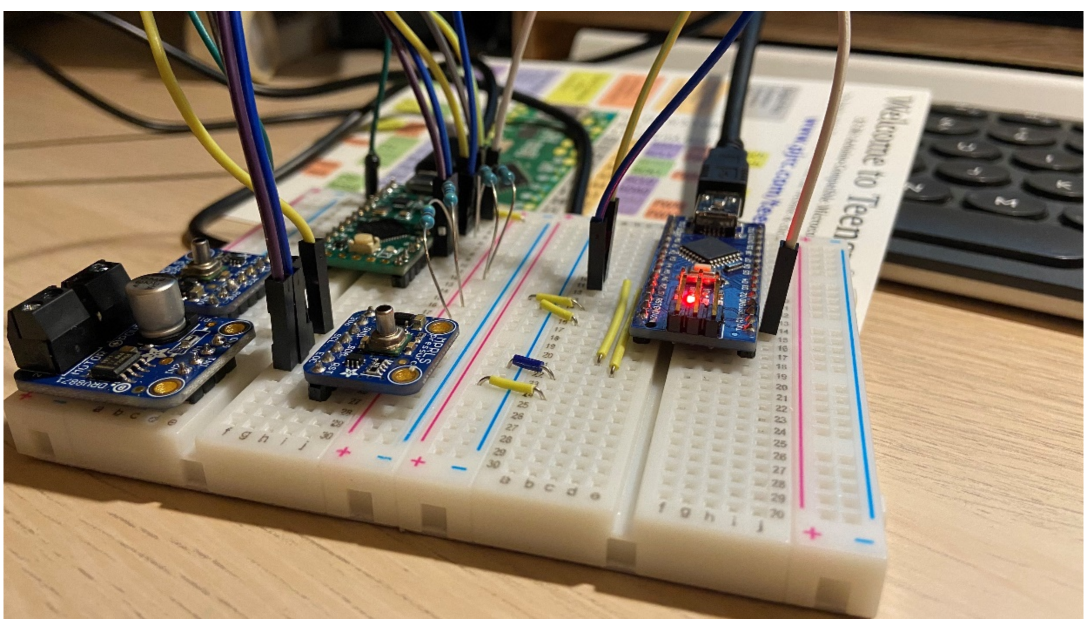
  Concern: Due to the pandemic most of the online sites have announced to have delayed shipping or no shipping at al. I have found suppliers for most of the components I need, except for the 4mm to 8mm couplings, which I could only find on Amazon.
- New design pared with the updated bill of materials (syringe output re-design):



March 30th, 2020

- McMaster-Carr & Digi-key orders have arrived. With most of the components at hand, I organized the components, soldered some boards, and started hooking up wires for the new electromechanical system.
    - Useful links:
        - DRV8871: https://learn.adafruit.com/adafruit-drv8871-brushed-dc-motor-driver-breakout/usage
        - Linear Pot: https://www.ttelectronics.com/TTElectronics/media/ProductFiles/Potentiometers/Datasheets/PS100.pdf
        - Pressure Sensor: https://www.digikey.com/product-detail/en/adafruit-industries-llc/3965/1528-2713-ND/9658071?utm_adgroup=Evaluation%20Boards%20-%20Sensors&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Development%20Boards%2C%20Kits%2C%20Programmers&utm_term=&utm_content=Evaluation%20Boards%20-%20Sensors&gclid=Cj0KCQjwsYb0BRCOARIsAHbLPhEKd4PRP7A6Z04qtc4TvoXcSPQ_mGHCGAPBQzXg4TaqRHrryTsytdYaArnTEALw_wcB
          https://learn.adafruit.com/adafruit-mprls-ported-pressure-sensor-breakout
        - Teensy-LC: https://www.pjrc.com/teensy/teensyLC.html
          Teensy i2c wire library: https://github.com/nox771/i2c_t3
- Outcome at the end of the day:
    - I found a modified wire library specifically designed for Teensy boards, including options to declare multiple i2c communication lines at the same time: https://github.com/nox771/i2c_t3
      One difficulty I encountered is that the library for Adafruit MPRLS pressure sensor was written with the original Arduino "Wire.h" library. At the end of the day I managed to get the Teensy wire library working with the sensor by incorporating part of the code from the original header files from the Adafruit library.

- There still exist one minor issue: Even though I tried to duplicate the original library to my best ability, the values read from the new Teensy code appears to be different from the values red from the original example code (e.g. at atm the former would read at around 6591085 while the later reads 139084.48). The values do change noticeably as pressure varies, indicating that it is the real data. I'm planning to bring this issue up in the Slack channel to see if other people have any insights.
- With one i2c line established, I then hooked up an Arduino Nano to the other i2c channel as the "ultimate master". With the i2c scanner I was able to detect the Teensy board at the address I specified, indicating a success in using i2c in multiple channels at the same time. I will test the system in more depth tomorrow.

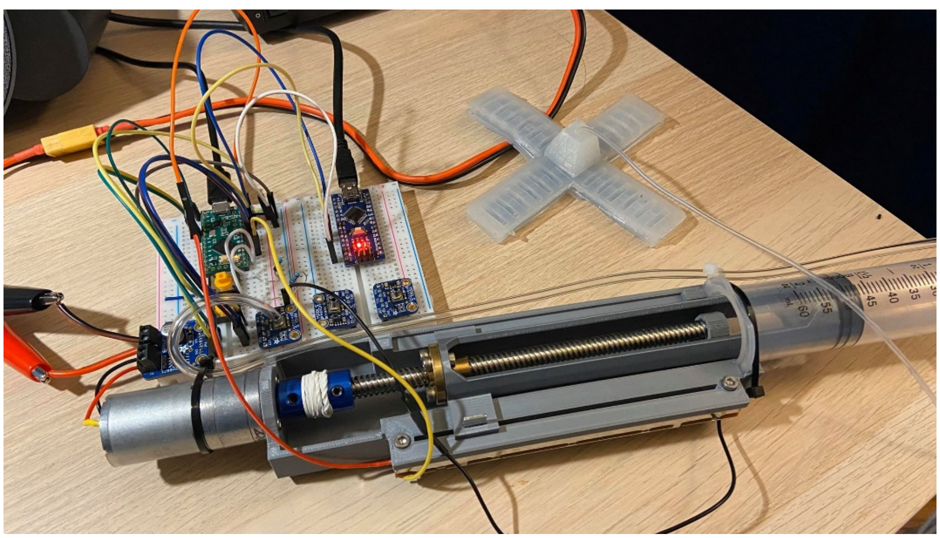


March 31st, 2020

- Out of serendipity (also kind hints from Chris) I solved the MPRLS reading issue by studying the datasheet of the Honeywell pressure sensor. The issue was that out of the 4 bytes of data that the sensor sent to the master, the first byte is always a status indicator. So I did an extra “Wire.read()” which would read and clear the first byte of data from the RX buffer. The last 3 bytes (24 bit) now contains the raw data which could be well-calibrated using the existing equations in the documentation (kindly provided in the Adafruit example library).
- Performed a pilot test with the y connectors, luer connectors, and the new tubing. Used the syringe to actuator the silicone gripper while taking pressure readings in the system. Video here: https://youtu.be/JfOo434jeLU
- Finalized test code for “big master” (now an Arduino Nano) and “mid_master” (now a Teensy LC). Updated Github repo.

April 1st, 2020

- Finished a prototyping system with a Teensy LC as the “mid master” and the Nano as the “top master”. The Teensy has 2 independent I2C line operating at the same time, acquiring data from the pressure sensor and sending information to the Nano. Ideally, there would be as many “master” as the I2C line allows (up to 119 devices?).
  Video here: https://youtu.be/PBJPoOoM7Gk



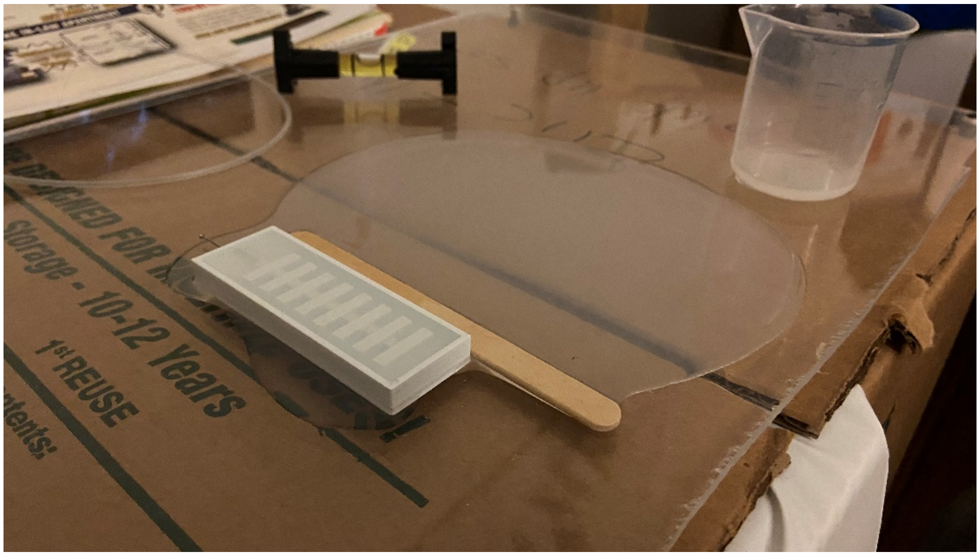
- Issues:
  - Linear Potentiometer could not be red by the EV3. I tested my code by using a 10k rotational potentiometer and it worked just fine (from 1 to 1023). I checked all 5 of the linear pot I purchased from Digi-key with a voltmeter. One thing I noticed is that they are not perfectly linear – I slid it to the middle but reads around 8k…
  - Friction inside the syringe – For the old ones I have I tried vegetable oil and slide bearing grease – it seems like one of them made the rubber piston to swell, decay, crack, and the syringe unusable… Looking for suggestions (it seemed that silicone gel works pretty well?).

April 2nd, 2020

- Fixed the potentiometer issue by switching from 5V line to 3V3 line.

April 6th, 2020

- Started incorporating the potentiometer to my manual code
  - Added hard stops at both ends of the linear rod. This is just a temporary add-in, and I will later investigate how to smoothen the movement near the rod – right now the motor would go back-and-forth quickly instead of “softly holding” at the end position.
- As of now, I’m planning to add the following features to the system (open to any suggestions):
  - Top_master:
    - **Initialize:** Detect the units connected to the I2C channel on bootup.
    - **Move unit to desired pressure:** Return true if succeeded and false if reached the end of the rod. Ideal for advanced movements that involves sequencing and pressure feedback.
    - **Move unit to desired position:** Move the selected syringe in the system to the desired position. This is useful for pilot testing if combined with pressure data.
    - **Ask for data:** Grab pressure and potentiometer(position) data from selected unit.
    - **Fail-safe Features** (Default Active): Stop relative action if received error message from a unit.
  - Mid_master:
    - **Obtain Pressure:** Constantly grabs data from the MPRLS pressure sensor. Stores the current pressure data and wait for Top_master to request for it.
    - **Analog Read to Position:** Grabs current analog reading from the potentiometer and converts to the position of the rod using a calibration curve.
    - **Manual Override:** a) Has two buttons for manual forward/backward action, or b) has one button that, when pressed and held, will maximize the compliancy of the system and allows user to manually move the syringe position. This feature should have top priority above all other commands, but as soon as the buttons were released the system will return to the previous command sent from its Top_master (hold at pressure, move to position, etc.)
    - **Fail-safe Features** (Default Active): Stop after reaching the ends; stop if reaching pressure limit. The selected unit will try to hold the last viable state and send its Top_master an error message.
  - E-stop for Motor
- 3D-printed a small bending actuator mold and started casting it. It takes less air to inflate a single bending actuator than a gripper, making it more suitable for consistent pressure/position controller test.



- Final Version, Bill of Material

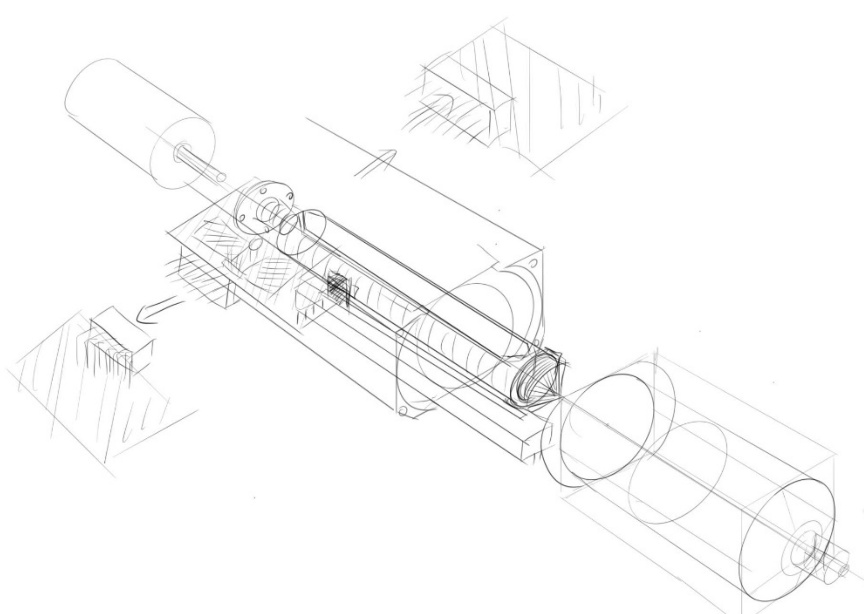
Item Name	Link	Qty	Price Per	Total Price			
Mechanical							
M3 Flat Head 5r	https://www.mcmaster.com/92125a125						
Pneumatic							
PCB/Electronic							
Mini Push Butto	https://www.pololu.com/product/1400						
Straight M-Hea	https://www.pololu.com/product/965						
Right Angle M-I	https://www.pololu.com/product/967						
Qwiic I2C Break	https://www.sparkfun.com/products/14495						
Qwiic to M-Jum	https://www.sparkfun.com/products/14425						
Qwiic 50cm Cab	https://www.sparkfun.com/products/14426						

April 8th, 2020

- Started sketching & CADing & designing the final, stackable version of the syringe system
    - New find: All current Teensy boards have internal pull-up resistors, so I don't need to wire in an external resistor at all for push buttons!
      Just do "pinMode(pin,INPUT_PULLDOWN)".
- Chris suggests using a more standard (RJ11, Cat 5, etc.) for data transmission, and separate power.
- After some research, I decided to go with the Qwiic system from Sparkfun:
  https://www.sparkfun.com/qwiic#products
  Sparkfun makes a series of board using short, 4-wire JST-like cables for I2C communication - perfect for what I need. In the future I'm also considering making my board "Qwiic-compatible" by following their rules and putting a Sparkfun Qwiic logo on there etc.
    - I was particularly looking at the Qwiic-to-breadboard breakout and the 50mm short cables that they sell:
      https://www.sparkfun.com/products/14495
      https://www.sparkfun.com/products/14426

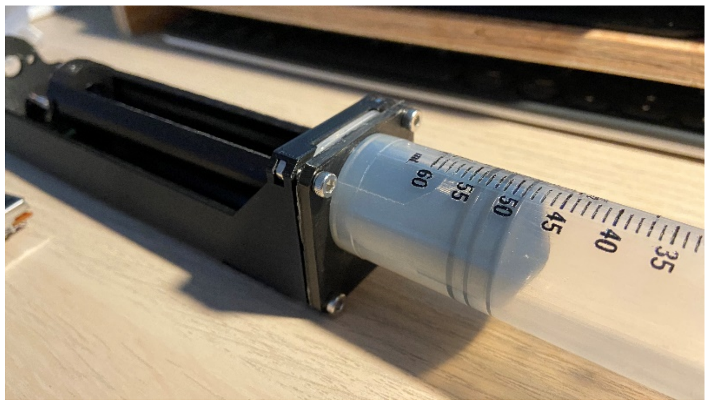
April 10th, 2020

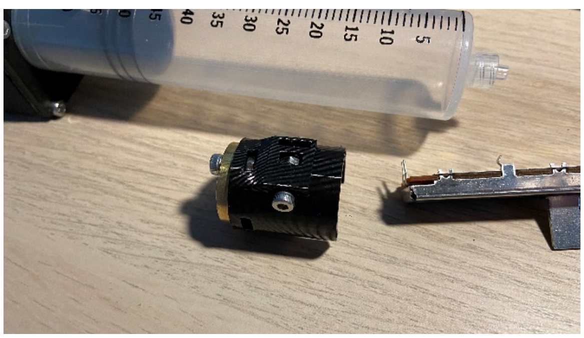
- After a lot of preparation (brainstorming, research, frequent visits to Digikey/Sparkfun/McMaster-Carr), I finally started designing & prototyping for the final stackable version of the syringe control system.
- I began with some concept sketches (they don't usually contain all the details I have in mind due to my limitation in drawing skills, but they are most helpful in helping me organize my thoughts.)

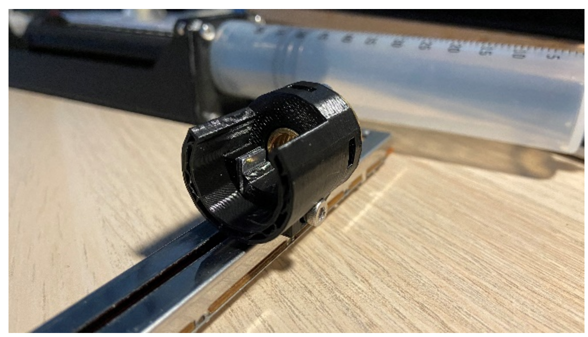


As shown by the image above, I plan to mount the linear potentiometer under the lead screw mechanism in order to save space and keep the entire setup within the size of NEMA-17 (42mm by 42mm). I'm also planning to try to fit all the electronics within the back of the base (under the shaft coupling). That way the PCB can have holes for the linear potentiometer to solder into, as well as screw terminals and Qwiic ports facing the back of the base, which will not get in the way of multiple units stacking against each other.

- Some pictures during the prototyping process.



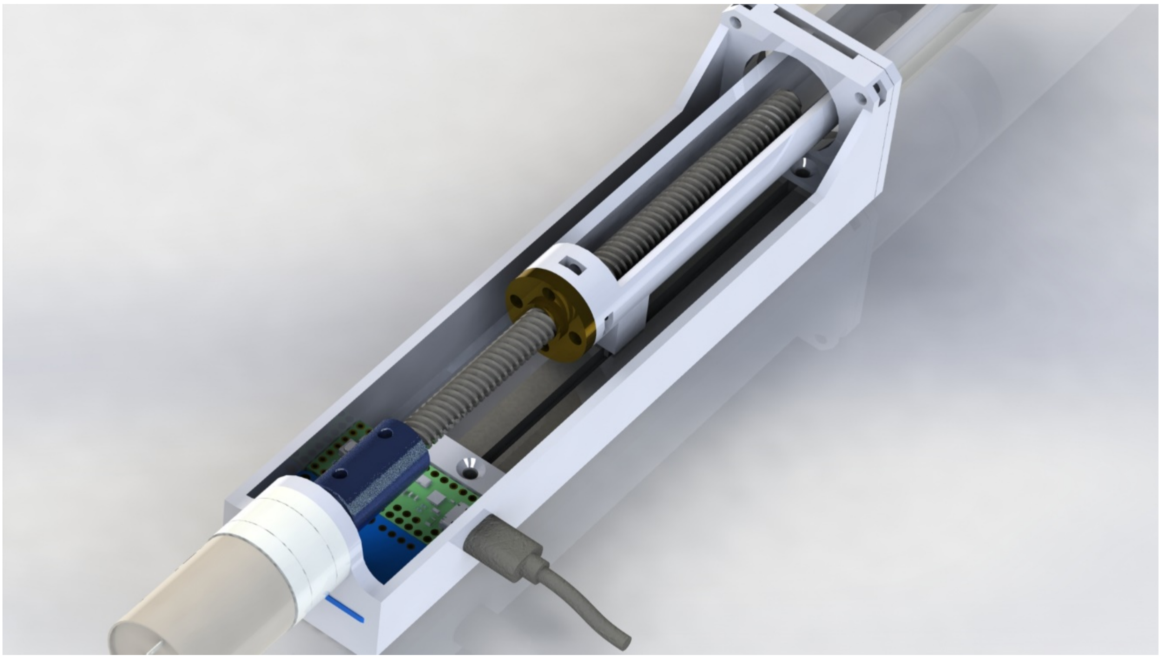




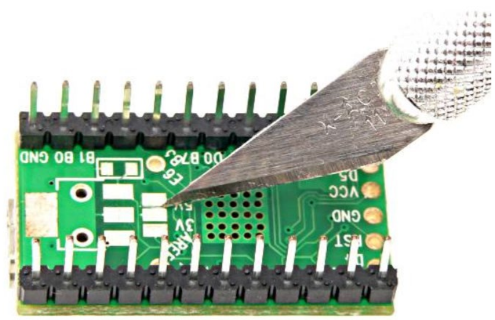
  - I changed the way the syringe is mounted from top-entry to front-entry with a clamp to increase stability
  - I used a nut and a bolt as a "set screw", which grabs onto the handle on the linear potentiometer. This way I'm hoping to use the potentiometer as a slide for the lead screw mechanism.

April 11th, 2020

- Rendered Design:



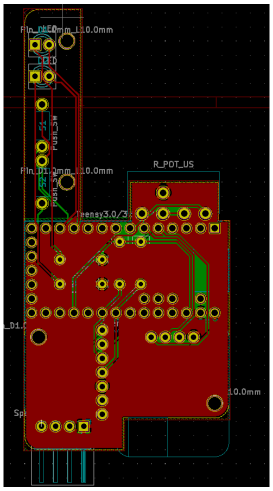
- Started the PCB design in KiCAD.
- While designing the PCB, I found that the recommended VIN for Teensy LC is 3.7 – 5.5V, while the Qwiic cable conventional voltage is 3.3V. The Teensy could theoretically be powered with 3.3V by using a voltage regulator.
    - When using external power, the trace to VUSB must be cut off!

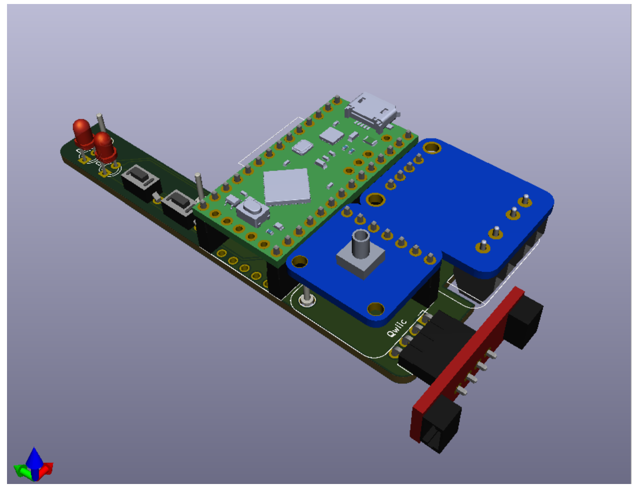


    - A diode can help prevent plugging in the battery backwards: https://www.digikey.com/product-detail/en/diodes-incorporated/1N5817-T/1N5817DICT-ND/190530
      (although if using Qwiic, chances of plugging the cable in backward is quite low)

April 13th, 2020

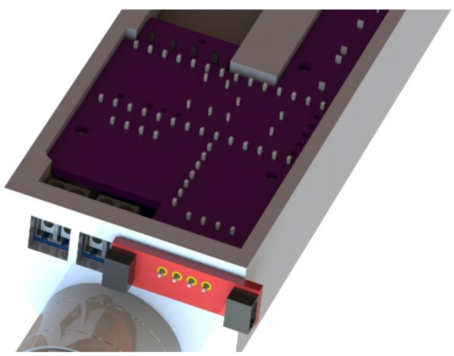
- After a lot of back-and-forth between Solidworks and KiCAD, I have my PCB version 1 ready to be sent for manufacturing:

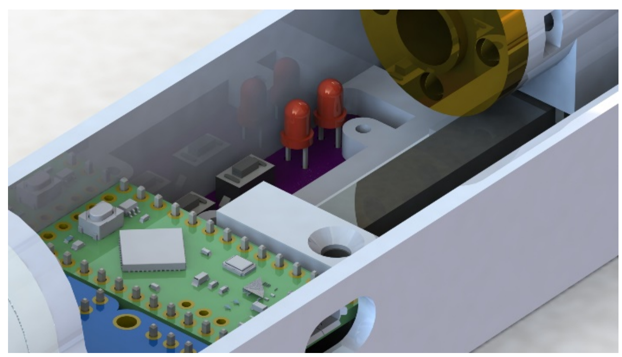


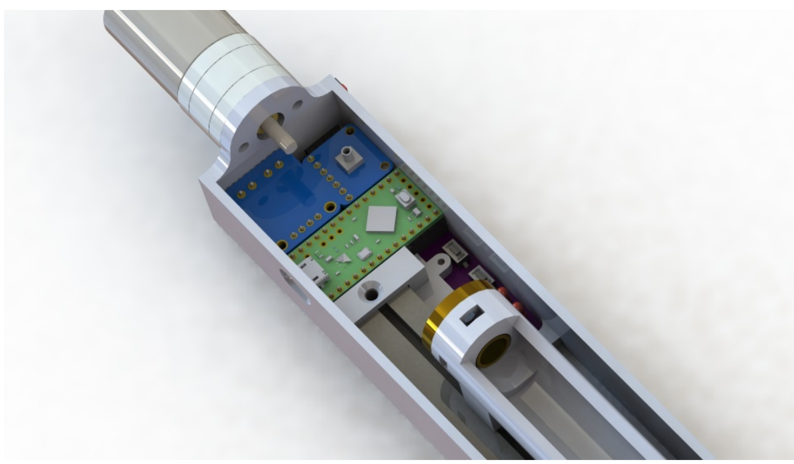


I made use of the 3D visualization tool inside KiCAD to generate a STEP file which I can directly import into my Solidworks Assembly. To do that, I had to make the STEP files for my custom footprints and upload them to KiCAD individually first; then all the electronics will be mounted on the bare PCB board as I export the final 3D model. This function significantly helped speeding up my design process.

- I also placed some orders (hopefully the last patch before semester presentation) from McMaster-Carr, Pololu, Sparkfun, and Digikey.
- By the end of the day I finished the preliminary attachment of the PCB assembly inside the main CAD assembly. I modified the 3D-printed parts by cutting out slots, holes, etc. in order to fit the PCB from below the entire setup (and learned how to swap components within the assembly).
  Some rendering of the most up-to-date Model:





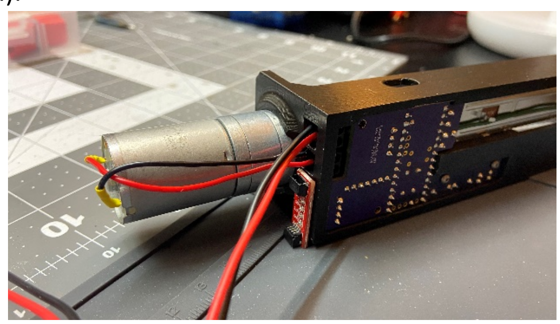
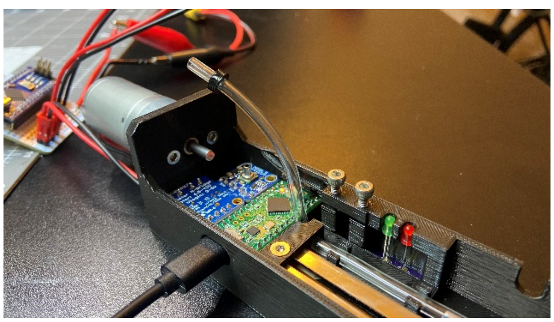
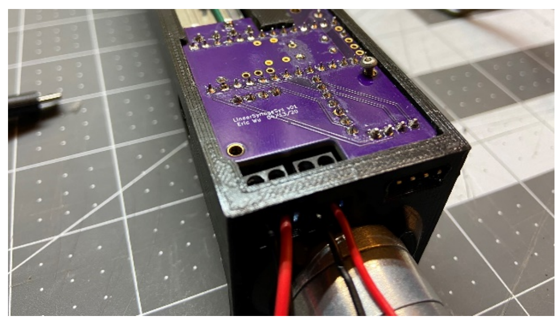


April 21st, 2020

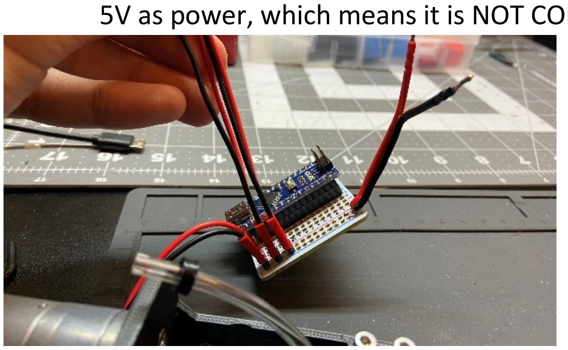
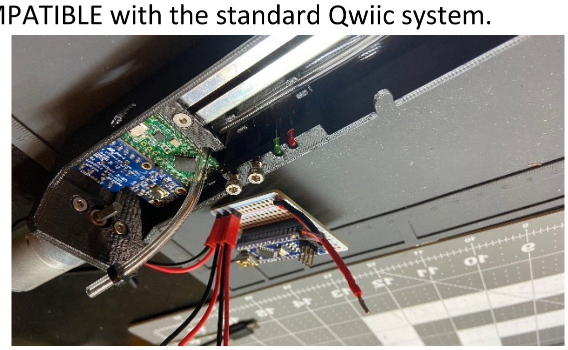
- OSH Park order arrived. I did some quick sanding to get rid of the tabs and made sure that the board fits within the 3d-printed base.
- Revisited the main CAD model and made some minor adjustments – button clearance, cut-out placements, hole alignments, etc.

April 22nd, 2020

- I finished assembling one complete Version 3 unit and ran some tests using the previous manual code (added some simple LED feedback for proof of concept). The system worked pretty well! I filmed some video clips just in case nothing works in the end :).

  - Bad news of the day: I did accidentally fry a Teensy LC…Still don't know why yet. Could be way too much current draw?
  - In addition to I quickly made a power-distribution board with an Arduino Nano and the Qwiic master branch attached to the I2C line. Keep in mind that right now, the Qwiic system still uses 5V as power, which means it is NOT COMPATIBLE with the standard Qwiic system.

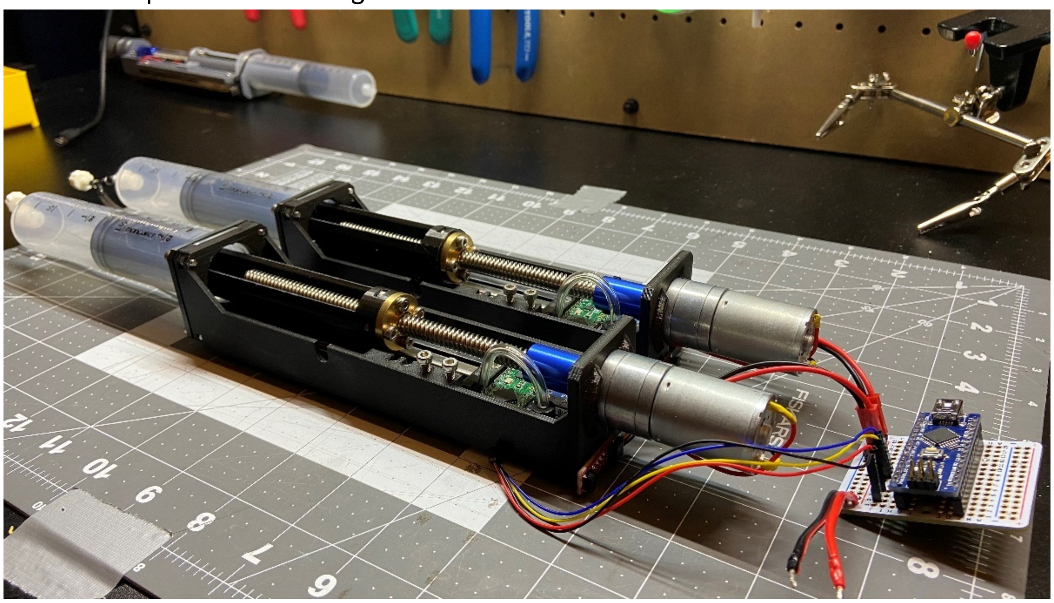
- Minor revisions which do not affect the setup:

- I got rid of the end portion of the air tube organizing channel – it causes the tube to bend, and the system looks better without routing the tube through the channel.
- Added a hole on the inner piston to allow anyone to take off the linear potentiometer screws without the need of dismantling the entire upper assembly.

- Will organize the last few entries tomorrow – need sleep!

April 24th, 2020

- Assembled the second unit with minor adjustments to the CAD model. Druing this time, I recorded the steps of assembly in details in preparation of any open-source publication or tutorials I will be making. I also filmed a time-lapse of me building the unit.



- The rough assembly steps are as follows:

Assembly Steps:

1. PCB

Solder Teensy, MPRLS, H bridge with short male headers. Sand the sensor & H-bridge a little. MAKE SURE TO SOLDER THE H-BRIDGE AT THE OPPOSITE SIDE!!!

Remember to cut the 5v trace on the Teensy

Solder Qwiic with short female headers

Use sandpaper to get rid of tabs ON THE CUSTOM PCB

Test fit onto the base part

Solder resistors (300K for Red, 100K for Green. Red is placed at the front position).

Solder buttons

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!

Fixate the POT to the base. Test fit again with Teensy and LED attached (do not attach MPRLS and H-bridge at this time yet!). Adjust if Necessary.

Solder the LED in position. Make sure they snap to the housing on the base at the other side.

Prepare a 130mm long solid-core wire. Detach the custom PCB, and solder one end of the wire to the POT position on the PCB.

Attach the PCB back to the base, and solder the other end to Position 3 on the linear POT

Keep the custom PCB inside the base, and solder the rest of the POT connections

Solder the wires onto the motor (Red wire – 100cm, Black Wire – 90cm). Mind the orientation of the motor and the color placement. Cover the connection with heat shrink.

2. Mechanical

Carefully take the finished PCB assembly out of the base. Use 2x16mm bolt and insert the button extension mechanism.

Feed in the air tube. LENGTH~, You will have room to trim it down later.

Attach back the PCB. Adjust the bolts until it engages with the buttons well. After that, add the MPRLS and the H-bridge, and secure then PCB with M2 screws.

Attach the motor and insert the motor wires to the PCB from the hole. Do not strip the wire to length until you've made sure your connection is correct.

Attach the VIN wires for the motor. I used a JR connector for my own power distribution board.

Slide the inner-piston in; take off the rubber seal from the syringe and attach it to the front of the inner-piston

Attach the lead screw nut to inner-piston with M3x12 or M3x10. Attach the lead screw to the shaft coupler

Slightly lift the inner-piston and screw the lead screw in the nut.

Optional: apply some Vaseline to the rubber seal.

Attach the inner piston assembly to the POT slider. Make sure the setup could slide back-and-forth smoothly.

Now take the syringe clamp ring and attach it to the outer part of the syringe from the front, slowly fit the piston inside the syringe, and secure the clamp to the base with M3x10 or M3x12 bolts (Nut-pulling). Remember to feed the tube through the hole on the syringe clamp.

3. Pneumatic

Attach the tube ring to the tube.

On the other side, attach 3/32-luer male&female tubing and Y connector. Make sure the tube is routed correctly, and cut the excess from the MPRLS side.

Attach the tube to the MPRLS and secure it with the ring.

Attach your own controller, the power-distribution board, and the unit is good to go!

April 25th, 2020

- Updates on software development:
  - Cleaned up “top_master”, integrated an I2C scanner code and had the program sutomatically detect the addresses of all units inline and store the address in an array.
  - Finished LED sequencing for “mid_master” – “manual mode” is almost finished; I also got rid of the jerk of the motors when they reach the ends of the rail.
- Next:
  - Map the linear potentiometer value to position.
  - Change the data being sent to the top master – add location data.
  - Finish writing “get_data”, “move_pressure”, and “move_pos” in the top master code.
  - Figure out how to use OBS Project.
  - Finish making the third one and document the process.
  - Think about what actuator to use for the demo.
- To fix the oversized files issue in Git : https://medium.com/@marcosantonocito/fixing-the-gh001-large-files-detected-you-may-want-to-try-git-large-file-storage-43336b983272

April 26th, 2020

- Linear Potentiometer Calibration Curve:

Analog Reading	Position
1023	0
1018	5
1007	10
996	15
984	20
971	25
960	30
947	35
934	40
920	45
908	50
891	55
864	60
813	65
715	70
591	75
444	80
260	85
71	90
29	95
1	100