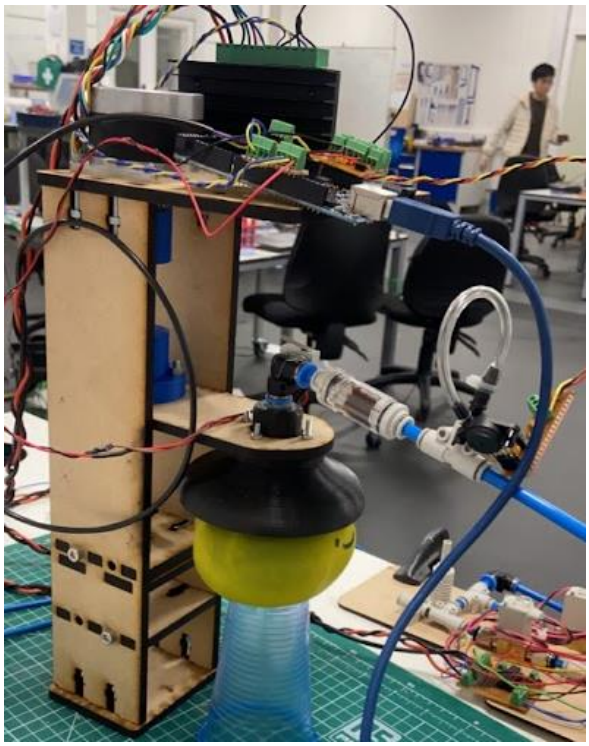
EE30220 Mechatronics 1: Beachcombing Gantry

Harry Mills: **12245**




Summary

This report details the process of designing, fabricating, and testing a pick-and-place gripper mechanism for integration with a 2-axis gantry and hall-effect sensors to scan for magnets in a 500 x 500 mm area and then pick-and-place plastic cups on the magnets.

This report will focus on the development of a soft, non-anthropometric pick-and-place robotic gripper, including: the mechanical hardware, pneumatics, electronics, and the software to control the gripper.

Ultimately, the gripper was developed successfully and picked up cups at least 75% of the time. This project provided valuable insight into the design and development of an integrated mechatronic project and was a good opportunity to research and experiment with soft robotic grippers – a current area of academic interest and research.

Contents

1. General project details

1.1. Task outline

The project involved using a 2-axis gantry covering 500 x 500 mm to scan for magnets of varying sizes using Hall-effect sensors and then place cups where magnets were detected. This was broken down into 3 functions: Software to move the gantry to scan the area and move to magnet locations, Sensor head and electronics to process signals from the Hall-effect sensors, and a pick-and-place gripper to place the cups on the magnets. This system was to be made to run fully autonomously.

1.2. Functional Decomposition

Before starting, a functional decomposition was performed to distribute tasks between team members by function. The overall function of the system is to find the treasure (magnets) – which was then split into 3 sub-functions with set deliverables that each team member would be responsible for.

<table>
<tr><th colspan="3">Sub-Functions:</th></tr>
<tr><th>Move Gantry</th><th>Detect Magnets</th><th>Pick-and-place cups</th></tr>
<tr><td>Home gantry to limit switches</td><td>Attach sensor head to frame</td><td>Gripper system to reliably pick up and drop cups</td></tr>
<tr><td>Scan area as fast as possible</td><td rowspan="2">Electronic signal conditioning/processing (reduce noise, centred at 0V, positive voltage only)</td><td rowspan="2">Attach mechanism to gantry with consideration for sensor head</td></tr>
<tr><td>Take measurements whilst scanning</td></tr>
<tr><td>Record measurement locations and generate magnet heatmap</td><td>Accommodate 3 Hall-effect sensors (HES) to make the scan quicker</td><td>Height adjustability (to avoid placed cups)</td></tr>
<tr><td>Return to magnet locations after scanning</td><td rowspan="2">Make sure both strong and weak magnets can be detected</td><td></td></tr>
<tr><td>Move sequentially between cup storage locations</td><td></td></tr>
<tr><td>Avoid knocking previously placed cups</td><td></td><td></td></tr>
</table>

1.3. Final Design Summary

1.3.1. Gantry Movement

The flowchart below (Figure 1) shows the code of the gantry to achieve autonomous detection of magnets and placement of cups. To summarise: the gantry is ‘homed’ to the limit switches, a scan is performed (Figure 2), the magnetic field peaks are calculated, and then the gantry moves sequentially to each cup and places it on each detected magnet in turn.

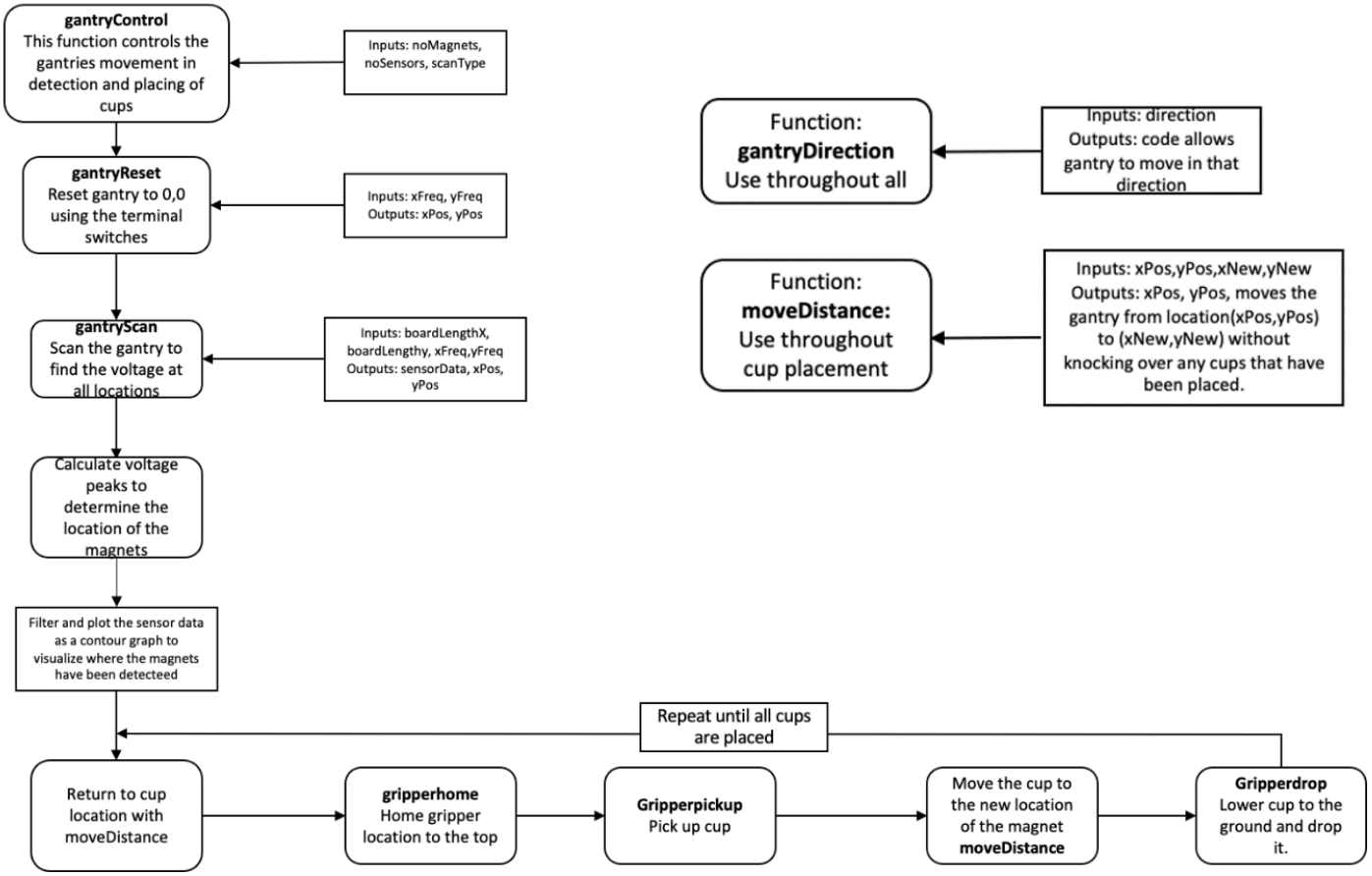


Figure 1 Flowchart of complete gantry movement code

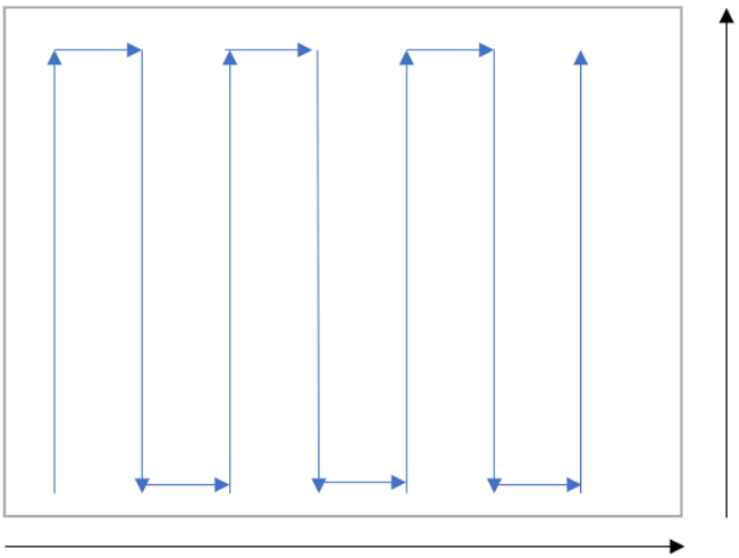


Figure 2 Gantry scan pattern - distance between each line can be modified based on number of HES used

1.3.2. Magnet Detection and Sensor Head

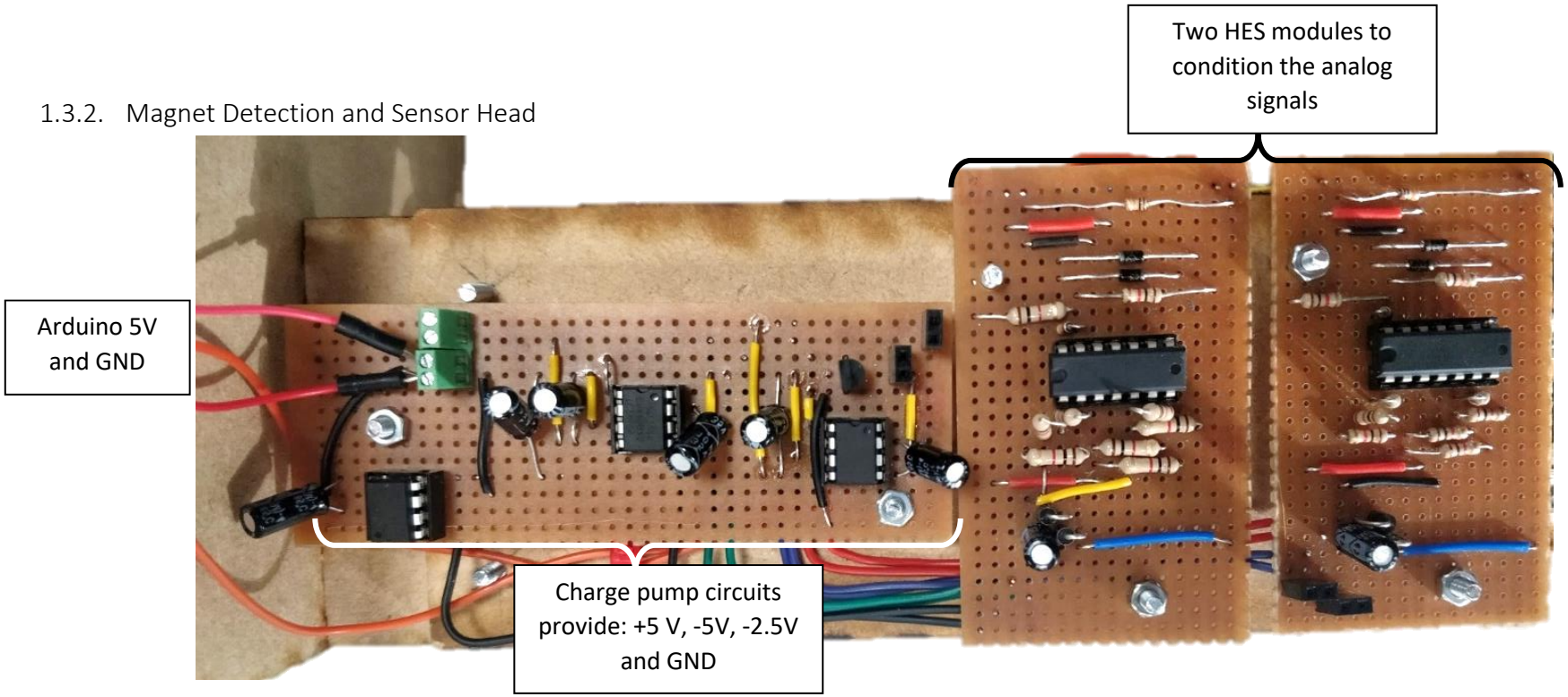


Figure 3 Final electronics assembly for conditioning, processing, and filtering the analog signals from the Hall-effect sensors

Figure 3 shows the final electronics circuitry for processing and conditioning the signal from the Hall-effect sensors (HES). The individual modules (right) process the output of one HES each by splitting the signal positive and negative values with 2 diodes, inverting the negative values, and then summing the signals to get absolute values. To make this possible, each HES is supplied with -2.5V and +2.5V from the charge pump circuits to centre the output signal at 0V.

After this the signal is run through a 16 Hz low-pass filter to eliminate noise and a power diode to reduce the voltage offset. From here the conditioned signal is sent to the Arduino.

1.3.3. Gripper and Gantry Integration

The following section in this report will detail the research, design, and fabrication of the gripper mechanism for the system. A universal jamming gripper with pneumatic control was used to pick-and-place cups, with a stepper motor driven lead screw to control the height of the gripper.

2. Gripper Development

Development of the gripper function of the system incorporated the 3 key aspects of a mechatronic project – mechanics, software, and electronics. This allowed development of the gripper to be done as a sub-project before integration with other aspects of the project.

2.1. Initial Research and Concept Selection

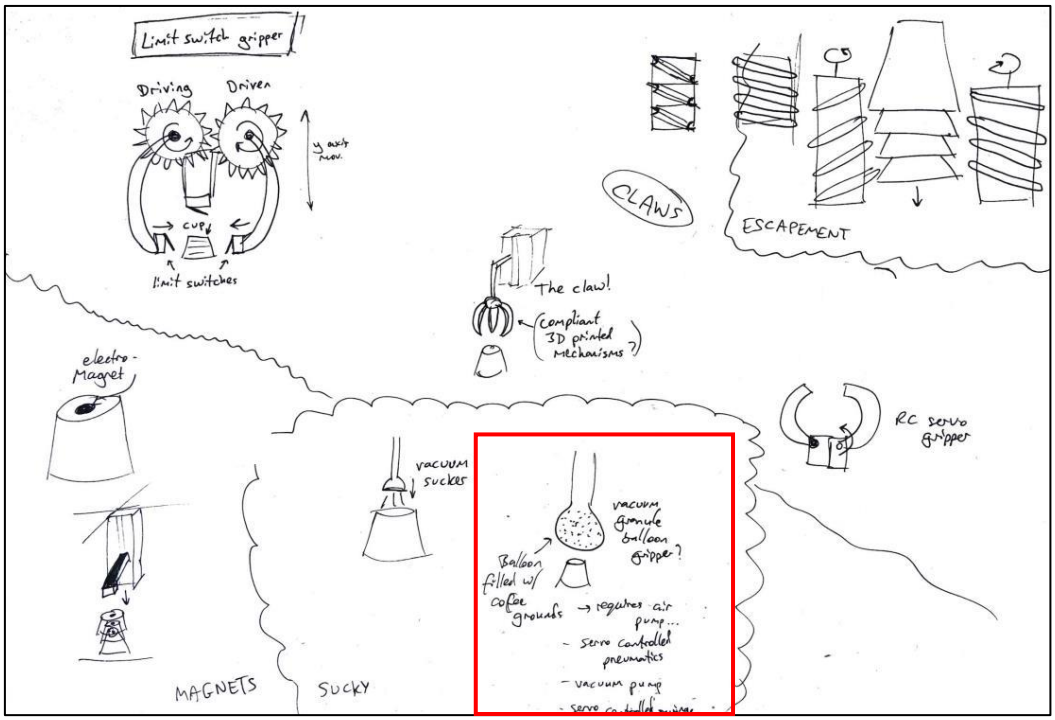


Figure 4 Initial rough concept sketches. The selected concept is highlighted in the red box

Figure 4 shows the initial concept sketches considered for the gripper system. These were selected by considering mechanisms that could pick-up and drop cups – simply a 'pick-and-place' mechanism.

Pick-and-place machines are extensively used in industry – for example: assembling PCBs, handling goods on production lines or sorting objects [1]. There are many different types of grippers used in these applications depending on the object being handled ([2]; Figure 5).

From research on pick-and-place grippers and considering the geometry of the object (plastic cups), 4 options were generated: claw-type (servomotor), electromagnetic, screw-escapement and compressed air grippers (pneumatic/vacuum).

Several factors were then considered to converge to a final concept for further development (in order of importance):

- Novelty (interesting and novel mechanisms are preferred)
- Robustness (how fragile/reliable is it?)
- Simplicity (easy to prototype and build?)

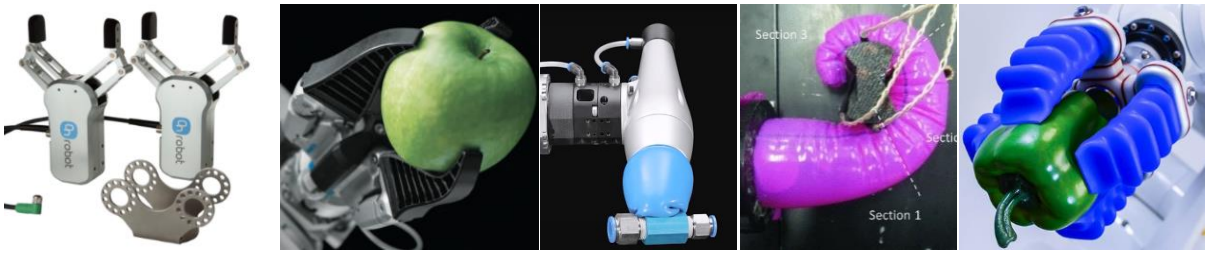


Figure 5 Examples of grippers used in pick-and-place applications.
(Left-Right) 2 finger grippers [3], Festo comlpiant MultiChoiceGripper [4]), Festo FlexShapeGripper [5], compliant tentacle gripper (I. Georgilas et al. 2014 [6]), mGrip soft gripper [7]

This may seem like a counterproductive order of priorities as it encourages more convoluted solutions over simple and reliable methods. However, in the final assessment of the system novelty was scored alongside successful operation and robustness; meaning a reliable and novel system was preferred over a reliable but simple gripper.

In the interest of maximising novelty, a universal jamming balloon gripper based on a concept developed by Amend et al. (2010) [8] was selected. It is a balloon filled with ground coffee that relies on the effect of 'jamming' to grip objects.

Firstly, the gripper passively forms around the shape of the object before a vacuum is applied to grip the object (Figure 7). The vacuum prevents the granules in the balloon from slipping over each other – the jamming effect – making the balloon rigidly form around the object. This type of universal jamming gripper "exploits the temperature-independent fluid-like to solid-like phase transition of granular materials known as jamming." (Amend et al., 2010).

Figure 6 shows holding force tests on this type of gripper for a range of shapes (Amend et al., 2010). Since the plastic cups weigh < 5 g this type of gripper will easily be able to lift the cup and any modifications that add weight to the cup up to at least 1 kg.

A simple claw gripper (left, Figure 5) would have been quicker to design and prototype, and probably more reliable, but lacked any novel or interesting features compared to the soft, non-anthropomorphic, pneumatic/vacuum grippers considered.

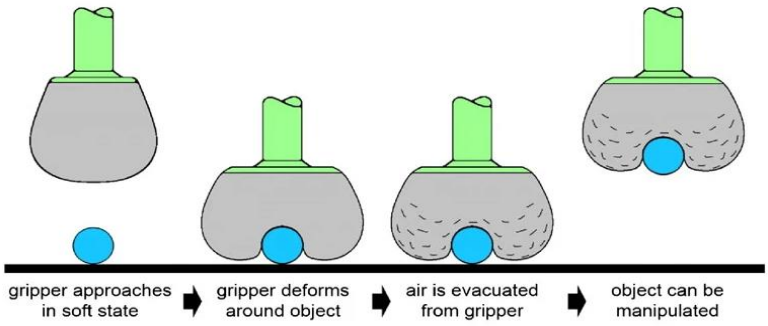


Figure 7 Operation cycle of a universal jamming gripper [8]

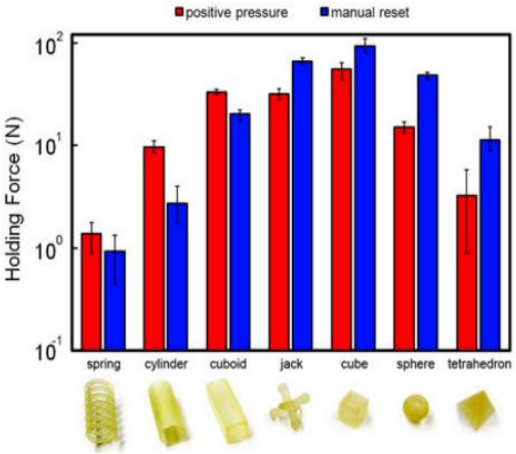


Figure 6 Holding forces of a universal jamming gripper for different shapes [8].

This type of gripper is an example of soft robotics, a recent branch of robotics concerned with "robots that are soft, flexible and compliant, just like biological organisms" [9]. Figure 5 shows some soft grippers. The Festo FlexShapeGripper [5] and the variable compliance, soft tentacle gripper [6] are examples non-anthropomorphic grippers meaning they are not multifingered with independent joints, but have fluid-filled elastic membranes to conform around different shapes – similar to the universal jamming gripper used for this project.

2.2. First Prototype and Proof of Concept

Figure 8 shows the first prototype balloon filled with coffee grounds. Using human powered suction only it was successfully demonstrated that small, light objects could be manipulated with the balloon.

However, human powered suction is greatly limited when trying to create a vacuum in this much volume and requires a lot of effort to reduce pressure below 0.8 MPa.

Figure 8 (Left) Balloon gripper proof of concept (named Esteban) and (right) the coffee grounds used in the balloon

2.3. Gripper Pneumatics

Due to the limitations of human sucking power and the requirement for the system to function autonomously a method of controlling the pressure inside the balloon was required.

This type of gripper has increased gripping ability when a positive pressure is applied while forming around the object since the granules are free to flow around the shape (Amend et al., 2010). Therefore, the pneumatic control was required to supply both positive pressure and a vacuum.

Figure 9 shows the pneumatic schematic allowing control of both negative and positive pressure in the balloon. Positive pressure from the air compressor (SIP 04383 QT100/10 Low Noise Compressor) at 4 Bar either flows directly to the balloon to supply positive pressure, or through a vacuum generator to suck air out of the balloon.

Air flow direction is controlled using two 2/2 normally closed (NC) solenoid valves. 2/2 refers to the 2 ports and 2 positions possible (open/closed), while NC refers to the spring shown on the diagram that returns the valve to the closed position when not powered.

2.3.1. Pneumatic Schematics

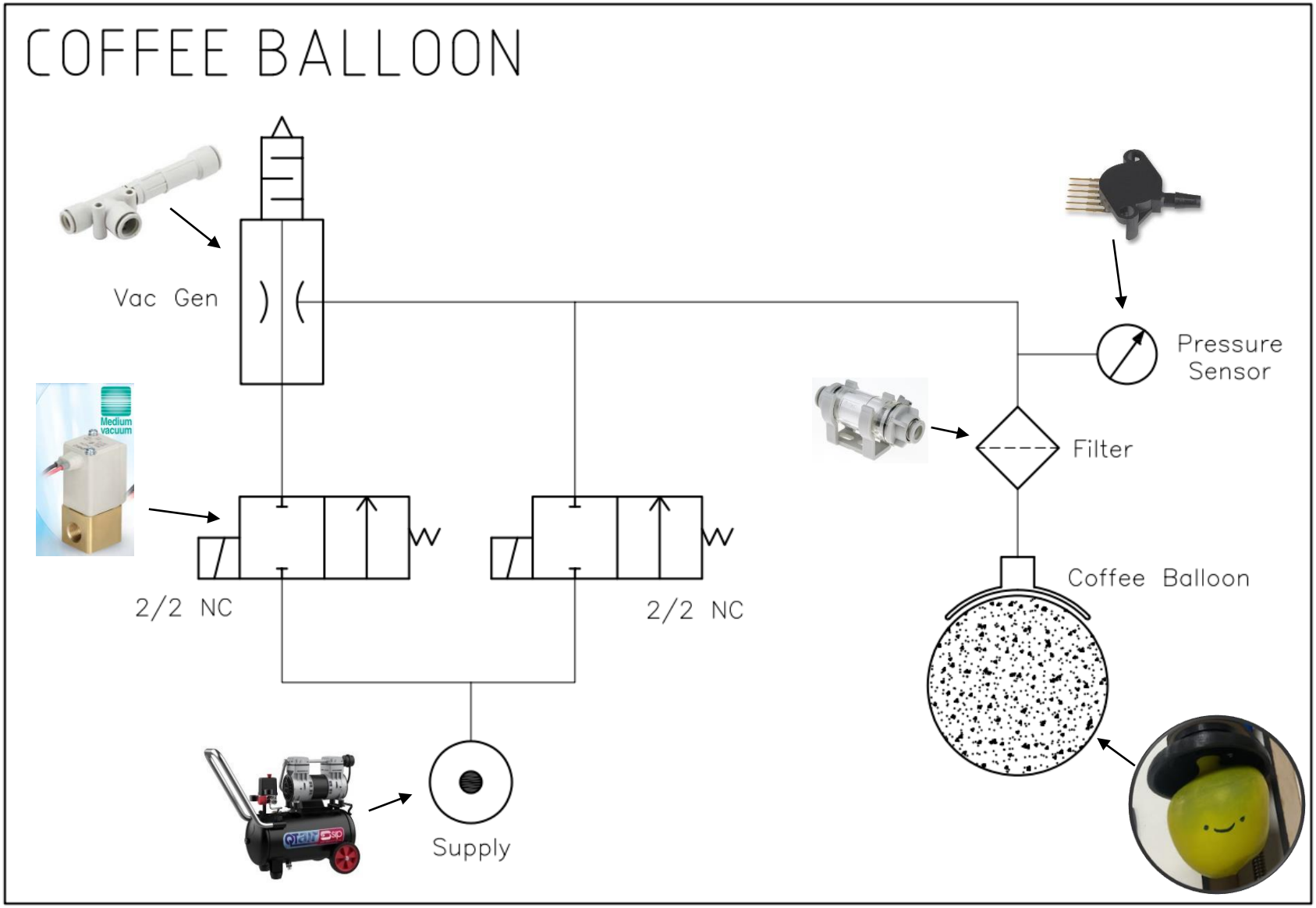


Figure 9 Pneumatic Schematic for controlling both pressure and vacuum in the balloon. Positive air supply is routed either straight to the balloon for pressure or through a vacuum generator for negative pressure

If the left valve is open and the right closed, then air flows through the vacuum generator. This operates according to the Venturi principle to create and negative pressure in the air hose connected to the balloon. Figure 11 shows a cross-section of how these vacuum generators use converging nozzles to accelerate the air flow from left to right, creating low pressure at the inlet port at the bottom – vacuuming air out of the balloon.

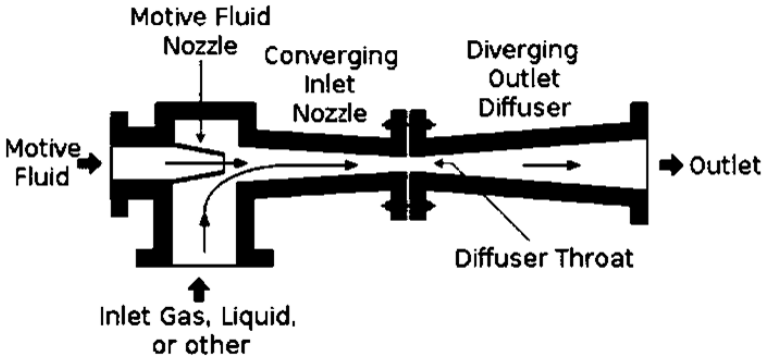


Figure 11 Cross-section of a Venturi vacuum generator [10]

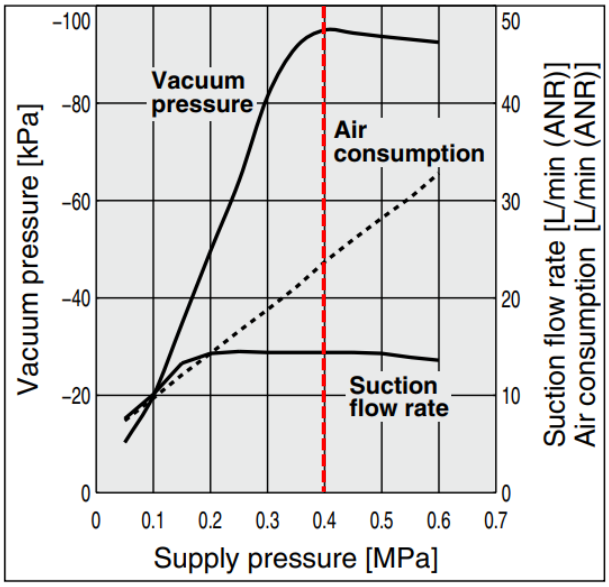


Figure 10 Graph of vacuum pressure against supply pressure for the Venturi vacuum generator used. Peak vacuum -90 KPa achieved at 4 Bar supply pressure

If the right valve is open and the left closed, then air flows directly from the air compressor into the balloon. Importantly, in this state or when both valves are closed there is air leakage from the Venturi vacuum generator (Figure 11) since there is always an open path to atmospheric pressure.

2.3.2. Pressure Sensor Characterisation

At the balloon there is a pressure sensor and inline filter. The inline filter prevents coffee granules from being sucked through the air hose and blocking the nozzles in the vacuum generator or contaminating the air supply. The pressure sensor feeds back a signal to the Arduino of the absolute pressure in the balloon. Characterisation of the pressure sensor is detailed below in Figure 12.

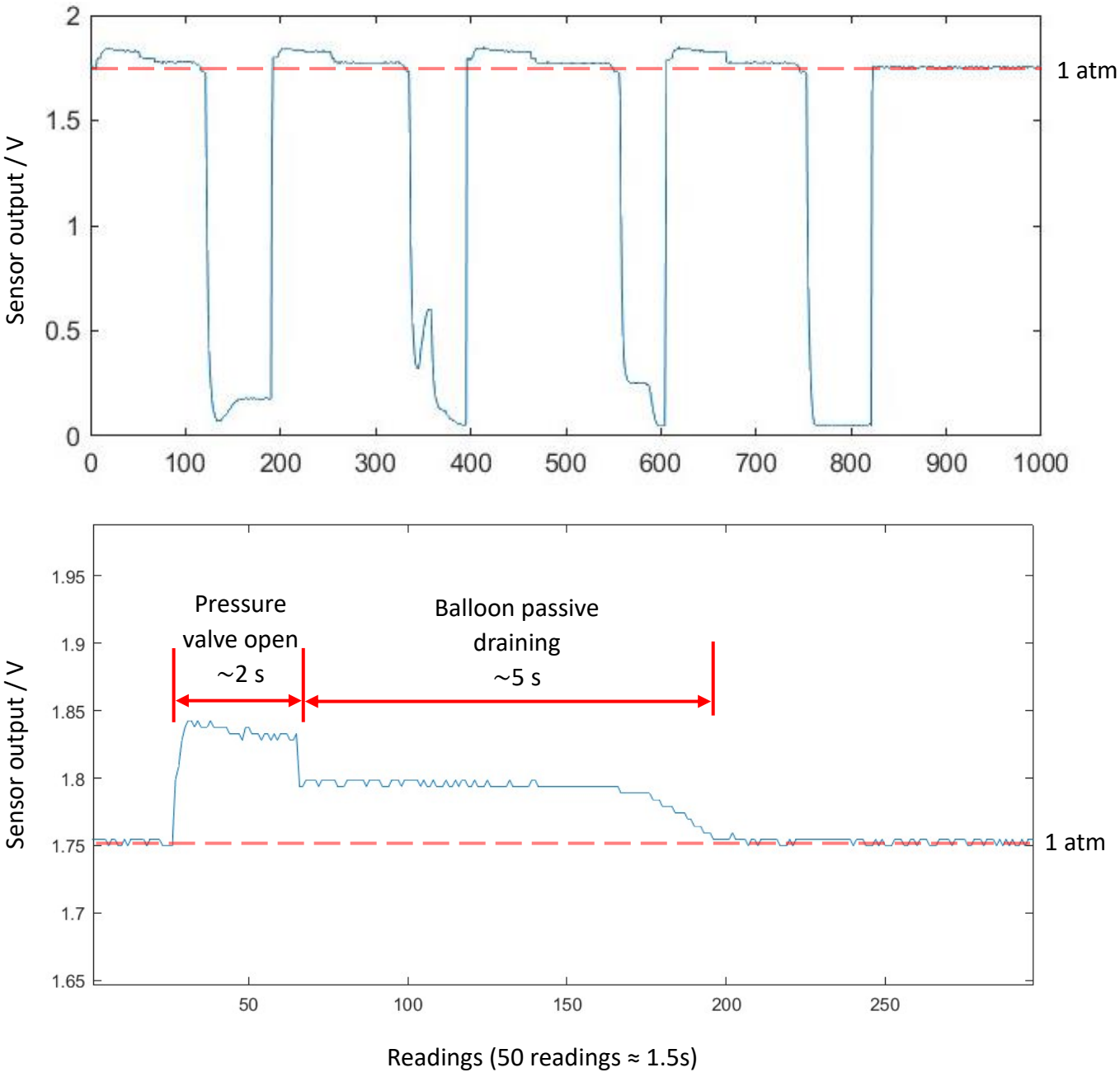


Figure 12 Pressure sensor outputs. Top: Repeated cycles of positive and negative pressure. Bottom: Passive air leakage after a short burst of air.
Voltage readings (x-axis) are taken approximately every 30ms. 100 readings ≈ 3 seconds, although exact timing is not crucial here

The top image in Figure 12 shows the voltage output during cycles of positive and negative pressure. When the vacuum is active the voltage drops to < 0.4V in every case. The bottom image closely shows the pressure sensor output after a short burst of air followed by passive air leakage. While the pressure valve is open the sensor reading is just under 1.85V, this drops to 1.8V when both valves are closed, and the air slowly discharges through the vacuum generator. Atmospheric pressure reads at 1.75V.

These readings are supported by Figure 10 and Figure 13 which indicate that a 0.4V output corresponds to at most 30 kPa (70% vacuum), and 1.75V to approximately 100 kPa.

Figure 14 shows the output filtering circuit recommended in the data sheet for the pressure sensor. Implementation of this and the pneumatics in Figure 9 is shown in Figure 15.

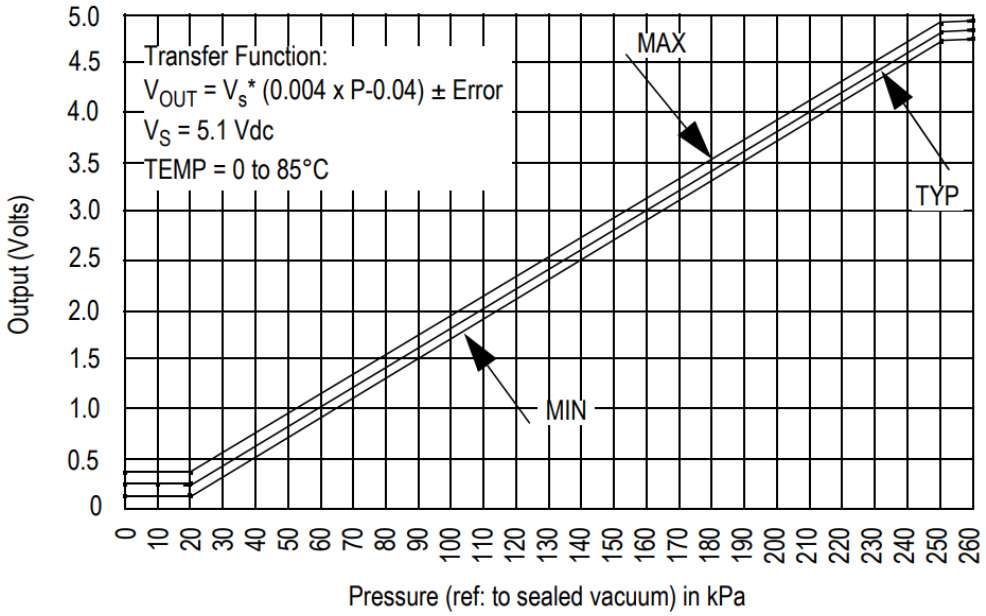


Figure 13 Pressure sensor output voltage against absolute pressure

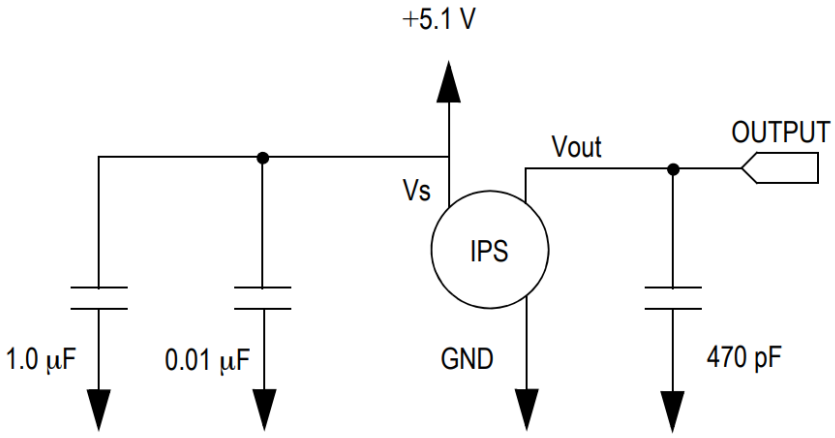


Figure 14 Recommended power supply decoupling and output filtering

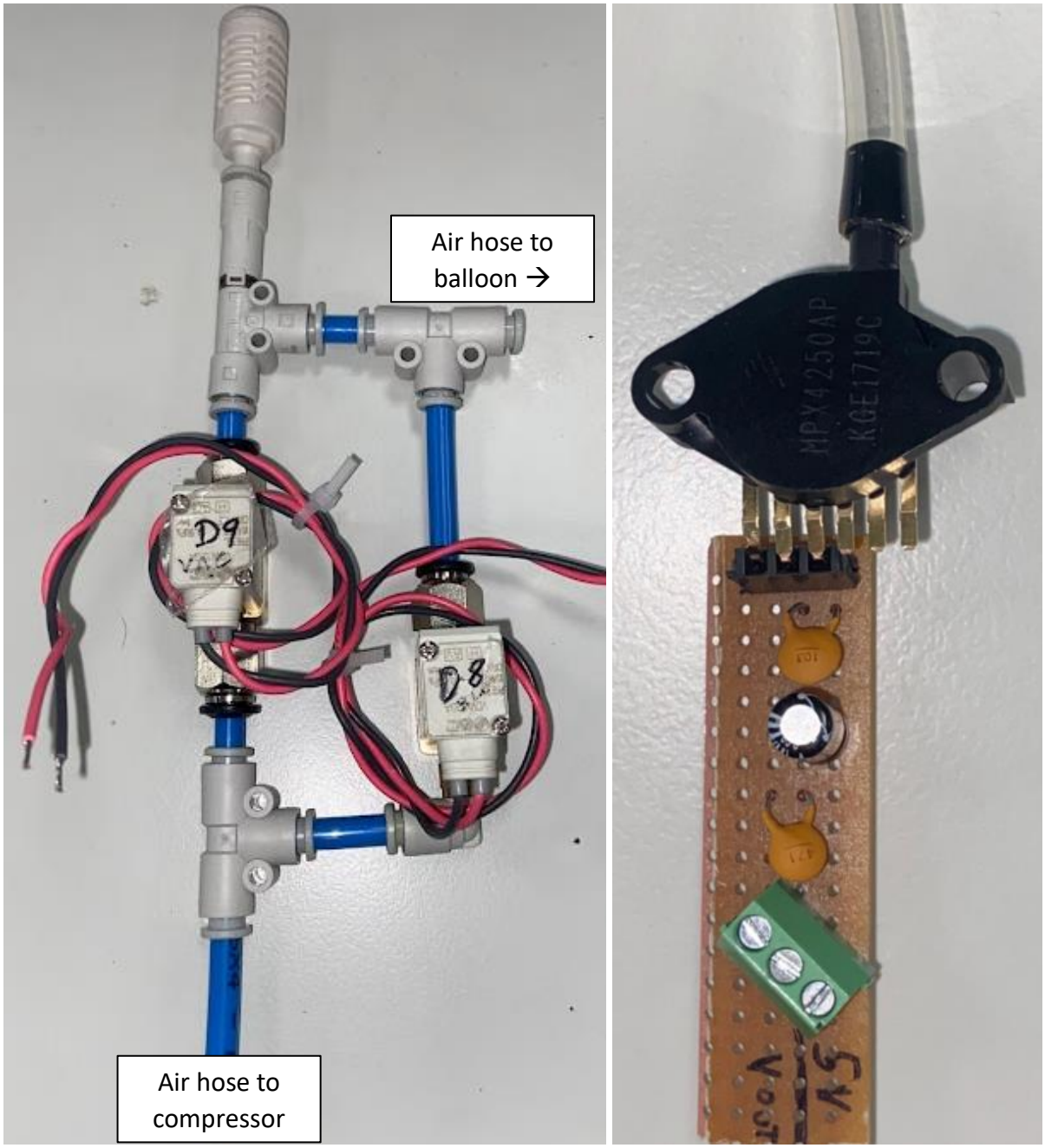


Figure 15 Left: Implementation of the pneumatic schematic shown in Figure 9. Right: Implementation of the output filtering circuit shown in Figure 14

2.4. Gripper Mechanics

Mechanical design for the gripper involved attaching the gripper to the gantry, including a method of height adjustment (so that cups could be picked-up, moved, and placed). Figure 16 shows a cross-section of the final mechanical assembly with key components annotated.

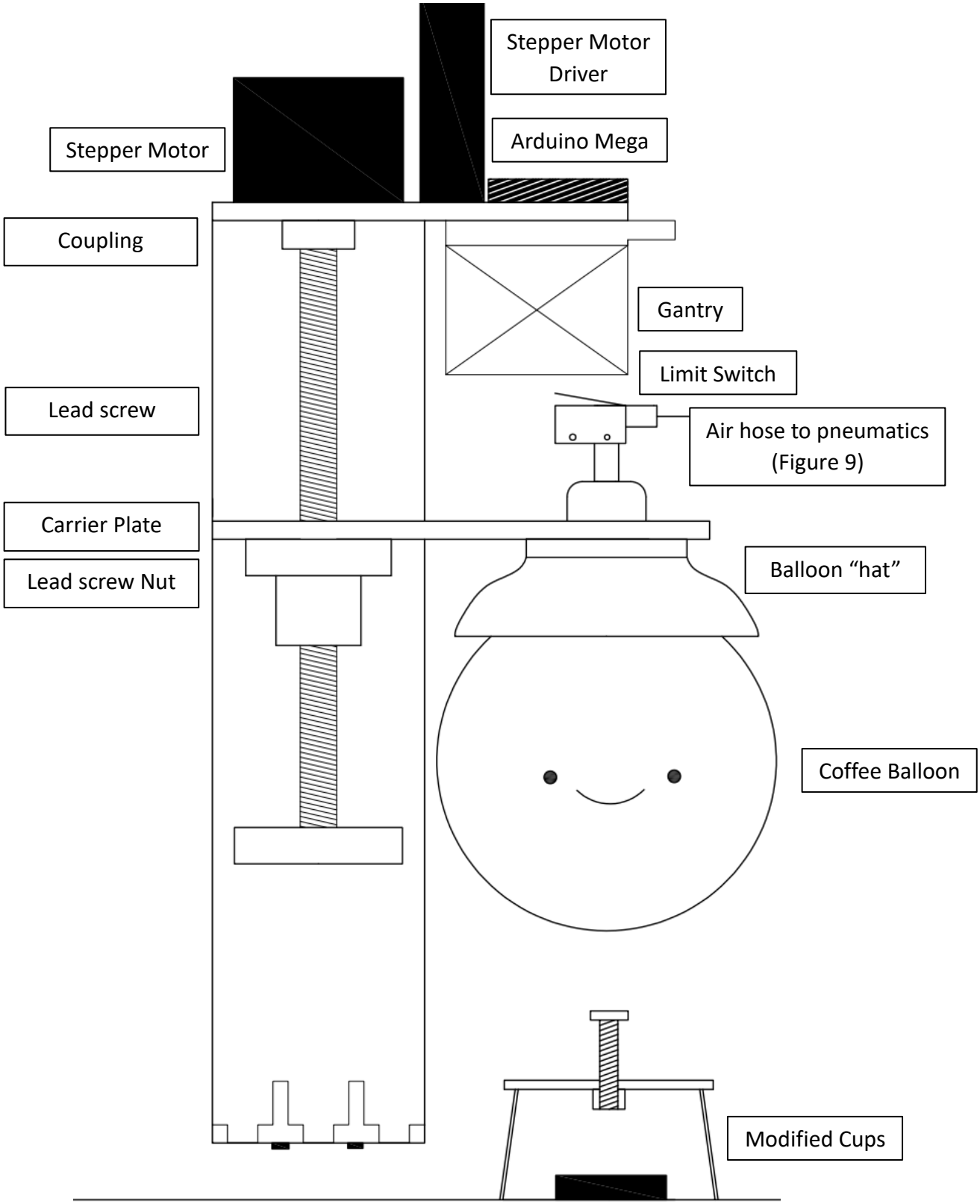


Figure 16 Diagram of mechanical components of the system

2.4.1. Gripper Height Adjustment

A 12mm diameter, 3mm pitch trapezoidal lead screw, driven by a stepper motor, with a 3D-printed flanged lead screw nut was used to adjust the height of the gripper (Figure 17: left).

To prevent rotation of the lead screw nut it was bolted to the balloon carrier which was made to fit as a square inside the vertical parts of the frame – so that rotating the lead screw made the gripper move up or down.

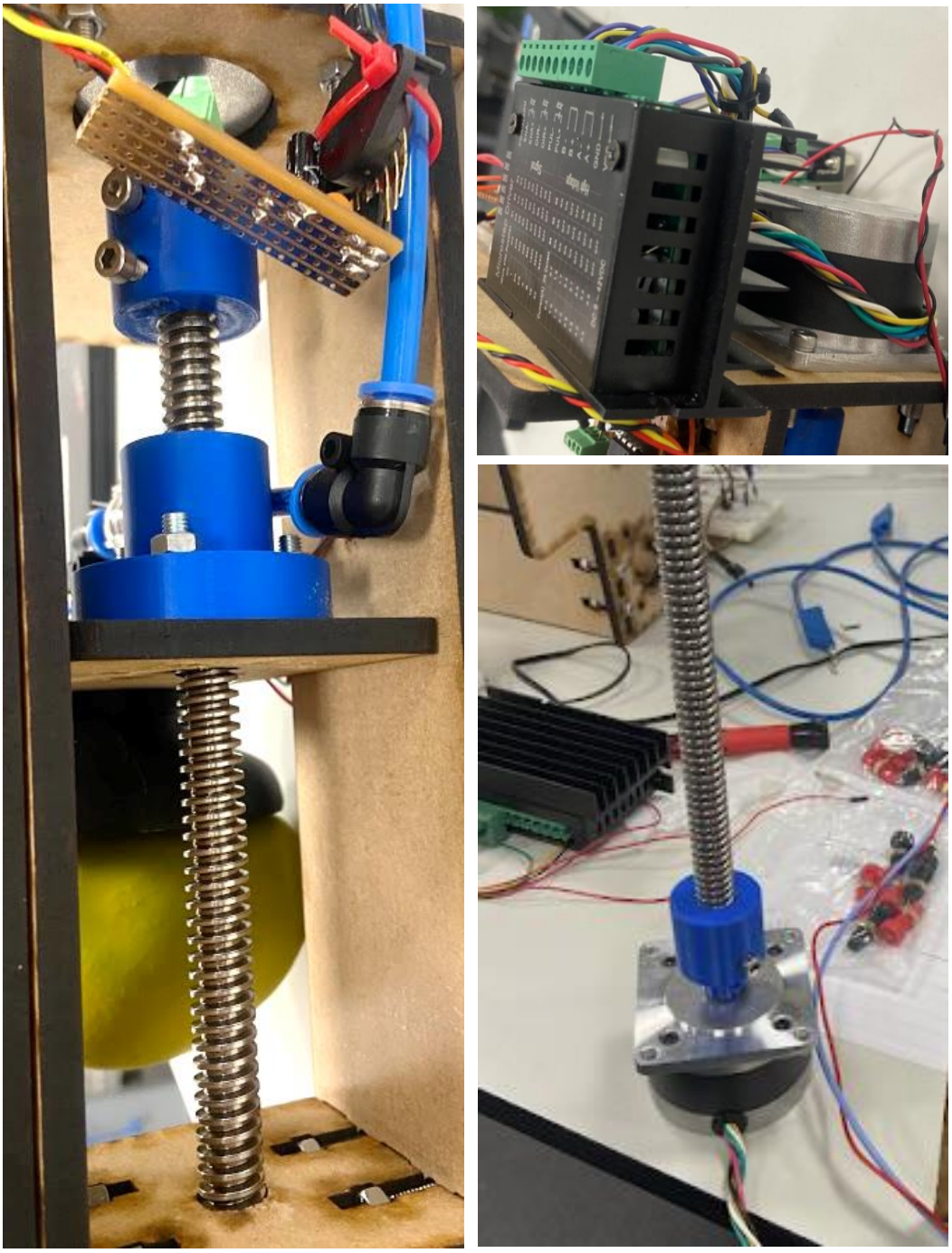


Figure 17 Height adjustment assembly. Left: Assembled on frame lead screw, nut, gripper, and stepper. Top right: Stepper motor and driver attached to frame. Bottom right: Stepper motor coupled to lead screw

2.4.2. Gripper Mechanics Calculations

The weight of the gripper assembly mounted to the lead screw nut was approximately 350g. With one cup gripped the total weight did not exceed 500g (4.9N). Acting at 90mm from the lead screw nut, this puts 0.44 Nm of torque on the lead screw (Figure 18).

It is bad practice in lead screw design to use the lead screw as both the driving component and load guidance member – as has been done here – since it can lead to excessive drag, causing heating and failure of the nut. Additional guide rails to take the torque are required to mitigate this.

Due to the low forces experience in this case, the 3D printed lead screw nut did not break and had minimal backlash during the project.

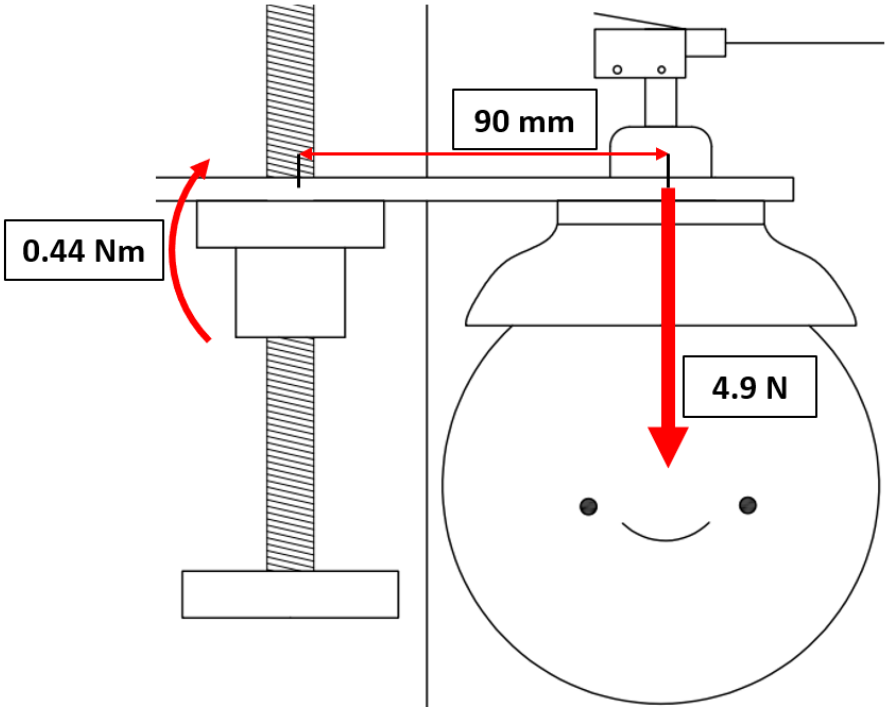


Figure 18 Diagram of torque applied to lead screw nut. This is bad practice in design but due to low forces it does not cause any issues with performance

Motor torque required to drive the lead screw is calculated using equation 1:

$$T_{stepper} = F \cdot \frac{L}{2\pi e} \quad (1)$$

Where $T_{stepper}$ = motor torque, F = linear force, L = pitch and e = lead screw efficiency. Assuming a low efficiency of 20% gives a required stepper motor torque of **0.012 Nm**.

In reality, it is difficult to accurately estimate the efficiency of the lead screw, so a low estimate as a benchmark is a helpful starting point.

A Teflon lubricant was also applied to the lead screw to reduce friction.

2.5. Gripper Electronics

2.5.1. Pneumatics Control

An L293D IC was used to control the signals to the solenoid valves controlling the air supply. This contains 2 H-bridge circuits and can drive inductive loads at up to 600 mA and up to 36V. The solenoid valves require 24V to switch on, this was supplied from an external power supply placed outside of the gantry.

Only 1 H-bridge was required for this application connecting to each switch with output 1 and 2 respectively (Figure 19). Care was taken when writing code and testing not to have both valves open at the same time as this would short the power supply through the L293D.

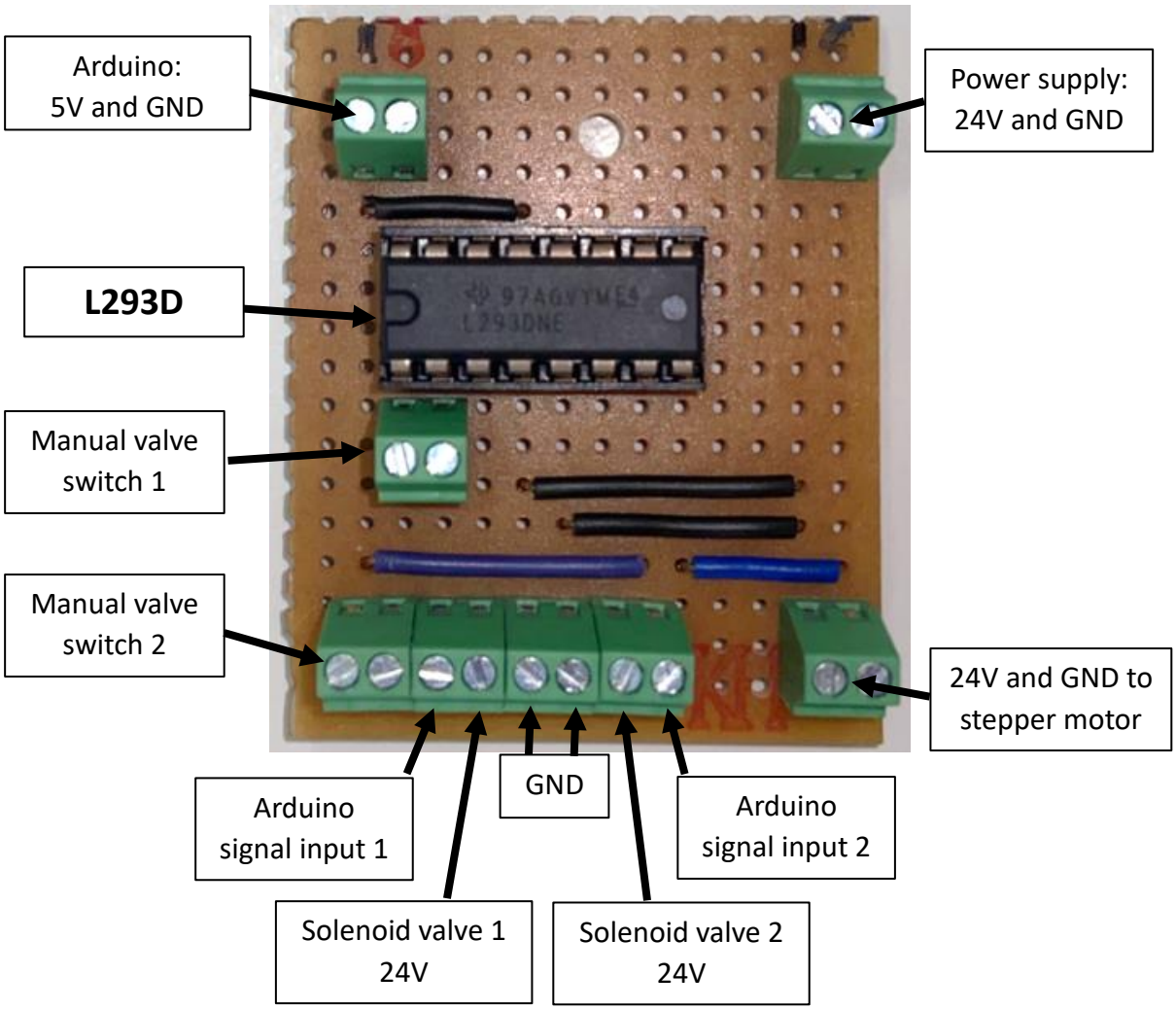


Figure 19 L293D solenoid valve control circuit stripboard.
The manual valve switches connect 5V directly to the Arduino digital inputs to bypass the need to run code to test the pneumatics

The pneumatics and L293D circuit (Figure 15 and Figure 19) were implemented as shown in Figure 20. A front control panel for connecting the external power supply, air supply and manual valve switches was added for ease of testing/set up with coloured LEDs indicating which valve is open.

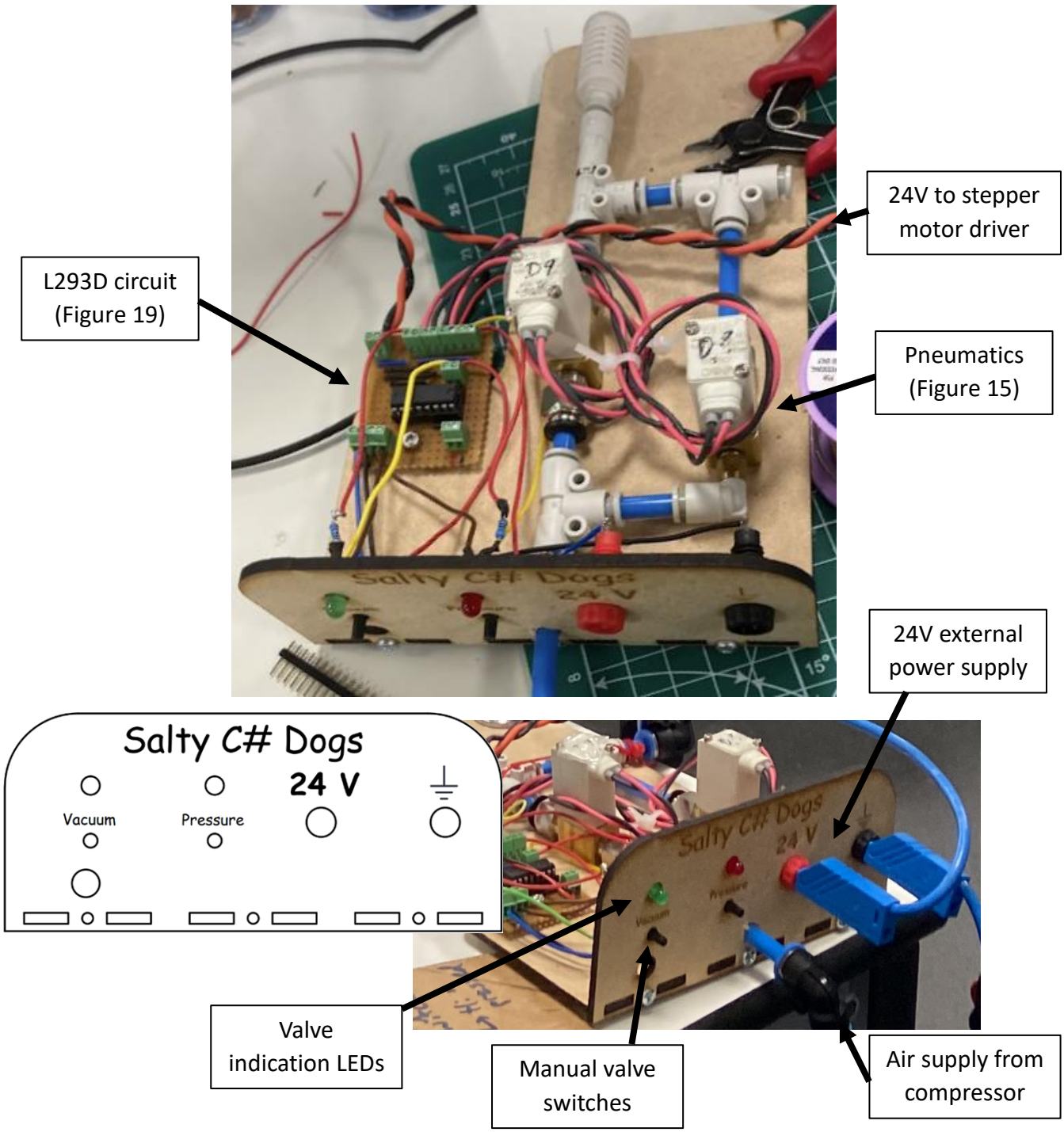


Figure 20 Final pneumatic control subassembly. This is mounted externally to the gantry. Inputs: 24V from external PSU, compressed air, 2 x Arduino digital signals, Arduino 5V & GND. Outputs: 24V to stepper motor driver and air hose to balloon

2.5.2. Motor Selection, Settings, and Limits

A unipolar stepper motor (RS180-5280) was used to drive the lead screw. This has a max current of 1.1A, and a holding torque of 0.29 Nm; much higher than the estimated torque in Section 2.4.2, allowing the gripper to be moved faster, saving time.

Figure 21 shows the connection diagram with a TB6600 stepper motor driver connected to the Arduino, 24V DC, and the stepper motor
Figure 23 shows the microstep settings and current limit set on the TB6600 driver. The current limit was set according to the manufacturer data sheet information. 1/8 microstepping gives 1600 steps/rev, or 1600 steps/3mm of vertical movement.
Figure 22 shows the limit switch attached to the top of the balloon mounting (see Figure 16). A home position for the lead screw was set by driving the gripper upwards until the limit switch pressed on the bottom of the gantry. Distances could then be calculated relative to this height.

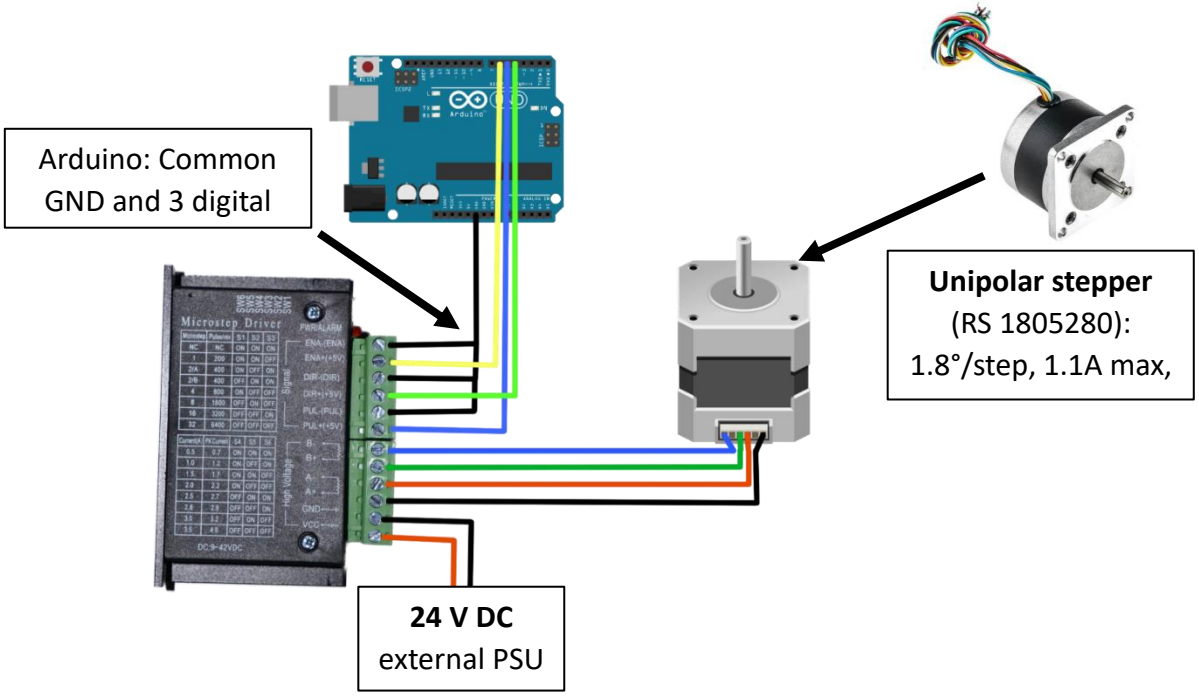


Figure 21 Stepper motor and driver connection diagram; image from: Arduino-projekte.info [11]. Common ground was used for EN-, PUL-, and DIR-; Digital output pins on the Arduino connect to the signal pins on the TB6600. The stepper motor used was unipolar but used in bipolar configuration.

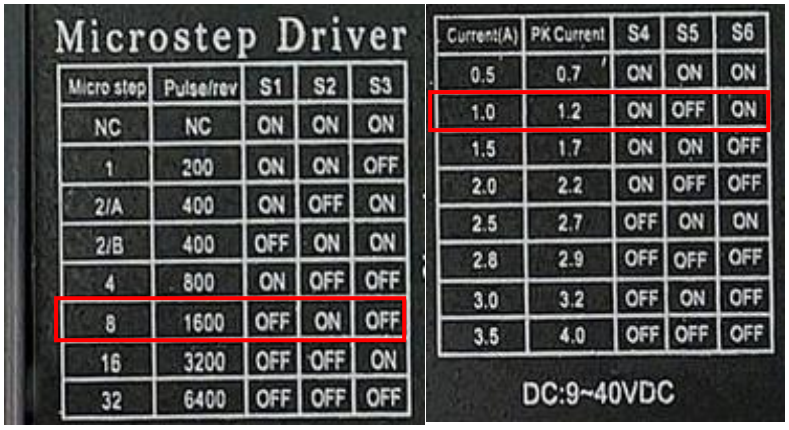


Figure 23 Microstep and current limit settings on the TB6600

Figure 22 Limit switch attached to gripper for homing lead screw

2.6. Gripper Software

Complete matlab code for controlling the gripper is contained in the Appendix

2.6.1. *gripperpickup.m*

Figure 25 shows the function of the *gripperpickup* script. Two key variables in this script are: 'Downdist': (in mm) the distance the gripper moves down on to the cup, and the pause length on line 44 controls how inflated the balloon is as it moves down.

When the gripper has moved into position the pressure valve is closed and the vacuum valve opened. The vacuum valve stays open. When the pressure sensor signal is less than 0.4V (~30 kPa abs.) the gripper moves to the home position, ready for the gantry to move to the magnet

2.6.2. *gripperdrop.m*

Assuming *gripperpickup* has been sent prior to this script, the gripper should be in the home position, vacuum on, holding a cup. This script (Figure 24) lowers the cup, turns the vacuum off and then drops the cup with a short bust of compressed air to the balloon. The gripper then returns to the home position.

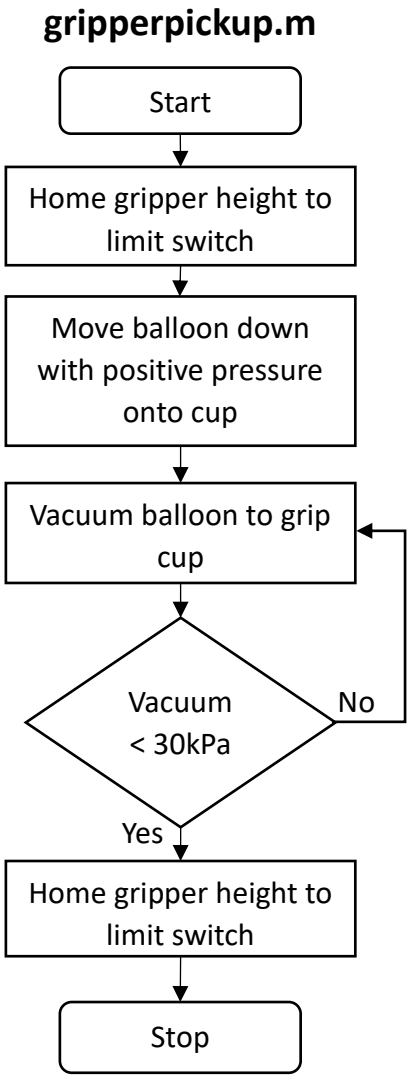


Figure 25 gripperpickup.m flow chart

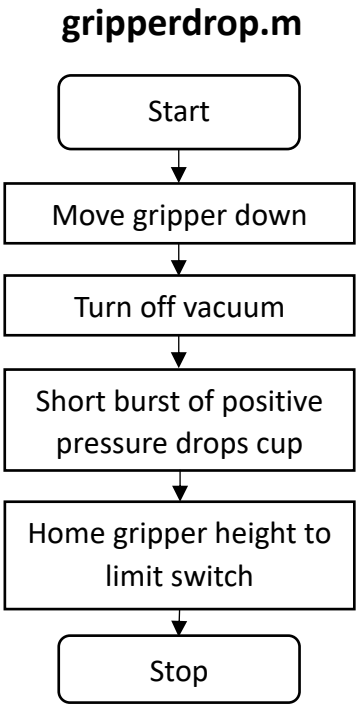


Figure 24 gripperdrop.m flow chart

3. Reflection on Outcomes

Prior to beginning the project, I had hoped to develop my understanding of electronics (analog signals, DSP, motor control), as this was something I was interested in learning more about that was not covered previously on the course. However due to the number of people also wanting to develop their understanding of electronics this was not practical for the project.

A compromise was reached where the project was split into functions assigned to each team member (see Section 1.2). This created a more integrated mechatronic experience when designing the gripper, requiring work in all 3 main branches of mechatronics; Figure 26 shows the integrated nature of the gripper design.

This allowed me to develop my understanding of electronics by using analog sensor signals and signal processing for the pressure sensor, driving the solenoid valves with the L293D circuit, and setting and driving the stepper motor to raise and lower the gripper. An unexpected outcome of the project was learning about and implementing pneumatics for use in the gripper – a valuable lesson as pneumatics is widely used in industry.

If I were to do this project again, a more practical approach would be to use a more standard pick-and-place gripper, like a finger-type gripper commonly used in industrial applications. This would be easier to design, prototype, and probably more reliable (the balloon was ~70% reliable). In the same timeframe this would allow more time for integration and testing – ultimately the most important factors for deciding whether the system will work or not! Although I do not regret using the balloon gripper since it taught me some valuable lessons and was a very interesting and novel concept to experiment with, it was trickly to build and test which could have been avoided using a more standard gripper.

Other potential improvements include: a ‘mechanical thorn’ as described by Reitelshöfer et al., 2014 [12] (where a protrusion into the granules of the gripper increases the strength and rigidity of the gripper - see appendix for more details), or including force feedback in the form of strain gauges on the gripper mounting to stop lowering the gripper when it is pressed onto the object enough – important for picking up objects of different sizes.

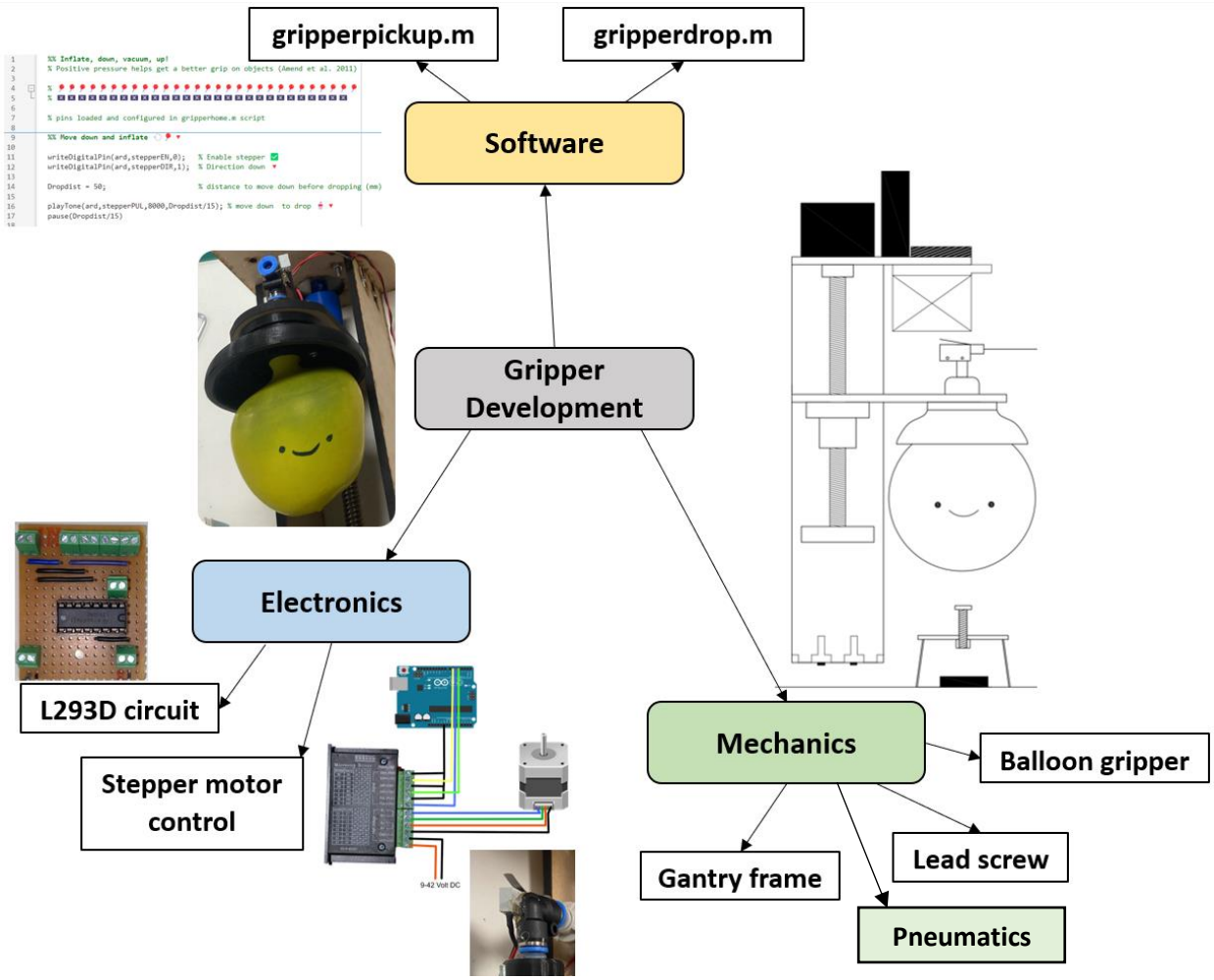


Figure 26 Sub-sections of gripper development integrate work from each branch of mechatronics

4. Conclusion

The design and fabrication of a universal jamming gripper for pick-and-place applications has been described in this report. Using inexpensive materials, it was demonstrated that this gripper could pick up, move, and drop objects successfully by varying the compliance of the gripper. Although this concept is more complex and unreliable than simple solutions such as a 2-finger gripper, choosing a novel gripper concept allowed insight and experimentation with soft robotic grippers, an area of interest for lots of recent academic publications and research.

Design for the gripper was fully integrated including: software, hardware, electronics, and pneumatics. Complete control of all elements of the gripper allowed easy and successful integration of elements within the sub-assembly. Although, integration of all elements of the system was not so successful for the final demonstration.

Personally, I met my Learning Outcomes by developing my understanding of electronics although not to the extent I had hoped due to the logistics of how many people wanted to learn electronics as well. Instead, this project became an integrated mechatronic sub-project with valuable takeaways in all branches of mechatronics.

References

1. PWR Pack. (2022). “What is a Pick and Place Robot and How Does it Work?”. [online] Available at: www.pwrpack.com/what-is-a-pick-and-place-robot
2. Ennomotive. (2022). “7 Types of Robot Grippers and their Industrial Applications”. [online] Available at: www.ennomotive.com/robot-grippers-industrial-applications
3. Image from: OnRobot A/S. RG6 Flexible 2 Finger Robot Gripper. [online] Available at: onrobot.com/en/products/rg6-gripper [Accessed 7 Jan. 2023]
4. Image from: Festo Corporation. MultiChoiceGripper. [online] Available at: www.festo.com/us/en/e/products/festo-gripper-selection-id_406053/ [Accessed 7 Jan. 2023]
5. Image from: Festo Corporation. FlexShapeGripper. [online] Available at: www.festo.com/us/en/e/products/festo-gripper-selection-id_406053/ [Accessed 7 Jan. 2023]
6. Giannaccini, M.E., Georgilas, I., Horsfield, I. et al. A variable compliance, soft gripper. Auton Robot 36, 93–107 (2014)
7. Image from: Soft Robotics inc. mGrip soft gripper. [online] Available at: www.softroboticsinc.com/products/mgrip-modular-gripping-solution-for-food-automation/ [Accessed 7 Jan. 2023]
8. Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M. R., et al. (2010). Universal robotic gripper based on the jamming of granular material. Proceedings of the National Academy of Sciences, 107(44), 18809–18814.
9. Bristol Robotics Lab. (2022). “Soft Robotics”. [online] Available at: www.bristolroboticslab.com/soft-robotics
10. Supreme air products. (2022). “What is a Venturi vacuum?”. [online] Available at: supremeairproducts.com/blog/what-is-a-venturi-vacuum/
11. Image from: Arduino Projekte. Controlling a stepper motor with Arduino + TB6600. [online] Available at: arduino-projekte.info/schrittmotor-ansteuern-mit-arduino-tb6600/ [Amatccessed 8 Jan. 2023]
12. S. Reitelshöfer, C. Ramer, D. Gräf, F. Matern and J. Franke, "Combining a collaborative robot and a lightweight Jamming-Gripper to realize an intuitively to use and flexible co-worker," 2014 IEEE/SICE International Symposium on System Integration, 2014, pp. 1-5, doi: 10.1109/SII.2014.7028001.

Appendix

Matlab code for operation of the gripper. Commented with emojis 😄

gripperhome.m

```
%% raise gripper at start
% so it doesn't hit cups as it goes to pick them up

%% configure pins

load('gripperpins.mat') % load Arduino pin numbers into workspace

% L293D inputs
configurePin(ard,vacvalve,  'DigitalOutput')
configurePin(ard,pressvalve,'DigitalOutput')

% stepper driver TB6600 signal inputs
configurePin(ard,stepperEN,  'DigitalOutput')
configurePin(ard,stepperDIR, 'DigitalOutput')
configurePin(ard,stepperPUL, 'DigitalOutput')

% stepper limit switch
configurePin(ard,limitswitch,'pullup') % enable the pullup - check this works?

% pressure sensor balloon
configurePin(ard,pressuresen,'AnalogInput')

%% home balloon to limit switch 🏠

writeDigitalPin(ard,stepperEN,0);   % Enable stepper ✅
writeDigitalPin(ard,stepperDIR,0);  % Direction up  🔺

while readDigitalPin(ard,limitswitch) == 0 % drive while limit switch not pressed
    playTone(ard,stepperPUL,6000,0.2); % ⬆️
    pause(0.11);

end

pause(0.5)

% release limit switch
writeDigitalPin(ard,stepperDIR,1); % Direction down 🔻
playTone(ard,stepperPUL,6000,0.6); % move down a bit to release switch
pause(0.5);

writeDigitalPin(ard,stepperEN,1);   % Disable stepper ❎ or it will BURN!!! 🔥🔥🔥
% add more enable/disables if adding more current (>1 amp hold)

configurePin(ard,stepperPUL,"Unset")
```

Figure 27 gripperhome.m script. Loads and configures pins before homing the gripper to the top of its stroke

gripperpickup.m

```
%% Inflate the balloon, move it down, vacuum, pick up cup!
% Positive pressure helps get a better grip on objects (Amend et al. 2011)

% 🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈
% ❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎

% pins loaded and configured in gripperhome.m script

%% home balloon to limit switch 🏠
% If motor gets too hot add enable/disable after each movement so it isn't
% permanently in hold - or turn down current limit 🔥🔥🔥

writeDigitalPin(ard,stepperEN,0);   % Enable stepper ✅
writeDigitalPin(ard,stepperDIR,0);  % Direction up 🔺

while readDigitalPin(ard,limitswitch) == 0 % drive while limit switch not pressed

    playTone(ard,stepperPUL,6000,0.2);
    pause(0.11);

end

playTone(ard,stepperPUL,0,0) % stop when limit switch pressed - ask Dave why needed
pause(0.5)

% release limit switch
writeDigitalPin(ard,stepperDIR,1); % Direction down 🔻
playTone(ard,stepperPUL,6000,0.6); % move down a bit to release switch
pause(1);


%% Inflate and move down 🎈🔻
% 1600 steps/rev = 2mm height/rev (leadscrew pitch: 2mm)

writeDigitalPin(ard,pressvalve,1); % open pressure
pause(1)
writeDigitalPin(ard,pressvalve,0); % close pressure

Downdist = 75; % distance to move down (mm)

playTone(ard,stepperPUL,4000,Downdist/7.5); % move down onto cup @ 7.5 mm/s 🥤🔻 (time=dist/speed)

for i=1:7
    pause(1.1)
    writeDigitalPin(ard,pressvalve,1); % close pressure valve
    pause(0.2)
    writeDigitalPin(ard,pressvalve,0)
end

%% Vacuum and move up
% the balloon is squidged on the cup (hopefully) at this point

writeDigitalPin(ard,vacvalve,1);   % This sucks 😎
pause(1)

while readVoltage(ard,pressuresen) > 0.4 % check vac pressure until it reaches 0.4 V (~30 kPa)
    pause(1) % wait for vacuum to suck < 30kPa (absolute)
end

writeDigitalPin(ard,stepperDIR,0); % Direction up 🔺

while readDigitalPin(ard,limitswitch) == 0 % while limit switch not pressed

    playTone(ard,stepperPUL,6000,0.2);
    pause(0.11);

end

% release limit switch
writeDigitalPin(ard,stepperDIR,1); % Direction down 🔻
playTone(ard,stepperPUL,6000,0.6); % move down a bit to release switch
pause(1);

writeDigitalPin(ard,stepperEN,1) % disable stepper ❌ or it gets hot!
configurePin(ard,stepperPUL,"Unset")

% the cup is ready to move to the treasure arhhh yohoho 🥤❎
```

Figure 28 gripperpickup.m script

gripperdrop.m

```
%% Inflate, down, vacuum, up!
% Positive pressure helps get a better grip on objects (Amend et al. 2010)

% 🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈🎈
% ❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎❎

% pins loaded and configured in gripperhome.m script

%% Move down and inflate 🎈🔻

writeDigitalPin(ard,stepperEN,0);   % Enable stepper ✅
writeDigitalPin(ard,stepperDIR,1);  % Direction down 🔻

Dropdist = 50;                      % distance to move down before dropping (mm)

playTone(ard,stepperPUL,8000,Dropdist/15); % move down  to drop 🍦🔻
pause(Dropdist/15)

writeDigitalPin(ard,vacvalve,0);    % close vacuum
writeDigitalPin(ard,pressvalve,1);  % open pressure
pause(0.2)
writeDigitalPin(ard,pressvalve,0);  % close pressure

% hopefully the cup has been dropped - add more weight if sticky balloon

%% Move up to home 🏠

writeDigitalPin(ard,stepperDIR,0); % Direction up 🔺

while readDigitalPin(ard,limitswitch) == 0 % while limit switch not pressed

    playTone(ard,stepperPUL,6000,0.2);
    pause(0.11);

end

% release limit switch
writeDigitalPin(ard,stepperDIR,1); % Direction down 🔻
playTone(ard,stepperPUL,6000,0.6); % move down a bit to release switch
pause(1);

writeDigitalPin(ard,stepperEN,1) % disable stepper
configurePin(ard,stepperPUL,"Unset")

% the cup has been placed on treasure (hopefully!) 🍦❎
```

Figure 29 gripperdrop.m script

Improved Balloon Interface

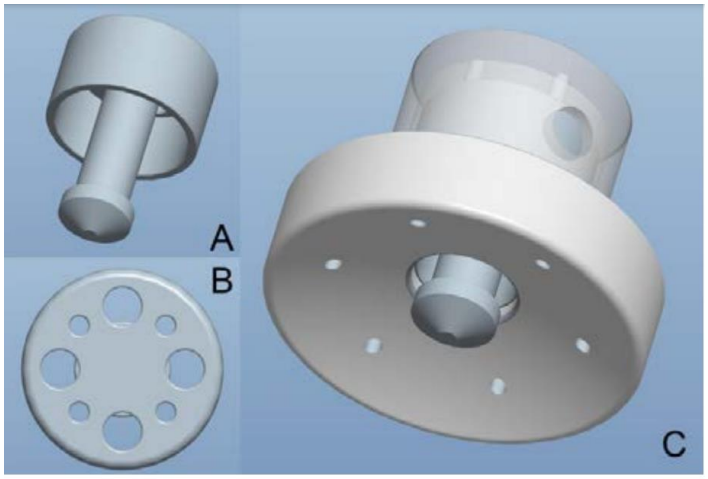


Fig. 7. The mechanical thorn A, with holes in its top part B allowing the air flow reaching through the basis of the gripper C into the granular material.

Figure 30 Reitelshöfer et al. 2014 [12], describe the construction of a similar universal jamming gripper. One suggested improvement on the original design is this 'mechanical thorn'

The “mechanical thorn” described in Figure 30 [12] reaches into the granular material inside the balloon. When a vacuum is applied this rigidly held in place and supports the weight of the balloon and any object it picks up instead of all the weight being transferred through the wobbly and weak neck of the balloon – as was seen in this project (Figure 31). This would be an improvement undertaken given another iteration of this design

Figure 31 Balloon neck supports the weight of the balloon