Design of a Biocompatible, Handheld, Reversible Pressure Pump

DAN 10 – Final Project Report

Presented to:

Prof. Mark Driscoll

Group 11

Donovan Blais

Louis Bourque

Siddharth Raghavan

Mark Sebastian

Team Advisor:

Emily Newell

April 10th, 2020

McGill University

Montreal, Quebec

Liability Statement

This project is part of a Mechanical Engineering Project Course of the Faculty of Engineering of McGill University. It is meant to fulfill academic objectives by having students apply their knowledge to the development, design, and construction of a prototype, which responds to the needs identified by the sponsor. The project is first and foremost a training tool for future engineers, which brings invaluable experience to us. The fact that the prototype developed by the students be fit for commercial, industrial, or private use is not a primary objective of the project. We did our best to provide you with a machine that works, but it is not necessarily fit to withstand the rigors of regular use. We do not guarantee that it will perform adequately and consistently if you put it to regular use. We are proud to have built a prototype, which apparently responds to your requirements. However, please keep in mind that it is only a prototype. If you are considering putting this prototype to use, you must ensure that it is first fully checked for safety. It may be necessary, for example, to provide emergency shut-off switches and other safety features. Some safety recommendations may be found in this report, but there may be others we did not think of. Furthermore, some features of this prototype have been designed with safety in mind; therefore, disabling or modifying parts of the equipment may have a direct impact on the safety of those using it. The safety of those using the machine should always be a primary concern and users should be properly trained to avoid accidents. McGill University, its students, employees, professors, agents and governors decline any responsibility with respect to this prototype, its performance and safety.

Additional Information

Due to current exceptional circumstances, the Exhibition date scheduled in early April 2020 has been cancelled. As such, the client is not yet in possession of the prototype. The client will take possession of the prototype at the earliest possible time. The prototype development is documented in the problem statement, conceptual design report, design report, and detailed drawings, in which the client had an opportunity to contribute to through regular meetings and feedback. A summary of the results presented in this documentation can be found in the attached paper which also describes prototype manufacture and the progress made in obtaining performance results.

We would like to thank Emily Newell for her constant dedication in advising the team and providing excellent guidance throughout the project. Her input was highly valued and proved very useful. Additionally, we would like to thank the client, Professor Mark Driscoll, for his technical and financial support, making this project possible.

Donovan Blais

Louis Bourque

Siddharth Raghavan

Mark Sebastian

Table of Contents

List of Tables

List of Figures

Executive Summary

The study of intra-abdominal pressure (IAP) can lead to many advancements in the medical field due to its close ties to intra-abdominal hypertension (IAH) and spinal stability. Current methods of measurement consist of a stopcock system that requires long invasive procedures. The following detailed work entails the development of a device that allows a quick, non-invasive, and cost-effective process of measuring IAP in patients. The reversible pump measures the skin's displacement response to ± 150 mmHg of pressure and plots the linear response. Using previously published research on IAP's correlation with skin deformation, the device allows for IAP diagnosis. The device is operated with the use of only one hand and is fully comprised in a single housing. The measurements for pressure display a resolution for 1 mmHg, and the displacement resolution is 1 mm. The pump consists of a syringe activated by turning a lead screw put pressure against the skin through a medical tape membrane in contact with a patient's skin. Due to unforeseen events, the final construction of the prototype was not possible. However, the proof of concept is proven to work through the testing of the individual components. For the prototype, expenses per unit were different than expectations due to individual piece costs, and in-house 3D manufacturing. The following report details the design of the prototype along with possible choices to ensure proper manufacturing costs in the future.

Introduction

It is the overall objective of this project to accurately measure IAP through an external device being applied directly on a patient's skin. The focus on IAP measurement is due to IAP's assistance in spinal stability during everyday motion by reducing compressive forces, whereby soft tissue support alone may prove insufficient [1]. The effect of the rising pressure in the system has been measured to reduce the forces experienced by the spine by up to 40% [1]. Thus, the measurement of the pressures experienced in the abdominal cavity has been a large area of study to better comprehend both how stresses in the body are naturally dealt with, along with how injuries can be prevented. This measurement, however, is commonly done through invasive methods due to the difficulty of accessing the abdominal cavity. Found between the diaphragm and the pelvic cavity, the abdominal cavity is surrounded by a double layer of tissue that contains a fluid that pressurizes and protects the cavity. Thus, measurements cannot be done by penetrating the tissue, and are instead done through an intravesical method in which a catheter is inserted into the bladder and subsequently filled with saline solution to allow for internal pressure to be read via a pressure transducer.

This method has been necessary in the past as the study of a body's IAP is important in terms of the medical treatment of intra-abdominal hypertension (IAH) and abdominal compartment syndrome (ACS). Caused when pressures within the body increase within the abdominal cavity, IAH and ACS are often the results of catastrophic injuries inside the body. The body cavity having a fixed size leads to the increased fluids pressures to cause internal swelling, harming internal organs and restricting blood flow [2]. Due to being caused by severe accidents, ACS can develop in intensive care unit patients, with 35% of ventilated patients being diagnosed with IAH or ACS [2]. Once diagnosed, a patient must be treated immediately as symptoms may worsen through further organ damage and eventually lead to death, particularly for ACS caused through abdominal aortic aneurysms (AAA), which have a 47% mortality rate [2].

The ability to measure IAP non-invasively has been previously studied and shown possible through the relation of force and displacement on two corpses. These studies were all based on previous studies on abdominal hernias in which the law of Laplace was used to estimate the tensile forces of the abdominal walls [3, 4]. Thus, the non-invasive device intends to use these relations to aid in the measurement of patient's IAP.

Problem Statement

The objective given to the design team is to develop a device with a reversible pressure pump for use in the medical field, specifically for the purpose of measuring the IAP of patients. An additional objective is to obtain pressure-volume curves to analyze mechanical properties of the abdominal wall. Previous known methods of measuring IAP are cumbersome, costly, and invasive. The proposed device aims to resolve these issues, delivering an improved IAP measurement device and a better patient experience.

The final product will be a non-invasive method of measurement that can distinguish changes in pressure and deformation with a resolution of 1 mmHg and 1 mm, respectively. This will be done with a quick reverse pumping system that will provide pressures between -150 mmHg and +150 mmHg. This two-way pump, generating both positive and negative pressures, will require the device to produce and maintain an air-tight seal with patient's skin on contact ensuring accurate IAP measurement.

Ultimately, the device must be handheld and quick to ensure ease of use by the medical professional, allowing for seamless patient experience. Additionally, the device must thus be light and compact, with a power source located either within the housing or externally with minimal connections to avoid interference with the device. Later on, the client expressed their need to have the device to be powered internally, so as to avoid any wiring external to the device which may be a hindrance to operation. Furthermore, the final product must be inexpensive to manufacture (less than $100 per component). This will require simple and effective solutions as too many moving pieces or unique manufacturing costs may result in a product that is too costly to use compared to other methods.

In summary, the objective is to design and build a device to measure the intra-abdominal pressure for patients, and to have this procedure done in a non-invasive, portable, and cost-effective manner.

Evaluation Criteria

The evaluation criteria for the design are the requirements outlined by the client. Prior to DAN-5, the client ranked these requirements in order of importance. The device was designed in accordance with this ranking. The aim was also to fulfill as many of these design considerations as possible.

Table 1: Design considerations ranked by client

Ranking	Design Considerations
1	Safe design with no sharp corners or edges
2	Restricts motion anterior to the body and provides reliable seal against skin
3	Resolution of at least 1mm when measuring skin displacement
4	Resolution of at least 1 mmHg when measuring pressure
5	The device should be able to create positive or negative pressures of at least ±150 mmHg
6	Capable of pressure tuning to within 5 mmHg
7	Pressure versus volume curves displayed in real time as measurement is being taken, as opposed to creating a plot once all data is collected
8	Measurement should take under 10 seconds from start to finish
9	Device can easily be handled by a single hand
10	Measurement should be automatically saved on the device and possible to consult later on, or for comparison to different measurements
11	$100 limit per unit for all manufacturing and component costs
12	Measured data should be displayed directly on a screen on the device (as opposed to viewing on an external device such as a phone or computer)
13	Device should be wireless, and battery operated
14	Device should have a streamlined design in which all components are in a single package

Having these design considerations ranked guided several design choices. For example, the $100 limit per unit was ranked relatively low. Thus, the design team prioritized other design considerations first, while having a general mindset of cost-effectiveness – rather than implementing a strict $100 limit per unit and risking not implementing the more important design considerations.

In general, the results of the client survey pushed the designs of the device toward a function over form style as the client preferred proper measurements and plotting over being all contained in one housing and easy to use. It is important to note that the final design was wireless and had a streamlined design with all the components in one package. Therefore, the low ranking of some design considerations does not necessarily mean that those considerations were not implemented in the final design.

Concept Generation

Morphological Chart

The device was based on the proposed limitations of the device itself. Such a device will use a fluid to create positive and negative pressures on the skin surface, and then measure the corresponding volumetric response of the skin. The proposals based on this concept are further elaborated on as Concepts 1-3.

Table 2: Morphological Chart

Aspect	Option 1	Option 2	Option 3
Skin Contact	Nitrile/Silicone Membrane	Direct adhesion/belt	Air
Pressure Application	Liquid Pump	Air pump	Linear Actuator
Displacement	Optical displacement sensor	Volumetric calculation	Linear actuator displacement
Power	Battery	Plugged in	
DAQ	Arduino	Raspberry Pi	
Results Display	Screen on device	Separate computer	Transfer to USB
Connectivity	Bluetooth	IR	Internet
Housing	CubeSat	Custom 3D Print	Medical grade metal
Pressure measurement	Transducer	Manometer	
Safety features	Fillet	Rubber	Round design
Data processing	On Device	On Computer	

Morphological Chart Legend	
	Concept 1
	Concept 2
	Concept 3

Table *2* is a morphological chart that was first proposed in DAN-5. While these concepts are explained in detail, it was upon receiving further feedback that the device design switched to a combination of these different concepts. This is explored in detail in the Final Design – Motivation of Preliminary Design section. This final design is not written in as part of the Concept Alternatives section to ensure narrative consistency, and to enforce the idea that the final design was not merely an alternative choice, but rather a combination of the best aspects of each concept.

Concept Alternatives

Concept 1

This concept utilizes a linear actuator to create the pressure ranges required by the device. A glycerin-filled cup is placed on the skin, with a membrane separating the fluid cavity from the skin. Glycerin has a higher viscosity than water and is also non-corrosive and non-toxic [5]. The membrane material would be made of nitrile, given their non-irritant properties, and would be lined with a medical adhesive to facilitate skin adherence to the membrane, creating an air-tight seal.

A linear actuator is used to drive a piston containing a fluid, creating the desired pressure ranges required for the device. By measuring the displacement of the actuator and calculating the subsequent volumetric change, it is possible to determine the volumetric response of the entire system, and thus the response from the skin. A pressure transducer will be used to determine the pressure being exerted by the fluid making it possible to then plot the acting pressure on the skin, and its displacement response. As a redundancy, a light sensor is used to determine the displacement of the piston, and thus provide another set of calculations to validate the measurements being made.

A custom three-dimensional (3D) printed body will be made to house the various components of the device, allowing for freedom in the internal layout of the components.

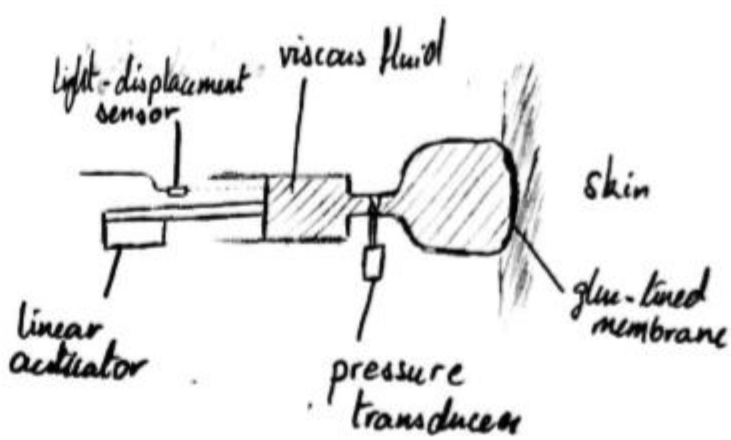


Figure 1: The overall layout of Concept 1

Concept 2

The second concept involves the use of a hydraulic cylinder with a piston. The piston divides the cylinder into two sections, one filled with air and the other filled with liquid. The air chamber is applied directly onto the skin and is held in place by double-sided medical tape.

A thick liquid such as glycerin is pumped from a reservoir into one chamber of the hydraulic cylinder. The pumping moves the piston, creating positive pressure. As the air chamber is adhered to the skin in an airtight manner, this increase in pressure will deform the skin. To create a negative pressure, the glycerin is pumped from the hydraulic cylinder back into the reservoir, such that the air is forced to expand, creating a negative pressure on the skin.

The use of a liquid is important to measuring the displacement of the piston. Since the liquid will not compress, the volume of fluid being pumped is the volume by which the air chamber will change due to the piston. This volume can be determined by measuring the volumetric flow rate of the glycerin as it is being pumped. The effect of the piston's volumetric change on air pressure and how much of it deforms the skin can be determined. The pressure can easily be measured by a pressure transducer connected to the liquid section of the apparatus [6].

Finally, the device housing is a cylindrical shaped, 3D printed casing which holds all components in a single package.

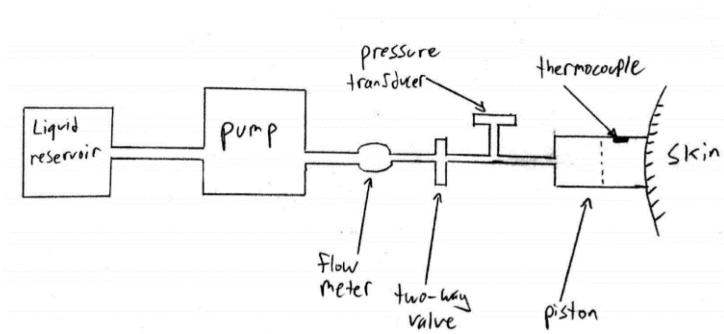


Figure 2: Internal systems for Concept 2

Concept 3

The third concept focuses on a new method of ensuring a proper seal by introducing a pressure vessel that is secured by the patient around the abdomen in the form of a belt. This concept allows the patient to setup the equipment themselves, create a proper seal and prevent the apparatus from moving as the belt is tightened around the body. To prevent fluid leaks at the contact between the apparatus and the skin, double-sided medical tape is used.

The fluid used throughout the system would be air, and thus requires an external vent in the housing that can be opened and sealed with ease using a valve. A vacuum pump is used to compress air and evacuate air from the pressure vessel. The displacement can be calculated by a volumetric flow sensor that measures the volume of air being pumped in and out. Thus, the van der Waals equation can be used to find the change in volume of the abdomen as air is pumped in and out. Next, to measure the pressure of the air, a pressure transducer is connected to the tubing that connects the pump to the pressure vessel.

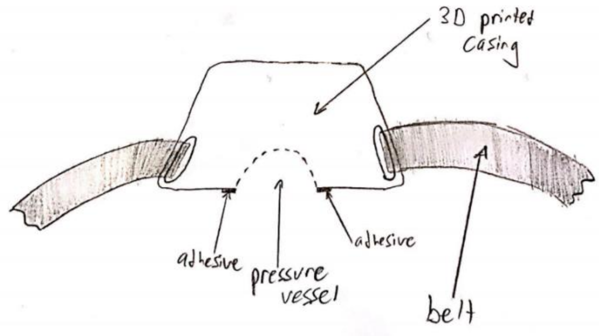


Figure 3: Concept 3 diagram

The design houses all components in a single device. However, it will require fixtures to attach it to the abdomen with a belt.

Concept Evaluation

To compare and choose the best proposed design, a weighted Pugh's matrix will be used. The weights included for the decision criteria were a combination of the results from the QFD shown in Appendix A (to recognize most important engineering metrics) and from the design considerations ranking list (Table 1) provided by the client.

The Pugh's matrix accounted for the different criteria and compared alternatives to a fixed datum based on those criteria. In Table *3*, the fixed datum was Concept 2, while Concept 1 and Concept 3 were compared to that fixed datum.

Table 3: Pugh's Matrix

Criteria	Weight	Concept 1	Concept 2	Concept 3
Pressure Inducer	5	+	0	-
Microprocessor/ DAQ	4.5	0	0	-
Pressure Sensor	4	+	0	-
Time of Operation	4	+	0	0
Displacement Sensor	3.5	+	0	-
Cost	3	-	0	+
Device Size and Weight	2.5	-	0	+
Power Consumption	2	0	0	-
Material Housing and Manufacturing	1.5	0	0	0
Total +		16.5	0	5.5
Total -		5.5	0	-19
Overall Weighted Total		11	0	-13.5

As seen in the Pugh's matrix, the Concept 1 was chosen as the best possible design. Soon after the team received feedback on the design, a novel best-bits-of-all design idea was formed as seen in the next section.

Final Design – Motivation of Preliminary Design

Upon receiving feedback through a design review presentation, design refinement was encouraged. While Concept 1 was chosen as the best alternative through the Pugh's matrix, the team decided that it would rather employ a clever combination of the different concepts.

Concept 1 used glycerin as the operating fluid of the device, where it would produce the positive and negative pressures required for the device. Air was not chosen as the fluid because of compressibility effects. However, it was found that compressibility effects of air are negligible for the temperature and pressure conditions that it would experience. Air may be treated as incompressible for a given Mach number less than 0.3 [7]. This was a major change toward a simpler design that did not require dealing with a liquid as the operating fluid of the device. Additionally, this concept suggested using external wiring for powering the device. This wiring was later thought to be undesirable, as the client requested a device that could be operated without any external hindrances.

Concept 2 used a half-glycerin half-air system to exert the required pressures on the abdomen. Again, newly found knowledge that the air could be assumed as incompressible in the device made the presence of glycerin irrelevant.

Finally, Concept 3 used air as the operating fluid. It employed a thermocouple to account for the compressibility of air, but this was now found to be redundant as air was assumed to be incompressible in the device. Moreover, this concept utilized a vacuum pump to drive the pressures, and a belt to hold the contraption in place against the abdomen. The unintuitive handling and complexity of the design clearly showed the need to have a more intuitive and simple design.

For all these concepts, the method of applying the pressure was always powered – not manual. This was another important change for the design. The design process was steered towards a simpler system where the pressure could be induced manually, and thus the design itself became lighter and more manageable.

Table 4: Morphological Chart for the final design

Aspect	Preliminary Final Design
Skin Contact	Medical adhesive tape
Pressure	Manual inputs
Displacement	Optical displacement sensor
Power	Internal rechargeable battery
DAQ	Raspberry Pi
Results Display	On separate computer
Connectivity	Internet
Housing	Custom 3D print
Pressure measurement	Transducer
Safety features	Filleted smooth housing body, no external wiring
Data processing	On device

To summarize, the three biggest changes to the design concepts were:

1. Understanding that compressibility effects for air were negligible, for the temperature and pressure it would experience within the device.
2. Realizing that a device that was not wired (for power) was better for operation purposes. While this may add weight to the device, the operability of the device was of paramount importance.
3. Moving away from a powered method of applying pressure, and hence an associated system of instruments, toward a manual input to apply pressure. This greatly decreased device complexity and shifted the design towards being more operable and ergonomic.

Many of the other design features that were in Concept 1 (best alternative according to the Pugh's matrix) remained in the final preliminary design. For example:

1. Using an optical displacement sensor to measure displacements with the required 1mm resolution.
2. Using a Raspberry Pi (RBP) as the microprocessor, and a method to view results. The RBP's ability to communicate via the internet, as well as collecting data from multiple sensors and processing data was ideal for the device.
3. Using a separate computer to view the results because of unnecessary added power requirements of an OSD (On-Screen Display).
4. Using a 3D printed housing body, which was the most versatile option. The body would be lightweight, and strong.
5. Using a conventional pressure transducer to provide the desired 1mmHg resolution.

For the manual inputs to induce pressure, a manual piston was considered. Additionally, the client requested a system where the piston can be "locked" into a specific position. Various locking systems were considered, such as a pin within the piston that can be turned into small holes along a cylinder to lock the piston in place. However, a discrete system was deemed undesirable due to the potential difficulty in achieving the requirement of pressure tuning within 5 mmHg. A mechanism that can continuously lock the piston in place was much preferred. Inspiration was taken from a caulk gun mechanism, in which a spring pushes a small plate at an angle to lock the rod in place using friction.

Figure 4: Caulk-gun mechanism

However, the client requested that the device be operable with a single hand. Therefore, it was preferable to examine using a bi-stable mechanism. The solution found was similar to a conventional electric motor linear actuator. A lead screw connected to a turning knob would move a nut up and down. Attached to this nut is the syringe's plunger. The device and mechanism is illustrated below.

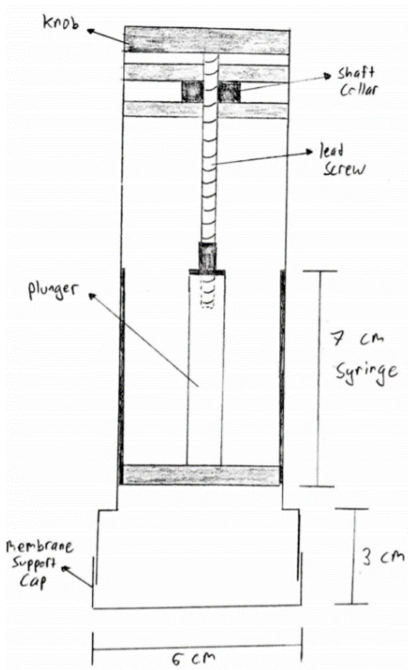


Figure 5: Lead screw and Knob Mechanism

Final Design Concept Selection

The final concept chosen very closely resembled Figure 5 discussed in the previous section. In this figure, the housing and rotating knob are both transparent to view the device components placed inside. Several design constraints were taken from the required pressure inducement ranges (-150mm Hg to +150mm Hg). From these constraints, the syringe dimensions were chosen. The length of the entire device was then chosen in a way to make the device ergonomic. The housing was placed around the components, and provided the necessary mounts required for the different components. The individual sub-systems will not be discussed in detail in this section, because the next sections are dedicated to explaining them individually. The CAD rendering for the device is seen in Figure 6.

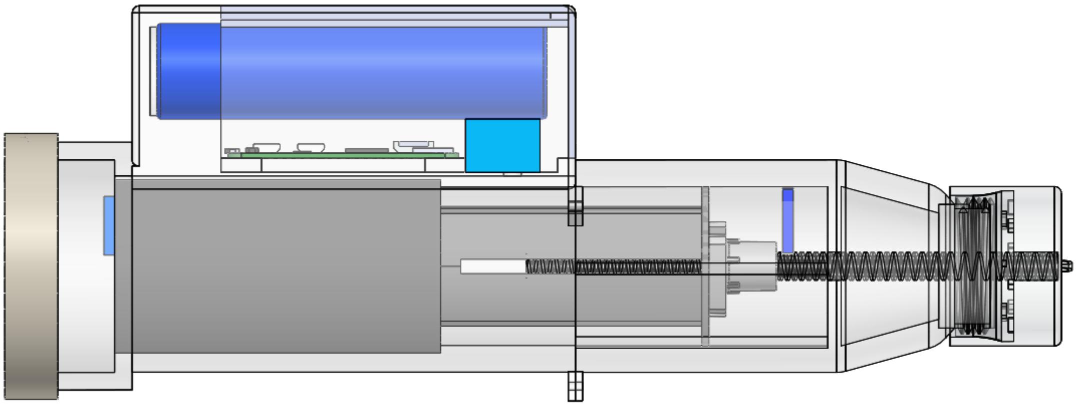


Figure 6: CAD of Final Design

A design drawing, isometric view and, bill of materials is also provided in the Appendix to detail each of the components present. The entire device is around 30 cm in length, and the maximum diameter of the cylindrical-like housing is 7 cm (the larger cross-section on the left).

Design Embodiment

Syringe, Plunger and Mechanism

The initial design called for the use of a 200 mL syringe. However, due to time constraints, the team had to forgo this option in favor of a 150 mL syringe as it had a more favorable shipping time. The syringe is approximately 15 cm long, as opposed to the 13 cm of the 200 mL syringe, and as a result the overall design of the device had to be modified to accommodate for this additional length. The inner diameter of the syringe is 4 cm. As the CAD design in Figure 4 shows, the syringe was to form the core of the device. Thus, a 4 cm diameter means that the device diameter will be approximately 6 cm. This is within the 50 – 65 mm maximum diameter requirement for the distance between the thumb and knuckle joint for a device to be easily handheld [8].

Between the bottom of the syringe and the membrane is a small air chamber. This chamber has a 6 cm diameter base and a 3 cm height, meaning it has an 84.8 cm^3 volume. From previous iterative calculations it is known that only approximately 100 mL of the 150 mL syringe will be required. The syringe can be cut to the required dimension – it is estimated that the maximum length of the device should be approximately 25 to 30 cm [9]. Thus, 9 cm was decided to be an ideal height for the syringe, since the plunger will need to extend 9 cm as well, meaning that the minimum device height is 18 cm. The remaining available height is reserved for the turning knob, air chamber, and other components. Thus, 100 mL of a 150 mL and 15 cm long syringe is rounded up to 8 cm, which gives 108 mL. Next, the plunger is set to its zeroed position by putting it at the 40 mL increment on the syringe. Knowing that the initial volume of air in the device is 84.8 cm^3 + 40 cm^3 = 124.8 cm^3, and the final volume after the syringe is pushed down to 0 mL is 84.8 cm^3, we have:

$$P_2 = \frac{P_1 V_1}{V_2} = \frac{101.3 \text{ kPa} \times 124.8 \text{ cm}^2}{84.8 \text{ cm}^2} = 149.0 \text{ kPa}$$

The same is repeated for the negative pressurization, for a final volume of 84.8 cm^3 + 108 cm^3 = 192.8 cm^3:

$$P_2 = \frac{P_1V_1}{V_2} = \frac{101.3 \text{ kPa} \times 124.8 \text{ cm}^2}{192.8 \text{ cm}^2} = 65.6 \text{ kPa}$$

The two pressures calculated are noticeably greater and lower than 101.3 kPa ± 20 kPa. This is because it is anticipated the membrane will deform as it is subjected to positive and negative pressures and deforms the abdominal cavity. The deformation of the membrane depends on the membrane's properties, the adhesive between the membrane and the skin, and the skin. Testing will have to be conducted on the prototype to quantify the membrane's deformation. It is currently assumed that the membrane's deformation will be marginally small, thus it will be assumed that the membrane's deformation is negligible. Thus, current calculations are completed using negligible deformation and a tolerance margin (TM) that will encompass the membrane deformation:

$$TM_{\text{positive pressure}} = \frac{149.0 \text{ kPa}}{101.3 \text{ kPa} + 20 \text{ kPa}} = 1.23$$

$$TM_{\text{negative pressure}} = \frac{101.3 \text{ kPa} - 20 \text{ kPa}}{65.6 \text{ kPa}} = 1.24$$

Thus, the 40 mL "zero" point on the syringe was chosen to ensure that both positive and negative pressurization have a maximized tolerance margin. After conducting tests, the team was to screw a lead screw nut onto the plunger to allow it to traverse up and down the lead screw. With testing, the team would have been able to determine the volume of the syringe required for the device, and thus machined both the syringe and the lead screw and the syringe to their respective final dimensions with the assistance of workshop technicians. Epoxy is to be used to secure the syringe in place, and to create an airtight seal.

Membrane and Adhesion

A glue lined membrane is to be used to adhere the device to the skin. The adhesion is to be strong enough to allow the membrane to deform both into, and out of the device, while remaining adhered to the skin. The team went with a transparent adhesive with a polyurethane backing. This is a strong plastic and would be able to undergo the deformations that are necessary for the device to work. A simple calculation based on the strain the membrane would experience helps define materials that can be used for the application. Using Hooke's Law and assuming a two-dimensional case, we get the following relation:

$$\sigma = E\,\varepsilon$$

Where the strain, ε can be found by assuming a deflection in the material. The Young's modulus, E, is a material property and thus determined by the material choice, while the stress σ must fall within the elastic region of the material.

As an example, were the maximum deflection in the material to be 0.5 cm from the initial position at the center, the strain can be determined as follows.

Maximum stretching of the membrane occurs when the overall deflection is circular in form. In this case, the chord length would be 6.11 cm.

$$\varepsilon = \frac{\Delta L}{L} = \frac{0.11}{6} \approx 0.0183$$

From which the stress can be determined. Taking the average material properties of polyester [10] as an example, we find the following.

$$\sigma = E\,\varepsilon = 4\ \mathrm{GPa} * 0.0183 = 73.2\ \mathrm{MPa}$$

The resultant stress falls within the average flexure yield strength of polyester (95.1 MPa), meaning that the membrane would not undergo plastic deformation during the operation of the device. In this manner, it is possible to predict whether a potential membrane is suitable for the application or not. The 0.5 cm deflection is something that was assumed, not accounting for any resistance the skin itself may provide. Further testing would allow for more defined models to be created where this methodology can be used to predict the behavior of the membrane.

A preliminary application and material test of the membrane showed that it adhered easily to the skin and was very flexible and pliable. It could be stretched easily without much loss in its adhesion. Removal of the membrane was also simple and was not a painful process.

There are some concerns regarding the airtightness of the main chamber, however the use of epoxy or a thermal set plastic film will create an airtight seal. An O-ring may be placed on the membrane cap should there be any leakage of air in the area.

Housing and Turning Knob

The housing is the structure that houses all the components in a safe and reliable manner. As described previously, the main design constraints stemmed from the selection of an appropriate plunger and syringe. These mechanical components are unwieldy and make it hard for an operator (either patient, or physician) to take measurements. To streamline the design, and to reduce arbitrary operation of device (incorrect hand placement over components), the housing was deemed essential. The device also consists of electrical components, and the associated necessary wiring. Not only do these impede an operator's ability to interact with the device, but they also impose direct safety hazards. A shorted circuit may pose danger. The housing also provides the required safety, shielding these components away from the operator. The housing is pictured in Figure 7.

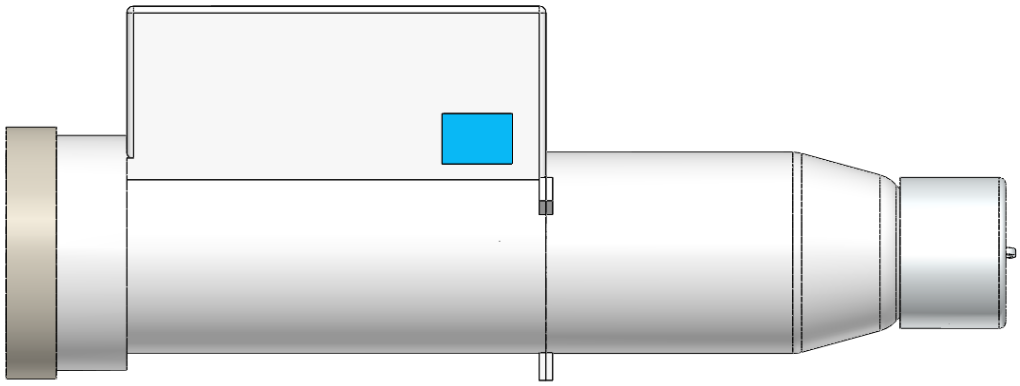


Figure 7: CAD of Final Design Housing

The left section is where the membrane will be attached and contact with skin will be made. The section of increased cross-sectional area on the left is where the pressure transducer will be housed, and where the required air pressure will be created. The long cylindrical smaller cross-sectional area section is where the syringe and plunger operate. Attached below this section is a compartment for housing most of the electronics, like the RBP. At the right end, the rotating knob will be affixed.

The housing was chosen to be as ergonomic as possible. The tapering end of the device is meant to resemble features commonly seen in water bottles, further attesting to the ease with which the device can be used. It is around the tapered end where the operator will hold the device. At the very end of the tapering, large threads (with few revolutions) are placed. The threads were made relatively large because 3D printing finer threads through several revolutions is a challenge. If these large threads cannot be 3D printed, they can be machined as standard thread sizes were chosen. These threads are meant to screw into the rotating/turning knob.

The housing was 3D printed in six different parts to overcome the limitations of printing certain shapes that the design incorporated. These included the membrane cap, membrane section, electronic housing, syringe housing, tapered section and the knob. The membrane section and the electronic section were glued together to form one part, while the syringe housing and the tapered section were glued to form a second part. These would be fixed together with the use of external screw ledges. The knob was to be assembled and could simply screw onto the tapered end. Similarly, the membrane cap was also to screw onto the tapered end.

Some machining and processing had to be conducted on the printed part to account for support structures, printing errors and unforeseen limitations. Screw holes were drilled for the external screw mounts and for the board standoffs in the electronic housing. Sanding work was completed to fit the glued parts together and allow for the smooth operation of the sliding door. It came to the team's attention that the power cables were not pliable enough for the space made in the electronic housing, so a rectangular hole was dremeled into the side to allow space for the cabling. A misprint in the syringe housing section meant that a channel had to be chiseled to allow room for the installation of the pressure sensor, and to run its wiring.

Due to the importance of an airtight seal for the pump to properly function, along with a high possibility of imperfect printing and cutting, a proper test is to be done on the whole device. Using a leak detector such as Snoop® fluid, the application of liquid soap throughout the system would create bubbles for easy identification of leaks. These would then easily be fixed through the application of epoxy glue and then tested once again for leaks in the same manner.

As seen in the report of ergonomics in industrial use [8], the device can be quantitatively classified as portable, as it can be carried for at least ten minutes without resting. The device itself weighs around 560 g, and so can be carried with a single hand comfortably.

Table 5: Weight Table

Part	Material	Weight (g)
Housing	PLA	317.60
Plunger w/ Syringe	Silicone Rubber and Polypropylene Plastic	90
Lead Screw	4140 Alloy Steel	50
Lead Nut	Acetal Plastic	4
Rotating knob	PLA/Standard Plastic	1.84
Mason Jar Lid	PLA	40
Membrane	Biomedical Adhesive Tape	~ 0
Raspberry Pi	Various	9
Battery	Various	39.50
Electronics (Sensors)	Various	2.60
	Total	**559.54**

The turning knob, seen in Figure 8, will be around 3cm in height for adequate room for the thumb and forefinger to rotate the knob. The knob will have the threaded rod embedded into it. Rotation of the knob rotates the rod, which results in axial movement of the nut, and the subsequent displacement of the plunger into the syringe. Thus, a design had to be chosen, where despite all rotations of the knob (clockwise or counterclockwise), the knob would not unscrew from the threads. A child proof medicine cap was used an inspiration for this purpose. Thus, the knob consists of an outer casing, and an inner threaded casing. It is in this outer casing that the threaded rod is embedded. It can only be unscrewed from the thread on the housing by pressing onto the knob firmly towards the device, just like a child proof medicine cap.

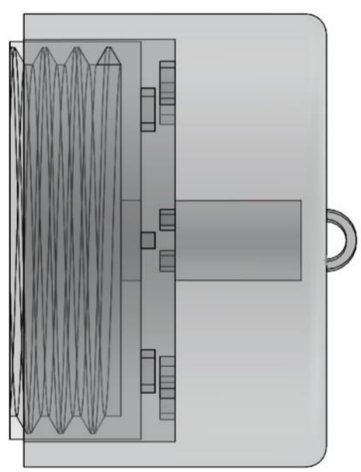


Figure 8: CAD of Turning Knob

The infill of the print was set to 40%, with a wall thickness of 1 mm. In theory, the device would still be strong enough with a smaller infill percentage, but to ensure that housing does not fail in assembly and testing, and to ensure a strong prototype design, the infill percentage was set to be slightly higher at 40%. This did increase the time of the print and the material cost, but it mitigated the need of a reprint in the event of a structural failure. Although Figure 9 below shows a single view of the assembled prototype, Appendix G – Photographs of 3D Printed Casing is a whole section showing many different angles of the sub-components of the device, and gives a clearer view of how the device functions together.

Figure *9*: Manufactured Prototype Housing with Lead Screw

Electronics and Software

An RBP Zero W was selected as the microprocessor for this design. This board was selected over a microcontroller such as an Arduino for its capabilities to function as a computer, specifically its ability to connect to Wi-Fi and its ability to store data for future viewing. The RBP Zero W was selected over other RBP models due to its small size. Other RBP models have USB or ethernet ports, which are not required for this project and would take up space for nothing. The Pi Zero W has a 1 GHz, single-core CPU and 512 MB of RAM, which is sufficient for the purposes of this project. The Pi Zero has 40 general purpose input/output (GPIO) pins. Some of these can be used to control and collect data from sensors, which can be directly powered by the Pi Zero's 5V or 3.3V pins to provide the necessary power requirements to these sensors. The RBP is normally powered by 1A, 5 V power supply. The RBP requires an SD card for storage, which will need to be purchased separately.

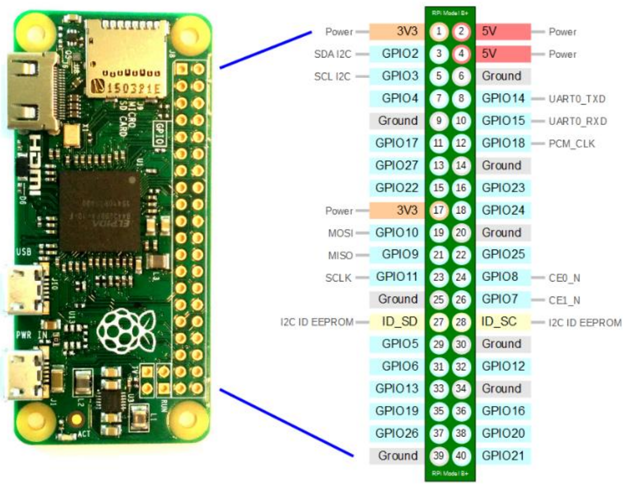


Figure 10: Raspberry Pi Zero W and Pins [10]

Table 6: Raspberry Pi Specifications [11] [12]

Specification	**Value**
Voltage	5 V
Current	150 mA
Cost (RBP + Power supply + SD card)	$47.87
Dimensions	65 mm x 30 mm x 5 mm
Weight	9 g

The current design calls for a wireless device. Thus, the RBP requires a battery. Given that the RBP is powered by a standard micro-USB port, it requires a 5V power source. Thus, any portable phone battery pack is capable of powering the RBP. The Poweradd EnergyCell battery was chosen for its good price and small size. Its side and weight was further reduced by removing the plastic casing around the battery. The battery was tested on the RBP with the sensors connected and it was able to properly power the system for several hours.

Table 7: Poweradd EnergyCell Battery Specifications *[13]*

Specification	**Value**
Voltage	5 V
Current	2.4 A
Cost	$19.98
Dimensions	110 mm x 27 mm x 27 mm
Weight	97.2 g

Figure *11*: Poweradd Energy Cell battery removed from its product casing

Next, an optical displacement sensor is required to measure the distance travelled by the piston in order to calculate the volumetric displacement. The Adafruit VL6180X Time of Flight Distance Ranging Sensor is selected for this purpose. This sensor can precisely measure the distance between itself and an object between 5 mm and 100 mm away, with a resolution of 1mm. This sensor is also ideal because Adafruit provides code libraries which can be used. Therefore, it should not be necessary to write code that interacts with the basic hardware, which greatly simplifies the process. This sensor is digital and uses i2C communication, meaning that it can directly interact with the RBP, with no need for an analog to digital converter.

Table 8: Adafruit VL6180X Specifications *[14] [15]*

Specification	**Value**
Voltage	2.8 V
Current	5 mA
Cost	$13.95
Dimensions	20.5 mm x 18 mm x 3 mm
Weight	1.4 g
Resolution	1 mm
Range	0.5 - 100 mm

Next, the pressure sensor used is the Adafruit BMP388 - Precision Barometric Pressure and Altimeter. The BMP388 has an operational range between 300 and 1250 hPa absolute pressure (225 to 938 mmHg) which puts it within the project requirement range of +/- 150 mmHg gage pressure. Additionally, it has an accuracy of +/- 0.4 hPa, or 0.3 mmHg, which means it also conforms to the project requirements. This sensor is digital and uses i2C or SPI communication, meaning that it can directly interact with the RBP. Adafruit also provides code libraries for interaction with this sensor.

Table 9: Adafruit BMP388 Specifications *[16] [17]*

Specification	**Value**
Voltage	3.3 V
Current	2.7 µA
Cost	$9.95
Dimensions	21.6 mm x 16.6 mm x 3 mm
Weight	1.2 g
Accuracy	0.4 hPa
Range	300 - 1250 hPa

The following schematic shows the circuit used to connect the sensors to the Pi Zero W:

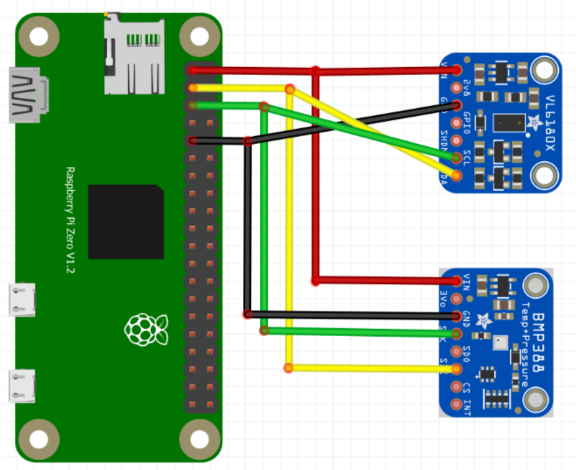


Figure 12: Circuit Schematic

In this circuit, the two sensors are connected to the RBP. It was decided to operate both sensors on the i2C communication channel as it simplifies the circuit and i2C is widely used and reliable.

For the distance sensor, the following pins are connected:

- Pi Zero W 3V3 to sensor VIN
- Pi Zero W GND to sensor GND
- Pi Zero W SCL to sensor SCL
- Pi Zero W SDA to sensor SDA

For the pressure sensor, the following pins are connected:

- Pi Zero W 3V3 to sensor VIN
- Pi Zero W GND to sensor GND
- Pi Zero W SCL to sensor SCI
- Pi Zero W SDA to sensor SDK

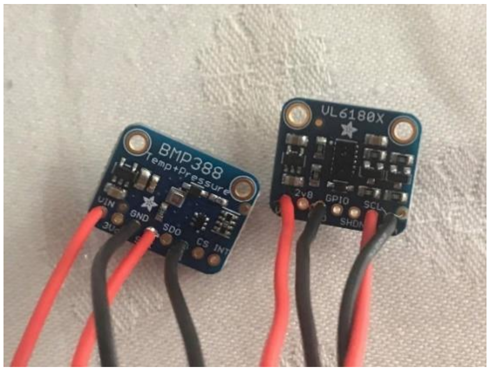


Figure *13*: Photograph of both sensors with connections soldered

Next, in order to fulfill the requirement of displaying pressure-volume curves in real-time as the measurement is being taken, a system was devised to enable the viewing of real-time curves on a separate computer. However, while it would be possible to create an application for a Windows computer, it would not be compatible on a Mac. To avoid this problem, an internet-based solution taking advantage of the RBP's Wi-Fi capabilities will be created, with the output being in displayed in a web browser.

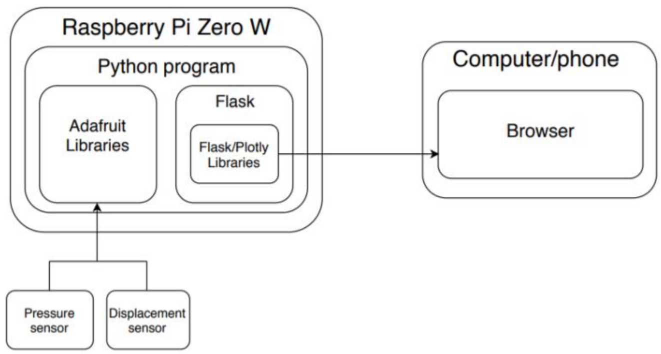


Figure 14: Software Schematic

The diagram above illustrates the principle of operation of the software. When researching for ways to display live graphs in a browser, Plotly was found to be a good solution [18]. Plotly, specifically its Dash web framework, allows the creation of interactive, real-time plots in a browser [19]. It uses Python, a commonly used programming language, which is the same language RBP programs are typically written in.

The RBP will also collect data from the sensors. This will be done using the libraries that Adafruit have created to allow users to easily interact with their sensors. A python program collects this data, and writes it into the RBP's filesystem, to allow for future viewing after the test. While performing a test, the program will send this data to the Flask web server, for creation of live plots using Plotly and Dash.

Due to the exceptional circumstances taking place at the time of writing, the electronics were not fully integrated into the physical prototype. However, the electronics and software were completed and tested outside of the prototype. The code can be found in the Appendix E. The set-up in Figure 16: Software and electronics testing set-up used to thoroughly test the software, sensors and electronics as much as possible outside of the prototype.

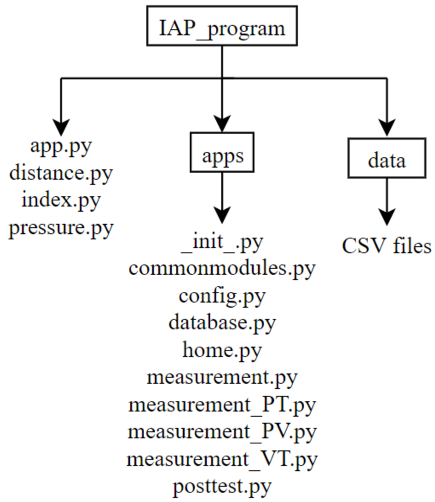


Figure *15*: Code module map

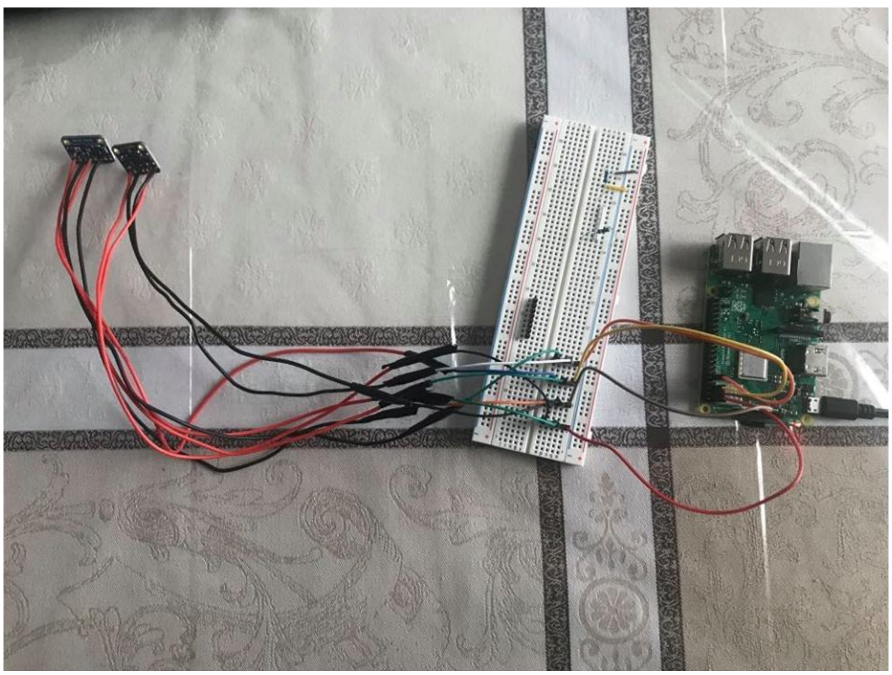


Figure *16*: Software and electronics testing set-up

The RBP used in Figure 16: Software and electronics testing set-up is a slightly different model which is powered by a wall source. However, all models are fully interoperable, and the set-up had previously been tested using the battery and it was fully functional. The wires were soldered to the sensors to facilitate testing.

The program is accessed by typing the IP address of the device connected to the local network. This IP can be found by pinging "raspberrypi.local" on the local network. Alternatively, for use in an area with many routers such as a school or hospital, it is better to have the RBP emit its own hotspot that can be connected to and in which the IP address remains constant. This procedure is relatively simple and has been tested to work [20]. The RBP is configured to start the IAP measurement program on startup, therefore the device needs only to be powered on and the user can then access it from any browser [21]. The program includes three different pages: the homepage, IAP Measurement page and Database page. The Home page displays information about the device and how to use it. The IAP Measurement page enables the user to take an IAP measurement and view live-plots, and the Database page allows the user to retrieve previous measurements as CSV files or delete these measurements. Screenshots of the full functionality of the program can be found in Appendix F.

Next, the performance of the sensors is demonstrated.

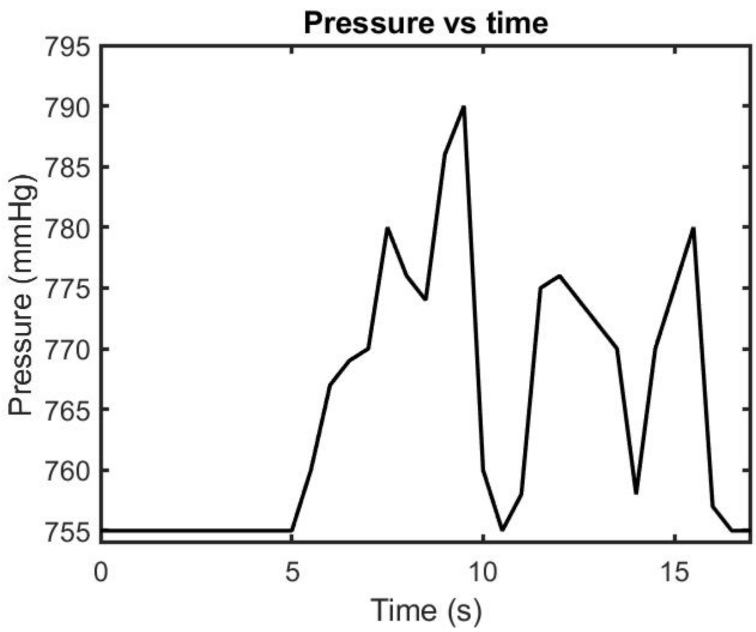


Figure *17*: Pressure sensor test plot

In this test, the pressure sensor was placed into an inflated plastic bag. When pressure was applied on the bag, the pressure change was reflected. For the first 10 seconds, the pressure sensor is registering the ambient atmospheric pressure. Then, pressure was applied to the bag in varying amounts. This test also shows that the pressure sensor is capable of measuring +25 mmHg above ambient atmospheric pressure.

Figure *18*: Pressure sensor in plastic bag

Next, the distance sensor was tested. This test was done by holding something in front of the sensor and gently moving it back. This was verified to be the correct distance using a ruler.

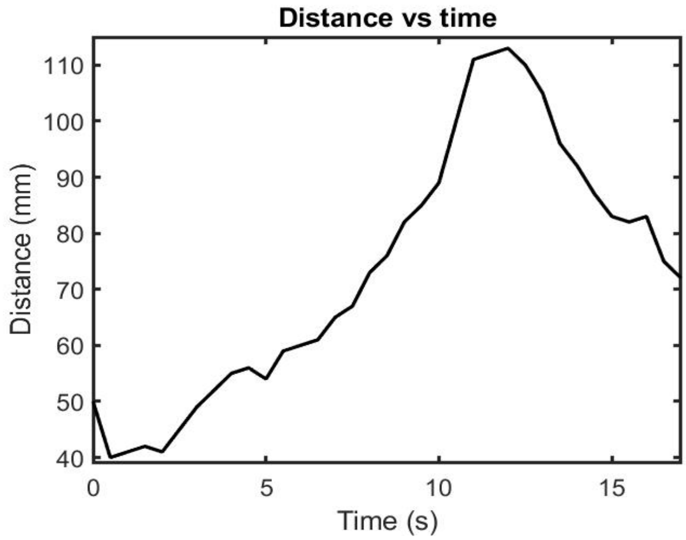


Figure *19*: Distance sensor test plot

Prototype Operations

Once the prototype is fully assembled, with all the machining done, and testing and calibration finished, an order of operations can be established. This is a list of the expected operations for use in a medical space.

1. Cut out a piece of medical adhesive tape from the roll approximately 10 cm long, such that the piece looks like a square.
2. Place the piece of tape on the bottom of the device and screw on the mason jar lid. Make sure the piece is not loose.
3. Mount the rotating knob on the top and close it. Rotate the knob till the plunger reaches a marked area to denote that the knob should be removed.
4. Remove the knob by pressing downward on it, and then twisting it. The device should now have an internal pressure equal to atmospheric pressure.
5. Mount the knob back, and ensure the plunger is at the appropriate marking.
6. Rest the patient on his/her back on a bed and expose the abdominal area right below the sternum. Apply the device on the test section and make sure the medical adhesive tape is stuck firmly onto the body.
7. Loop the safety cord on top of the rotating knob around the operator’s wrist.
8. Place the thumb and forefingers on the rotating knob. The palm should rest comfortably on the tapering part of the device.
9. The operator can now rotate the knob counterclockwise to move the plunger down or rotate it clockwise to move the plunger up. Rotations should be made slowly, with thumb and forefingers rotating it approximately every 1-2 seconds.
10. Keep rotating till the plunger reaches either the lower marking (if the plunger is moving down), or the upper marking (if the plunger is moving up).

11. Remove the device, remove the medical tape, and set aside the device. All the results should be obtained remotely on a nearby computer.

Prototype Predicted Performance

Due to unforeseen events that led to the shutdown of university facilities, the full construction of the prototype was not possible to finish. The device with all of its components could not be fully assembled and tested. Hence, no final measured values were obtained. The best way to lend constructive value to future work done in this project is to predict how exactly the device would perform.

The flowchart below has the steps to convert the final expected measurements to the desired values of IAP. Pressure-volume curves were expected by the client, and so the flowchart also includes that as a necessary output.

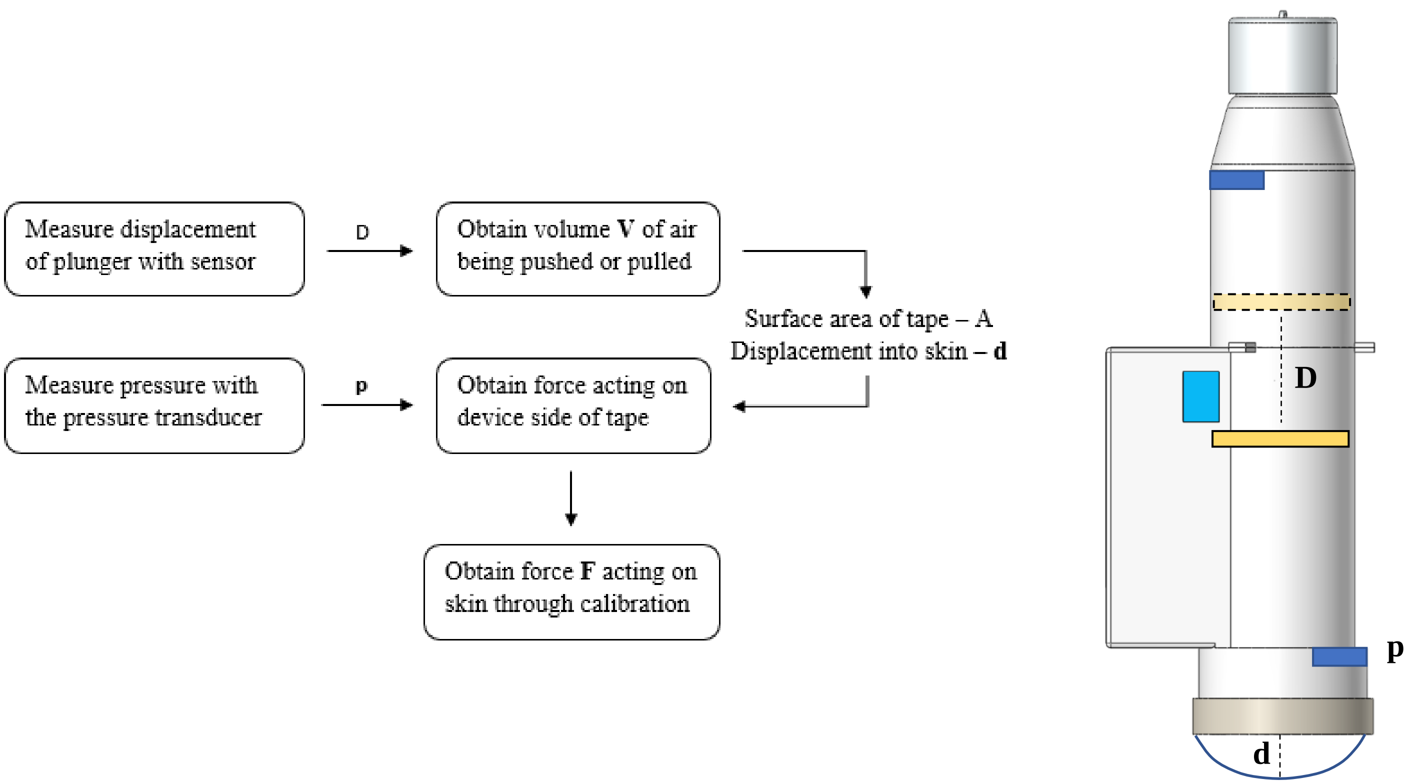


Figure 20: Flowchart with measured variables, and outputs

As seen in the above flowchart, from the displacement sensor (measuring D) and from the pressure transducer (measuring p), the force acting on the skin $\mathbf{F_{skin}}$ and the displacement into the skin **d** can be obtained. Hence, $\mathbf{F_{skin}/d}$ can be found, and the results of the aforementioned study [4] can then be used to estimate the IAP. Additionally, pressure-volume curves can be obtained with both the pressure parameter **p** and the volume of air **V**.

While no current measurements are taken, a hypothetical run-through of the flowchart would be useful in predicting what sort of results could be obtained. After setting up the device as explained in the Prototype Operations section, the plunger is at a position with zero-gauge pressure (compared to atmospheric pressure) applied. Also, the distance sensor is zeroed. Assume that positive pressure was first applied. The plunger moves downward, pushing air downward through a distance **D** into the chamber at the bottom. Assume D = 3cm. The volume being pushed is,

$$\text{Volume V} = \pi * \text{D} * \frac{(\text{Diameter of plunger})^2}{4}$$

For the fixed diameter of plunger as 4cm,

$$\text{Volume V} = \pi * 3\text{cm} * \frac{(4\text{cm})^2}{4} = 37.7\ \text{cm}^3$$

Meanwhile, a pressure increase is expected (relative to the atmospheric pressure, 760 mm Hg, initially). Say the transducer measures an absolute pressure of **p** = 771 mmHg. The distance (and associated volume) measured, and the pressure measured are both instantaneous. Thus, real-time pressure-volume curves can be obtained.

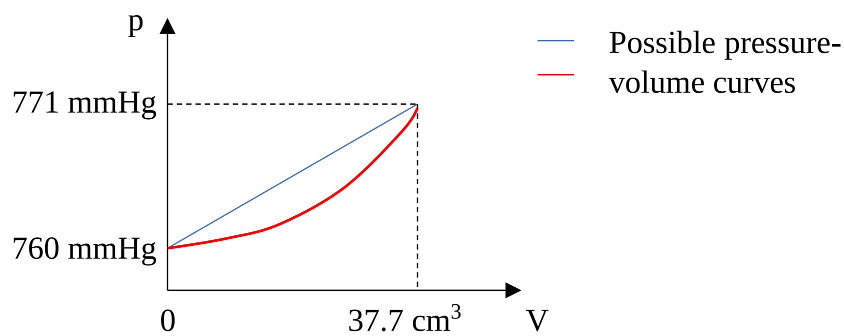


Figure 21: Possible pressure-volume curves

The medical adhesive tape will be pushed into the skin. Recall the tape was chosen such that the mechanical properties of this tape allow the expected amount of deformation without yielding and tearing. The tape will not remain elongated after the pressure has been removed. The tape deformation can be modelled to be part of the sphere (blue region). The volume V pumped into the chamber must go into deforming this tape.

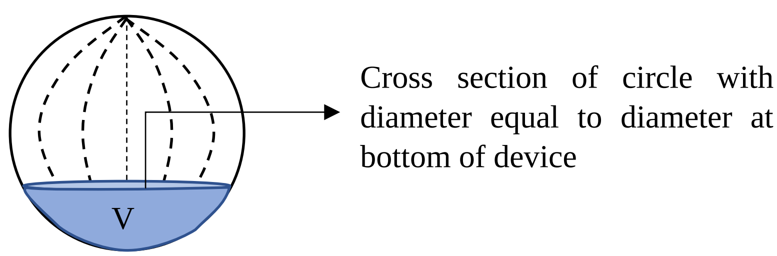


Figure 22: Model to find out how medical adhesive tape deforms

The displacement of the tape into the skin **d** is expected to be small – meaning the chord on this sphere is positioned lower. The larger the angle α is, the lower the chord is. Assume $\alpha = 75°$

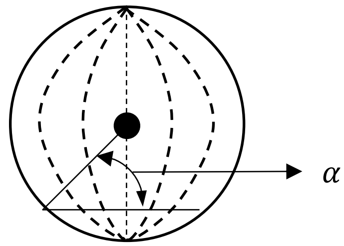


Figure 23: Alpha angle for chord determination

With the angle fixed, and the chord length known (equal to the diameter at bottom of device), simple trigonometry can be used to determine the radius, r, of the sphere.

$$r = \frac{\text{diameter at bottom of device}}{2} * \frac{1}{\cos\alpha}$$

With this diameter = 7 cm, and α = 75°,

$$r = \frac{7\text{cm}}{2} * \frac{1}{\cos 75°} = 13.52 \text{ cm}$$

Using the other trigonometric relation, the displacement into the tape **d** can be found as,

$$x = r\sin\alpha = 13.52\text{cm} * \sin 75° = 13.06 \text{ cm}$$

$$d = r - x = 13.52 - 13.06 = 0.46 \text{ cm}$$

The area of the tape on the device side can be found with the radius and this displacement, by simply looking at the formula of a spherical cap. This area is,

$$A = 39.08 \text{ cm}^2$$

Now, the force acting on the tape on the device side is found as,

$$F = p * A = 11 \text{ mmHg} * 39.08 \text{ cm}^2 = 5.73\text{N}$$

The force acting on the skin can then be obtained through calibration, with a factor applied.

$$F_{skin} = \text{Calibration Factor} * F$$

This calibration factor will be obtained through testing. For now, assume the calibration factor is found to be 1, that is, all the force applied on the tape ends up applying on the skin. This is a reasonable assumption to make. Hence,

$$F_{skin} = 5.73\text{N}$$

The abdominal wall tension is then,

$$\text{Tension} = \frac{F_{skin}}{d} = \frac{5.73\text{N}}{0.46\text{cm}} = 1.25 \frac{\text{N}}{\text{mm}}$$

This is the value that can be correlated to IAP according to [4]. For example, consider all these measurements were performed on a female subject at the Linea Alba Cranial test section right below the sternum, as seen in [4].

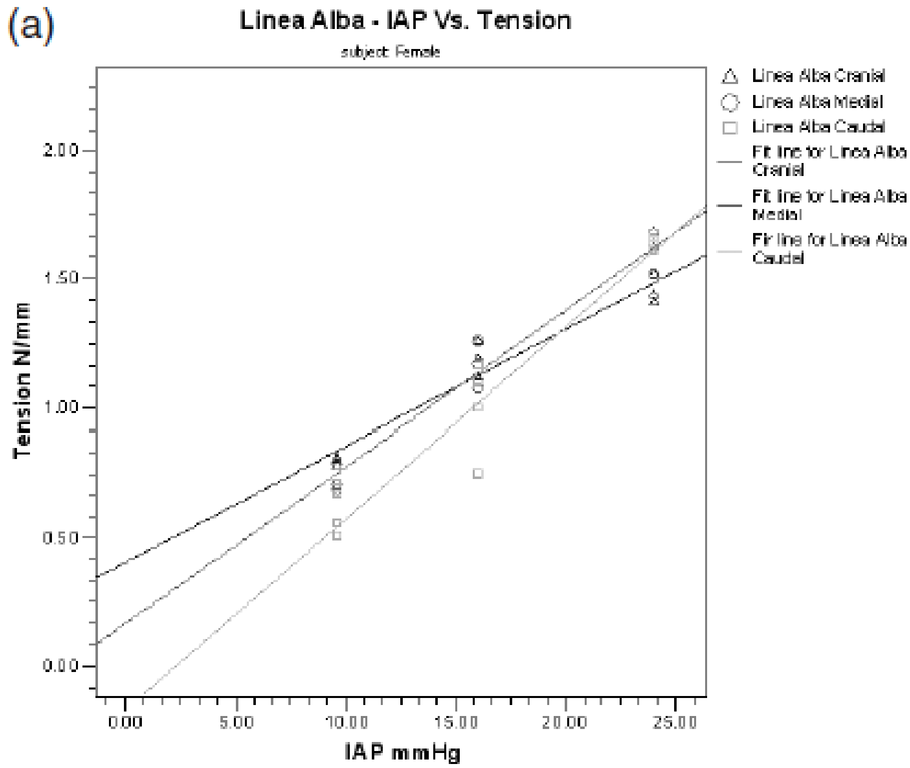


Figure 24: Linea Alba test section - IAP correlation

The best line approximation fit has the equation

$$\text{Tension} = \text{slope} * \text{IAP} + \text{Tension}\ (@\text{IAP} = 0\text{mmHg})$$

$$\text{Tension} = \ 0.04333 * \text{IAP} + 0.4167$$

Rearranging to solve for IAP,

$$\text{IAP} = \frac{\text{Tension} - 0.4167}{0.04333}$$

With the tension found to be 1.25 N/mm, the IAP (in mmHg) is,

$$\text{IAP} = \frac{1.25 - 0.4167}{0.04333} = \mathbf{19.23mmHg}$$

This IAP measurement is quite high. Around 20mmHg of IAP indicates Abdominal Compartment Syndrome (ACS) with evidence of organ dysfunction. However, this final value was obtained from reasonable, yet completely hypothetical measurements of 11mmHg of pressure (measured by transducer), and 3cm of plunger displacement (measured by optical sensor). The flow of logic in the above equations is of real value. Also, having this IAP calculated value in the same order of magnitude as normal IAP values certainly reaffirms the belief that the device can operate successfully.

Conclusion

The device built demonstrates that measuring IAP through a non-invasive, simple, and cost-effective manner is fully possible through the measurement of skin displacement induced by pressure on the abdomen. As all subsystems of the prototype have been built and tested separately, the data required to measure IAP is obtainable and can be compared to previous research done in the correlation of displacement and IAP [3]. Along with all the electronic systems being shown to work, the handheld pump was also tested for one-handed use and was shown to be ergonomic and simple to use. Using the lead screw pumping mechanism, the requirements for measurement resolution of 1 mm of displacement and 1 mmHg of pressure were met. All of this was furthermore compartmentalized within a single 3D printed housing that is portable and easily changed for future iterations.

This proves to be important for the future of the project as meeting the client demand for a prototype to cost less than $100 per unit proved to be difficult and was finally made of lower importance in order to ensure that the measurement requirements and handheld specifications were met. As seen in Appendix C – Bill of Materials, the estimated price was $391.55, with the majority of the spending on the housing, lead screw, and lead screw nut. The actual price (also in Appendix C – Bill of Materials) of all the shipped and manufactured components came to $383.28, with over a third of the cost ($142.73) coming from 3D printing the device housing. As the housing and parts were all acquired as single units, the price for mass manufacturing in the long run would be greatly reduced. In the future, a smaller battery that was unable to be acquired at the time of the project could be utilized to reduce the housing size and price of the electronics subsystem. This, along with a much thinner main body would reduce the total volume of material required for the device and allow the budget demands to be met. If pieces were mass manufactured through molds, the 3D printing prices could be removed entirely, along with the external screws. Furthermore, the prototype budget was slightly inflated due to 3D printing errors, which would not happen during mass manufacturing.

Finally, due to the device not fully being built, a full-on calibration of the measurements was not able to be completed. This comes from the membrane material's effect on the displacement response along with the change in contact area compared to previous published works [3]. This would be easily overcome through multiple tests of known IAP patients, where linear graphs would be scaled by a magnitude to represent the proportional effects of the device on the measurements.

Through the design and fabrication of the prototype, the importance of being able to rapidly adapt to client demands and specifications allowed for the best possible outcome. By implementing common designs in the world and applying them to solve design problems, the device achieved the goals set out at the start of the process. With proper planning and group management, the device was tested and proven to work even with unforeseeable events preventing meetings further into the fabrication process.

References

[1] E. A. Harman, R. M. Rosenstein, P. N. Frykman and G. A. Nigro, "Effects of a Belt on intra-bdominal pressure during weight lifting," *Medicine and Science in Sports and Exercise,* 1989.

[2] R. Milanesi and R. C. A. Cargnato, "Intra-abdominal pressure: an integrative review," Einstain (Sao Paulo), Sao Paulo, 2016.

[3] G. H. v. Ramshorst, M. Salih, W. C. J. Hop, O. J. F. Waes, G.-J. Kleinrensink, R. H. M. Goossens and J. F. Lange, "Noninvasive Assessment of Intra-Abdominal Pressure by Measurement of Abdominal Wall Tension," *Journal of Surgical Science,* no. 171, pp. 240-244, 2011.

[4] G. v. Ramshorst, "Non-invasive measurement of intra-abdominal pressure: a preliminary study," *Physiological Measurement,* pp. 41-47, 2008.

[5] T. S. a. D. Association, "Glycerine: an overview," The Soap and Detergent Association, New York, 1990.

[6] Omega, "Pressure Transducers," Omega Engineering, 28 August 2018. [Online]. Available: https://www.omega.ca/en/resources/pressure-transducers. [Accessed 13 October 2019].

[7] J. Anderson, Fundamentals of Aerodynamics, McGraw-Hill, 2007.

[8] Processforusability, "Ergonomics of hand held devices for industrial use," [Online]. Available: http://www.processforusability.co.uk/PIC54webslides/Slides/Ergonomics_of_hand-held_devices.htm. [Accessed 03 12 2019].

[9] P. C. Limited, Process Contracting Limited , 12 December 2019. [Online]. Available: https://www.he-alert.org/filemanager/root/site_assets/standalone_article_pdfs_1220-/he01255.pdf. [Accessed September 2004].

[10] MatWeb, "Overview of material for Polyester (Thermoset) - Rigid," 2019. [Online]. Available: http://www.matweb.com/search/datasheet.aspx?matguid=1d92ed366503454ba49b8a44099f90de&n=1&ckck=1. [Accessed 08 12 2019].

[11] Canakit, "Amazon," Amazon, [Online]. Available: https://www.amazon.ca/CanaKit-Raspberry-Wireless-Official-Supply/dp/B071DVLBC1/ref=sr_1_3?crid=V39KJ77V3KVY&keywords=raspberry+pi+zero+w&qid=1574899622&sprefix=electronics+star%2Caps%2C167&sr=8-3. [Accessed 2019 November 11].

[12] R. P. Foundation, "Raspberry Pi," Raspberry Pi Foundation, [Online]. Available: https://www.raspberrypi.org/documentation/faqs/. [Accessed 27 November 2019].

[13] Poweradd, "Poweradd EnergyCell 5000mAh, Portable Charger Power Bank External Batteries, Power Bank with Smart Charge Compatible with Smartphone,iPhone, Samsung Galaxy and More - Black," Amazon, [Online]. Available: https://www.amazon.ca/Poweradd-EnergyCell-Batteries-Compatible-Smartphone/dp/B07MSDHF67/ref=sxbs_sxwds-stvp?cv_ct_cx=5v+battery+pack&keywords=5v+battery+pack&pd_rd_i=B07MSDHF67&pd_rd_r=6bfd28d3-71a1-4785-bcdd-cfbe021bc260&pd_rd_w=NurIu&pd_rd_wg=IbYwm&pf_r. [Accessed 1 April 2020].

[14] Adafruit, "Adafruit VL6180X Time of Flight Distance Ranging Sensor (VL6180)," Adafruit, [Online]. Available: https://www.adafruit.com/product/3316. [Accessed 27 November 2019].

[15] Pololu, "VL6180X Time-of-Flight Distance Sensor Carrier with Voltage Regulator," Pololu, [Online]. Available: https://www.pololu.com/product/2489. [Accessed 27 November 2019].

[16] Adafruit, "Adafruit BMP388 - Precision Barometric Pressure and Altimeter," Adafruit, [Online]. Available: https://www.adafruit.com/product/3966. [Accessed 27 November 2019].

[17] B. Sensortec, "BMP388 Barometric Pressure Sensors," Bosch Sensortec, [Online]. Available: https://www.bosch-sensortec.com/bst/products/all_products/bmp388. [Accessed 27 November 2019].

[18] P. Programming, "Live Graphs - Data Visualization GUIs with Dash and Python p.4," Pythonprogramming.net, [Online]. Available: https://pythonprogramming.net/live-graphs-data-visualization-application-dash-python-tutorial/. [Accessed 27 November 2019].

[19] cwalk, "Pi-Temp," Github, [Online]. Available: https://github.com/cwalk/Pi-Temp/. [Accessed 27 November 2019].

[20] "Raspberry Pi - Hotspot/Access Point dhcpcd method," Raspberry Connect, 30 June 2017. [Online]. Available: https://www.raspberryconnect.com/projects/65-raspberrypi-hotspot-accesspoints/168-raspberry-pi-hotspot-access-point-dhcpcd-method. [Accessed 1 April 2020].

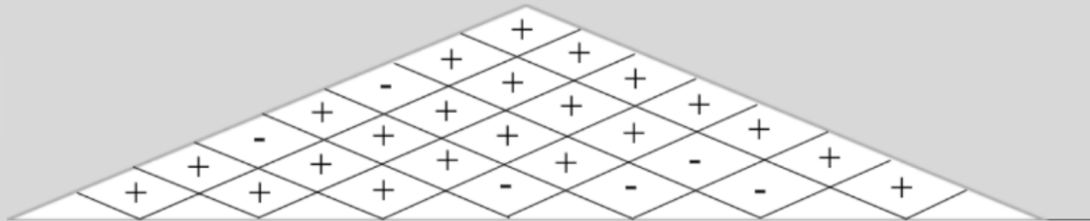
[21] S. Electronics, "How to Run a Raspberry Pi Program on Startup," Digikey Electronics, [Online]. Available: https://www.digikey.ca/en/maker/projects/how-to-run-a-raspberry-pi-program-on-startup/cc16cb41a3d447b8aaacf1da14368b13. [Accessed 01 April 2020].

Appendix A

QFD is a method to transform the client's design requirements into engineering requirements to achieve that design. The QFD allows for clear comparison of different design considerations and priorities from an engineering standpoint. Using the fourteen design requirements previously outlined, eight general functional requirements were established. These eight requirements aim to categorize all client requirements.

The client importance rating column on the left is a direct reflection of the client's design requirements ranking. The values in this column quantify the importance of each requirement and are chosen such that the sum of the column is 100. They are calculated as $\left(\frac{x}{105}\right) \times 100$, where x is the importance of that requirement (14 being the most important, 1 being the least important) and 105 being the sum from 1 to 14 (i.e. 1+2+3…+14 = 105). The turquoise column on the right lists these client requirements. The blue row on the top lists the functional requirements. The QFD is then filled, showing the importance of the relationship between each client requirement and functional requirement. The final results at the bottom of the table rank each engineering requirement by priority, taking into account the relative importance of each client requirement. The result obtained from the QFD is the most design requirements from most important to least important are: pressure inducer, microprocessor/data acquisition system (DAQ), pressure sensor, displacement sensor, cost, device size and weight, power consumption, and housing manufacturing and material.

Additionally, on the top of the QFD, the relationship between each of the functional requirements is shown by assigning a '+' to those that interrelate and a '–' to those that have no interdependence. From this, we can also see that the microprocessor, engineering cost, and pressure inducer each highly depend on all other functional requirements



	Client importance rating	Functional Requirements (How) → / Client Requirements - (What) ↓	Housing manufacturing and material	Pressure inducer	Engineering cost (BoM)	Power consumption	Device size and weight	Pressure sensor	Displacement sensor	Microprocessor, DAQ	Weighted Score
1	13.3	Safe design with no sharp corners	10	0	4	0	3	0	0	0	240.0
2	12.4	Restricted motion and reliable seal on skin	10	7	4	0	5	0	4	0	383.8
3	11.4	Displacement resolution of 1 mm	1	5	7	3	4	3	10	7	395.6
4	10.5	Pressure resolution of 1mmHg	1	10	7	5	6	10	3	7	457.5
5	9.5	Pressures of +/-150 mmHg	1	10	7	5	6	9	5	7	426.0
6	8.6	Pressure tuning to 5 mmHg	0	10	6	5	6	9	5	7	367.0
7	7.6	Real time pressure vs volume curves	0	8	1	4	0	7	7	8	221.3
8	6.7	10 second measurement	2	8	0	0	3	5	5	9	169.0
9	5.7	Single hand device	8	0	1	5	8	4	4	2	179.1
10	4.8	Saving measurement, consulting later	0	0	3	5	0	0	0	3	45.9
11	3.8	$100 cost limit	6	7	10	5	7	8	8	8	206.1
12	2.9	Method of viewing results	7	0	3	5	6	0	0	3	65.9
13	1.9	Wireless and battery operated	9	0	6	10	8	5	5	7	90.8
14	1.0	Streamline, single package design	10	4	3	3	6	5	5	4	39.2
		Technical importance score	307.1	587.6	426.9	328.6	425.9	469.5	453.3	496.2	3047.2
		Importance %	10%	19%	14%	11%	14%	15%	15%	16%	
		Priorities rank	**8**	**1**	**5**	**7**	**6**	**3**	**4**	**2**	

Figure 25: Quality Function Deployment

Appendix B – Final Design

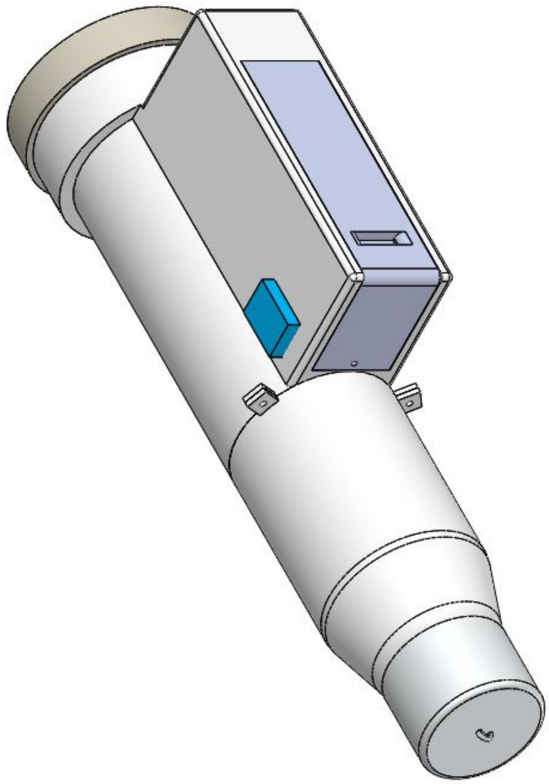


Figure *26*: An Isometric View of the Device

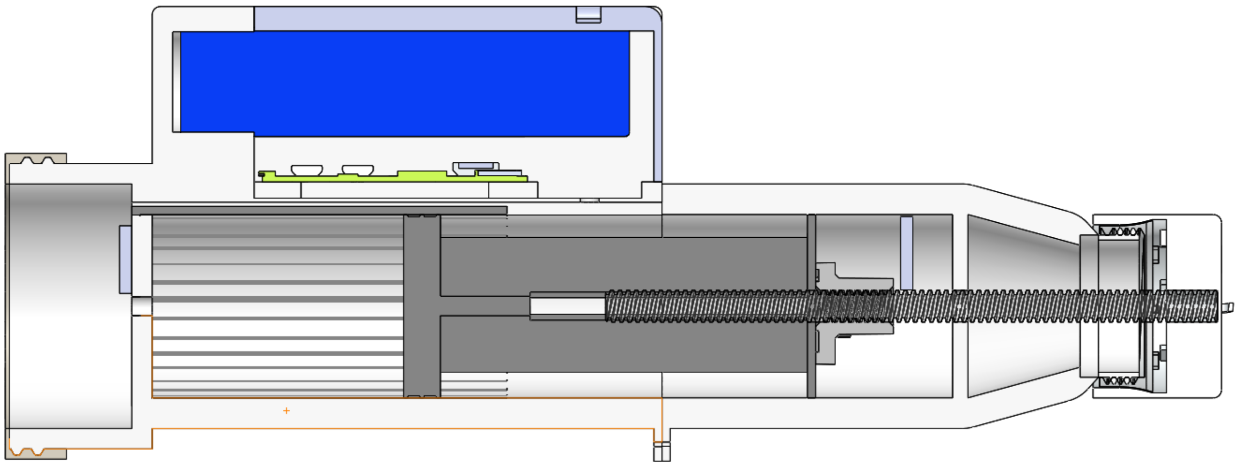


Figure *27*: A Cross-sectional View of the Device

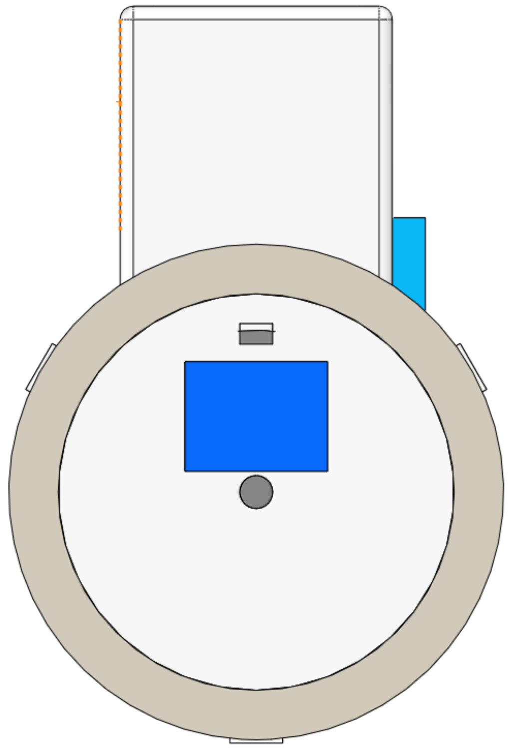


Figure *28*: A Front View of the Device

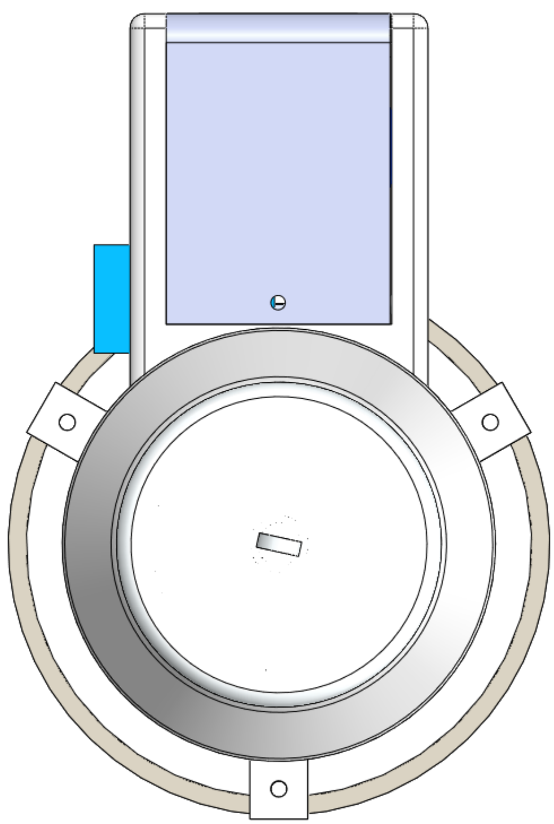


Figure *29*: A Back View of the Device

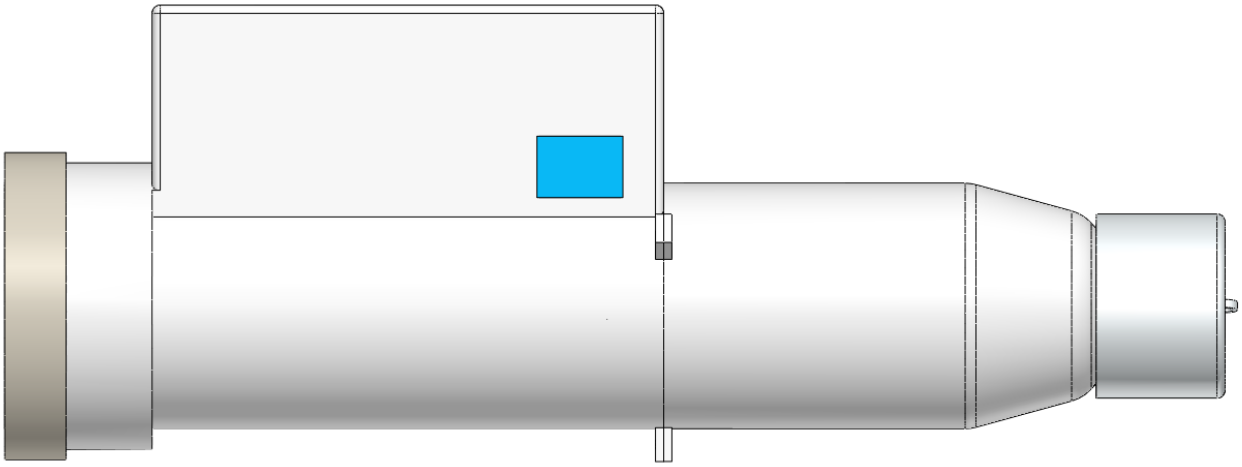
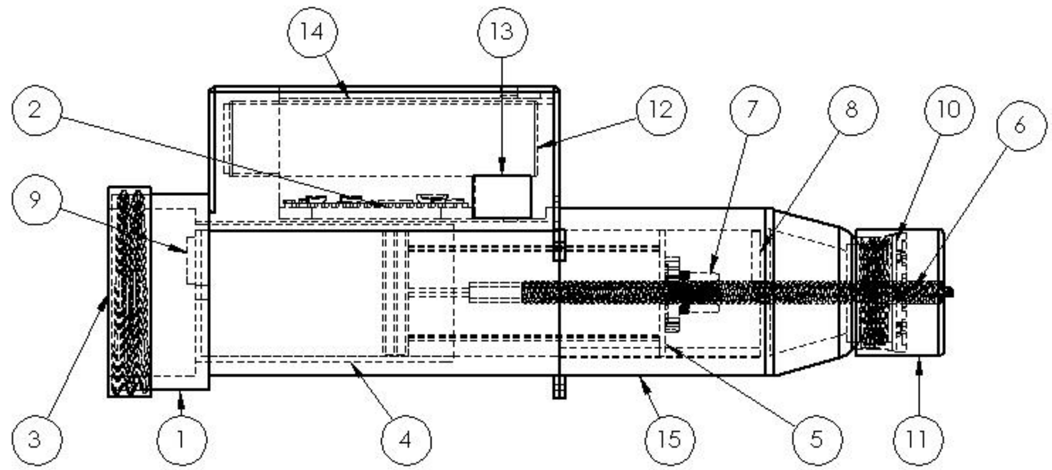


Figure *30*: A Side View of the Device

Appendix C – Bill of Materials



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Main Housing	3D Printer PLA Housing	1
2	RaspberryPiZeroW	Processing Computer	1
3	Mason Lid	Cap to secure adhesive film	1
4	Syringe Tube	Fixed Syringe Tube	1
5	Syringe Plunger	Movable Plunger	1
6	Fast Travel Lead Screw	Fixed, rotating lead screw	1
7	Fast Travel Lead Screw Nut	Ultra-precision, axially movable lead screw nut	1
8	VL6180X Distance Sensor	Optical distance sensor	1
9	BMP388 Pressure Sensor	BMP 388 pressure transducer	1
10	Knob Bearing	Threaded bearing with lock features	1
11	Knob Cap	Rotating knob with lock features	1
12	Battery	Rechargable battery to power device	1
13	Power Switch	Switch to turn on and off the device	1
14	Electronics door	Sliding door to acess electronic components	1
15	Syringe Housing	3D Printer PLA Housing	1

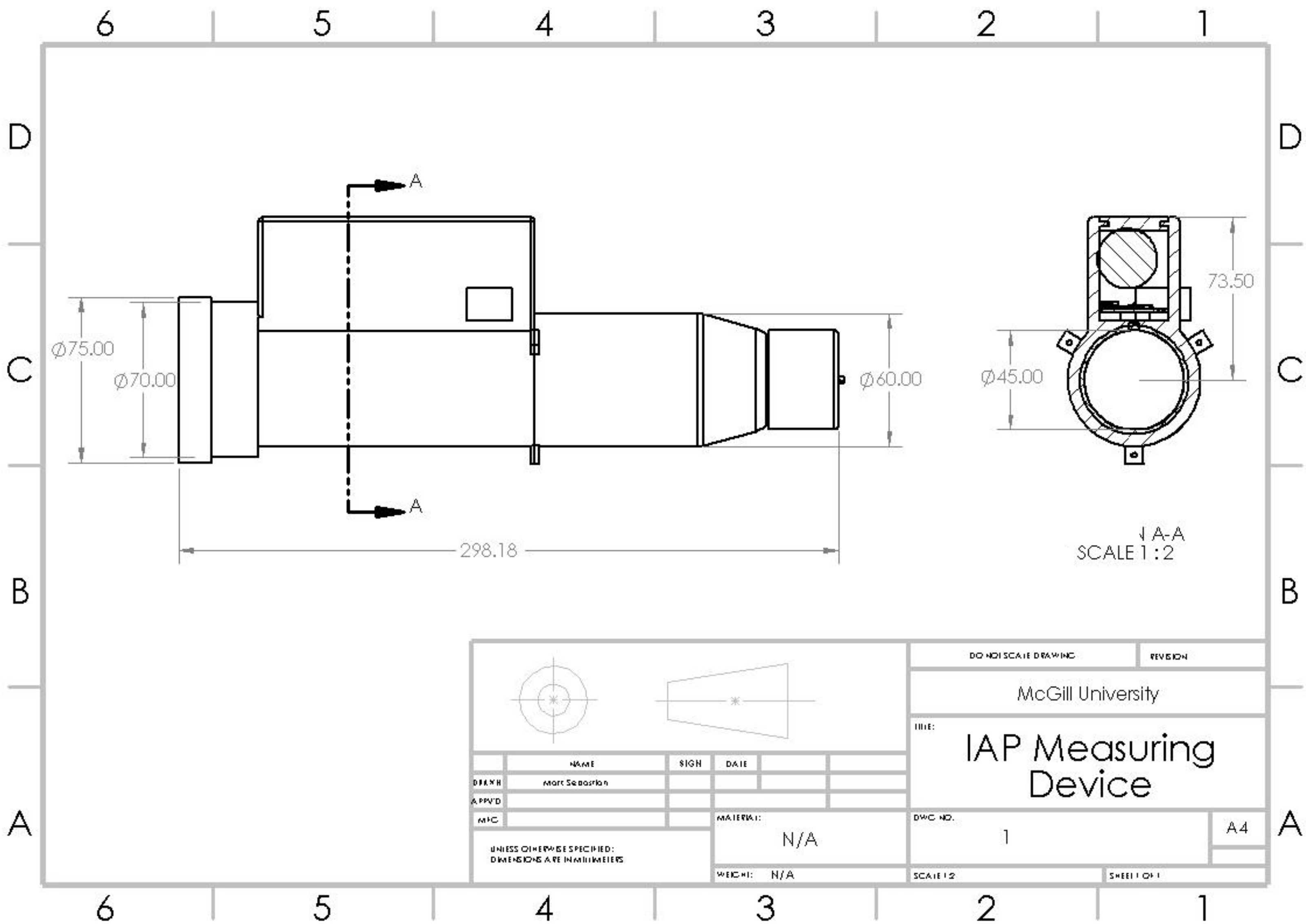
Cost of Materials and Part Numbers from Design

Item	Name	Company Part Number	Cost
1	Adafruit Micro-LIDAR Distance Sensor	Adafruit VL6180X	$19.81
2	Adafruit Precision Barometric Pressure	Adafruit BMP388	$14.13
3	Fast-travel Ultra-Precision Lead Screw M8 x 2.50 mm	McMaster-Carr 2391N24	$29.07
4	Flange Nut M8 X 2.50 mm	McMaster-Carr 2391N11	$60.73
5	Raspberry Pi Zero W	Raspberry Pi Foundation Pi W Zero	$42.99
6	32 GB microSD Card	SanDiskA1- SDSQUAR-032G-GN6MA	$10.23
7	22 Gauge Wire	BNTECHGO SW22A60008F10C2	$8.64
8	200 mL Syringe	OUNONA 8R6067551D909FP	$12.99
9	Portable 5000 mAh Power Bank	Poweradd PD-MP1037BK	$19.98
10	Rocker Switch	Linseray KCD1-101	$10.99
11	Medical tape	TMISHION wmv7adxp16-03	$11.99
12	Housing	NA	$150 (estimate)
Total			**$391.55**

Actual Cost of Materials and Part Numbers

Item	**Name**	**Cost**
1	Distance Sensor	$19.81
2	Precision Barometric Pressure Sensor	$14.13
3	Precision Lead Screw	$29.07
4	Flange Nut	$60.73
5	Raspberry Pi Zero W	$42.99
6	MicroSD Card	$10.23
7	22 Gauge Wire	$8.64
8	200 mL Syringe	$12.99
9	Power Bank	$19.98
10	Rocker Switch	$10.99
11	Medical Tape	$11.99
12	Housing	$142.73
Total		**$383.28**

Appendix D – Dimensions of Final Design



Appendix E – IAP Measurement Program

app.py

```
import dash
import dash_core_components as dcc
import dash_html_components as html

print(dcc.__version__)

app = dash.Dash(__name__)

server = app.server
app.config.suppress_callback_exceptions = True

#Start the dash server
```

distance.py

```
import time
import board
import busio
import adafruit_vl6180x
import statistics

i2c = busio.I2C(board.SCL, board.SDA)

sensor = adafruit_vl6180x.VL6180X(i2c)

def distance():

    distance_meas = []

    for index in range(25):
        distance_meas.append(sensor.range)

    distance_var = statistics.mean(distance_meas)
    return distance_var
    #already in mm
```

index.py

```
import dash_core_components as dcc
import dash_html_components as html
```

```
from dash.dependencies import Input, Output

from app import app
from apps import database, measurement, home, posttest, config
import pandas
import glob

app.layout = html.Div([
    dcc.Location(id='url', refresh=False),
    html.Div(id='page-content')
])

#callback sets the layout for the main menu
@app.callback(Output('page-content', 'children'),
              [Input('url', 'pathname')])
def display_page(pathname):
    if pathname == '/':
         return home.layout
    elif pathname == '/measurement':
         return measurement.layout
    elif pathname == '/database':
         return database.layout
    elif pathname == '/posttest':
         return posttest.layout
    else:
        return '404'

if __name__ == '__main__':
    app.run_server(port = 8077, host='192.168.0.103')
    #get the ip above by running hostname -
I on RBP. Call sudo python3 index.py to run the program.
```

pressure.py

```
import time
import board
import busio
import adafruit_bmp3xx
i2c = busio.I2C(board.SCL, board.SDA)
bmp = adafruit_bmp3xx.BMP3XX_I2C(i2c)
bmp.pressure_oversampling = 8
bmp.temperature_oversampling = 2

def pressure():
    return bmp.pressure*0.75
```

```
#hpa*0.75=mmHg

```

init.py (intentionally empty – required for Python)

```

```

config.py

```
#Config file necessary to share variables between the different page layouts
X = []
Y = []
time1 = 0
time2 = 0
filename = ''
```

database.py

```
import plotly
import plotly.graph_objs as go
import flask
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Output, Input
from apps import commonmodules, config
import pandas as pd
from app import app
import glob
import os

#Layout for database page
layout = html.Div([
    commonmodules.get_menu(),
    html.H3('IAP measurement database'),
    html.P('Select measurement to download or delete'),
    dcc.Dropdown(id="dropdown", style={'width': '45%', 'display': 'inline-
block', 'vertical-align': 'middle'}),
    dcc.Interval(
        id='page-update',
        interval=0.5*1000
    ),
    html.P(html.Br()),
    html.A('Download CSV', id='my-link', className="button"),
```

```
    html.A('Delete file', id='delete-
link', className="button", style={'color':'red'}),
    dcc.Graph(id='graph_database'),
    html.Div(id='IAP-output-database'),
])
#Callback for dropdown menu options, needs interval to refresh if ever one is del
eted
@app.callback(
    Output('dropdown', 'options'),
    [Input('page-update', 'n_intervals')])
def update_dropdown(value):
    config.filelist = glob.glob("/home/pi/app/data/*.csv")
    return [{'label': i, 'value': i} for i in config.filelist]

#Callback for downloading a CSV
@app.callback(Output('my-link', 'href'), [Input('dropdown', 'value')])
def update_link(value):
    return '/dash/urlToDownload?value={}'.format(value)

#Flask instance for downloading CSV
@app.server.route('/dash/urlToDownload')
def download_csv():
    value = flask.request.args.get('value')
    return flask.send_file(value,
                           mimetype='text/csv',
                           attachment_filename=value,
                           as_attachment=True)

#Callback for deleting file
@app.callback(Output('delete-link', 'href'), [Input('dropdown', 'value')])
def update_link2(value):
    return '/dash/urlToDelete?value={}'.format(value)

#Flask instance for deleting file
@app.server.route('/dash/urlToDelete')
def delete_csv():
    value = flask.request.args.get('value')
    os.remove(value)
    return ('', 204)

#Callback to produce graph
@app.callback(Output('graph_database', 'figure'), [Input('dropdown', 'value')])
def update_graph_scatter(value):

    df = pd.read_csv(str(value))
    X = df.iloc[:, 0].tolist()
```

```
    Y = df.iloc[:, 1].tolist()

    trace = plotly.graph_objs.Scatter(
        x=X,
        y=Y,
        name='Scatter',
        mode='lines+markers'
    )

    return {'data': [trace],
            'layout': go.Layout(
                height=550,
                xaxis=dict(range=[min(X), max(X)], title = df.columns[0]),
                yaxis=dict(range=[min(Y), max(Y)], title = df.columns[1])),

            }

#Callback to calculate and output IAP value
@app.callback(Output('IAP-output-database', 'children'),
              [Input('dropdown', 'value')])
def display_value_2(value):
    if (value == None):
        return
    df2 = pd.read_csv(str(value))
    Y_2 = df2.iloc[:, 1].tolist()
    IAP_2 = sum(Y_2)/len(Y_2)
    return html.Div([
        html.H4('IAP value of {:.2f} mmHg, gender is {}'.format(IAP_2, df2.column
s[2]), style={'textAlign': 'center'})
    ])
```

commonmodules.py

```
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output

#Returns menu common to all pages
def get_menu():
    menu = html.Div([

        dcc.Link('Home', href='/', className="button"),
        dcc.Link('IAP measurement', href='/measurement', className="button"),
        dcc.Link('Database', href='/database', className="button"),
    ], className="row all-tabs")
```

```
    return menu
```

posttest.py

```
import dash
from dash.dependencies import Output, Input
import dash_core_components as dcc
import dash_html_components as html
import plotly
import plotly.graph_objs as go
from app import app
from apps import config, posttest, commonmodules
import pandas as pd

#Layout for posttest page
layout = html.Div([
    commonmodules.get_menu(),
    html.H3('IAP measurement Plot'),
    dcc.Graph(id='graph'),
    html.Div(id='intermediate-value', style={'display': 'none'}),
    html.Div(id='IAP-output'),
])
#produces graph
@app.callback(Output('graph', 'figure'), [Input('intermediate-
value', 'children')])
def post_graph(value):

    df = pd.read_csv('/home/pi/app/data/' + config.filename + '.csv')
    X = df.iloc[:, 0].tolist()
    Y = df.iloc[:, 1].tolist()

    trace = plotly.graph_objs.Scatter(
        x=X,
        y=Y,
        name='Scatter',
        mode='lines+markers'
    )

    return {'data': [trace],
            'layout': go.Layout(
                height = 550,
                xaxis=dict(range=[min(X), max(X)], title = df.columns[0]),
```

```
                yaxis=dict(range=[min(Y), max(Y)], title = df.columns[1])),

            }
#produces IAP value at bottom of the page
@app.callback(Output('IAP-output', 'children'),
              [Input('intermediate-value', 'children')])
def display_value_1(value):
    df = pd.read_csv('/home/pi/app/data/' + config.filename + '.csv')
    Y = df.iloc[:, 1].tolist()
    IAP = sum(Y)/len(Y)
    return html.Div([
        html.H4('IAP value of {:.2f} mmHg, gender is {}'.format(IAP, df.columns[2
]), style={'textAlign': 'center'})
    ])
```

measurement.py

```
import dash
from dash.dependencies import Output, Input
import dash_core_components as dcc
import dash_html_components as html
import plotly
import plotly.graph_objs as go
from app import app
from apps import database, commonmodules, measurement, home, config, measurement_
PT, measurement_PV, measurement_VT
import time
import csv

#Produces the layout for IAP measurement page
layout = html.Div([
    commonmodules.get_menu(),
    html.H3('Real-time IAP measurement'),
    html.P('Please select what data to collect and display live:'),
    dcc.Dropdown(
        id='graph-type',
        options=[
            {'label': 'Pressure-Volume', 'value': 'PV'},
            {'label': 'Pressure-Time', 'value': 'PT'},
            {'label': 'Volume-Time', 'value': 'VT'}
        ],
        value='PT',
        style={'width': '45%', 'display': 'inline-block', 'vertical-
align': 'middle'}
    ),
```

```
    html.P(html.Br()),
    html.P("Please enter the patient's gender:"),
    dcc.Dropdown(
        id='gender-select',
        options=[
            {'label': 'Male', 'value': 'M'},
            {'label': 'Female', 'value': 'F'},
        ],
        value='M',
        style={'width': '45%', 'display': 'inline-block', 'vertical-
align': 'middle'}
    ),
    html.P(html.Br()),
    html.P('Please enter a filename to save the data and click start when you wis
h to begin the test'),
    dcc.Input(id='input-
box', type='text',style={'width': '20%', 'display': 'inline-block'}),
    html.Button('Start', id='button', style={'width': '10%', 'display': 'inline-
block'}),
    html.Div(id='page-1-content'),
])
#Click start button to start plotting the selected graph type from the dropdown m
enu. Calls the appropriate function to plot graph
@app.callback(
    dash.dependencies.Output('page-1-content', 'children'),
    [dash.dependencies.Input('button', 'n_clicks'),
    dash.dependencies.Input('graph-type', 'value'),
    dash.dependencies.Input('gender-select', 'value'),
    dash.dependencies.Input('input-box', 'value')])
def update_output(n_clicks, gtype, sex, name):
    if(n_clicks == 1):
        config.X = []
        config.Y = []
        config.time1 = time.time()
        config.filename = name
        if (gtype == "PT"):
            with open('/home/pi/app/data/' + config.filename + '.csv', 'w', newli
ne='') as file:
                writer = csv.writer(file)
                writer.writerow(["Time (s)", "Pressure (mmHg)", sex])
            return measurement_PT.layout
        elif (gtype == "PV"):
            with open('/home/pi/app/data/' + config.filename + '.csv', 'w', newli
ne='') as file:
                writer = csv.writer(file)
                writer.writerow(["Pressure (mmHg)", "Volume (cm^3)", sex])
```

```
            return measurement_PV.layout
        elif (gtype == "VT"):
            with open('/home/pi/app/data/' + config.filename + '.csv', 'w', newli
ne='') as file:
                writer = csv.writer(file)
                writer.writerow(["Time (s)", "Distance (mm)", sex])
            return measurement_VT.layout
```

measurement_VT.py

```
import dash
from dash.dependencies import Output, Input
import dash_core_components as dcc
import dash_html_components as html
import plotly
import plotly.graph_objs as go
from distance import distance
from app import app
from apps import config, posttest, commonmodules
import time
import csv

#Layout for VT measurement
layout = html.Div([
    dcc.Graph(id='live-graph-VT',
                animate=False,),
    dcc.Interval(
        id='graph-update-VT',
        interval=0.6*1000
    ),
    dcc.Link('Stop', href='/posttest', className="button"),

])

#Callback for VT graph
@app.callback(Output('live-graph-VT', 'figure'),
              [Input('graph-update-VT', 'n_intervals')])
def update_graph_VT(n):
    volume_meas = distance()
    config.time2 = time.time()
    time_interval = config.time2 - config.time1
    config.X.append(time_interval)
    config.Y.append(volume_meas)
```

```
    with open('/home/pi/app/data/' + config.filename + '.csv', 'a', newline='') a
s file:
        writer = csv.writer(file)
        writer.writerow([time_interval, volume_meas])

    trace = plotly.graph_objs.Scatter(
        x=config.X,
        y=config.Y,
        name='Scatter',
        mode='lines+markers'
    )

    return {'data': [trace],
            'layout': go.Layout(
                height=600,
                xaxis=dict(range=[min(config.X), max(config.X)], title = 'Time (s
)'),
                yaxis=dict(range=[min(config.Y), max(config.Y)], title = 'Volume
(cm^2)')),
            }
```

Measurement_PV.py

```
import dash
from dash.dependencies import Output, Input
import dash_core_components as dcc
import dash_html_components as html
import plotly
import plotly.graph_objs as go
from distance import distance
from pressure import pressure
from app import app
from apps import config, posttest, commonmodules
import time
import csv

#Produces layout for PV measurement
layout = html.Div([
    dcc.Graph(id='live-graph-PV',
              animate=False,),
    dcc.Interval(
        id='graph-update-PV',
        interval=0.6*1000
    ),
    dcc.Link('Stop', href='/posttest', className="button"),
```

```
])

#Callback for PV measurement
@app.callback(Output('live-graph-PV', 'figure'),
              [Input('graph-update-PV', 'n_intervals')])
def update_graph_PV(n):
    pressure_meas = pressure()
    volume_meas = distance()
    config.X.append(pressure_meas)
    config.Y.append(volume_meas)

    with open('/home/pi/app/data/' + config.filename + '.csv', 'a', newline='') a
s file:
        writer = csv.writer(file)
        writer.writerow([pressure_meas, volume_meas])

    trace = plotly.graph_objs.Scatter(
        x=config.X,
        y=config.Y,
        name='Scatter',
        mode='lines+markers'
    )

    return {'data': [trace],
            'layout': go.Layout(
                height=600,
                xaxis=dict(range=[min(config.X), max(config.X)], title = 'Volume
(cm^3)'),
                yaxis=dict(range=[min(config.Y), max(config.Y)], title = 'Pressur
e (mmHg)')),
            }
```

Measurement_PT.py

```
import dash
from dash.dependencies import Output, Input
import dash_core_components as dcc
import dash_html_components as html
import plotly
import plotly.graph_objs as go
from pressure import pressure
from app import app
from apps import config, posttest, commonmodules
import time
import csv
```

```
#Produces layout for PT measurement
layout = html.Div([
    dcc.Graph(id='live-graph-PT',
            animate=False,),
    dcc.Interval(
        id='graph-update-PT',
        interval=0.6*1000
    ),
    dcc.Link('Stop', href='/posttest', className="button"),
])

#Produces graph for PT measurement
@app.callback(Output('live-graph-PT', 'figure'),
              [Input('graph-update-PT', 'n_intervals')])
def update_graph_PT(n):
    pressure_meas = pressure()
    config.time2 = time.time()
    time_interval = config.time2 - config.time1
    config.X.append(time_interval)
    config.Y.append(pressure_meas)

    with open('/home/pi/app/data/' + config.filename + '.csv', 'a', newline='') a
s file:
        writer = csv.writer(file)
        writer.writerow([time_interval, pressure_meas])

    trace = plotly.graph_objs.Scatter(
        x=config.X,
        y=config.Y,
        name='Scatter',
        mode='lines+markers'
    )

    return {'data': [trace],
            'layout': go.Layout(
                height=600,
                xaxis=dict(range=[min(config.X), max(config.X)], title = 'Time (s
)'),
                yaxis=dict(range=[min(config.Y), max(config.Y)], title = 'Pressur
e (mmHg)')),
            }
```

home.py

```
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
from apps import database, measurement, commonmodules
import os
from app import app

#Layout for home page
layout = html.Div([
    commonmodules.get_menu(),
    html.H3('Welcome'),
    html.P('Before taking an IAP measurement, please read the instructions below.
'),
    html.Br(),
    html.P('This device is intended to be used to measure intra-
abdominal pressure (IAP). [Give instructions for IAP measurement].'),
    html.P('To start a test, navigate to the "IAP measurement" tab, input the req
uired information and click start. To end the test, click stop.'),
    html.P('To view previous tests, download data or delete the test from the dev
ice, navigate to the database tab.'),
    html.P('To turn off the device, click on the power button below. Wait 10 seco
nds and press the power button on the device.'),
    html.P('Add titles and axes to graphs'),
    html.Button('Power', id='button', style={'width': '10%', 'display': 'inline-
block', 'color':'red'}),
    html.Div(id='home-content'),
])

#callback for power button
@app.callback(
    Output('home-content', 'children'),
    [Input('button', 'n_clicks')])
def update_output(n_clicks):
    if(n_clicks == 1):
        os.system('sudo shutdown -h now')
        return "Powering off...Please wait 10 seconds and press the power button 
on the device."
```

Appendix F – Program Interface

When the program is first accessed, the Home page is displayed by default.

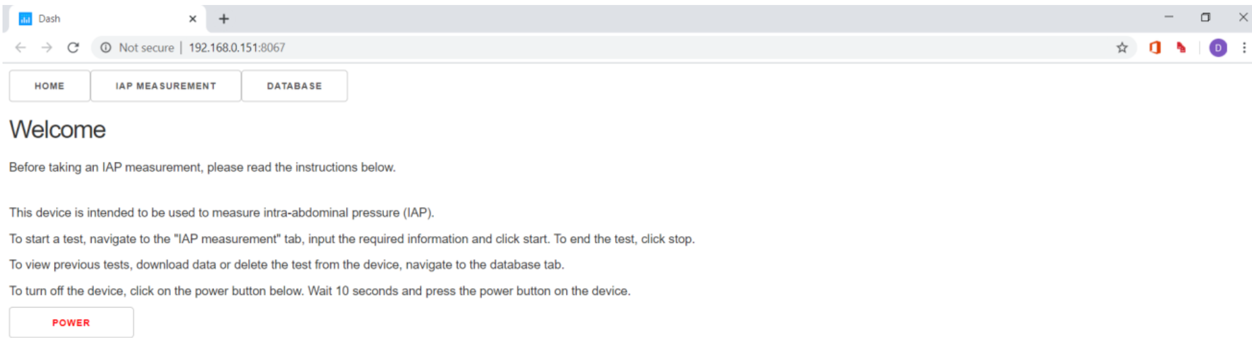


Figure 31: Home page of program

The user can navigate to the IAP Measurement page.

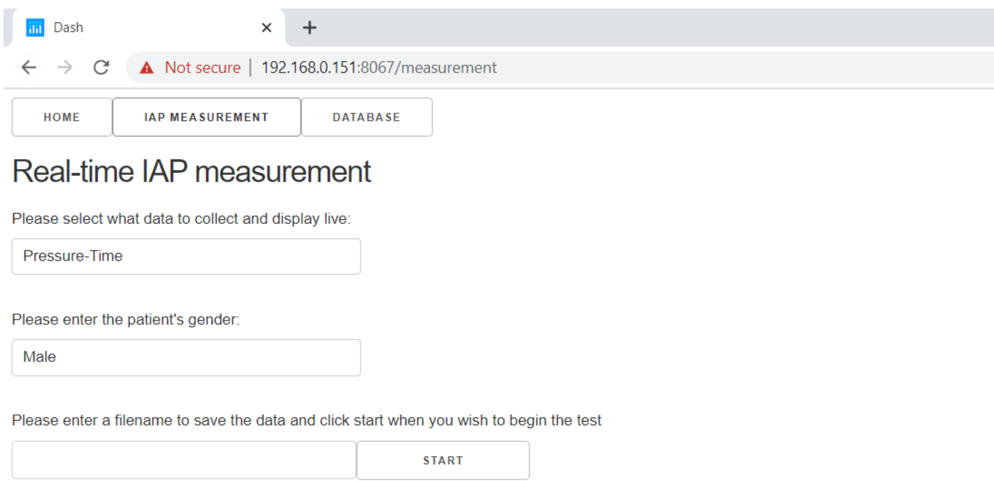


Figure 32: IAP Measurement page of program

Different options are available to the user. Three types of graphs may be plotted:

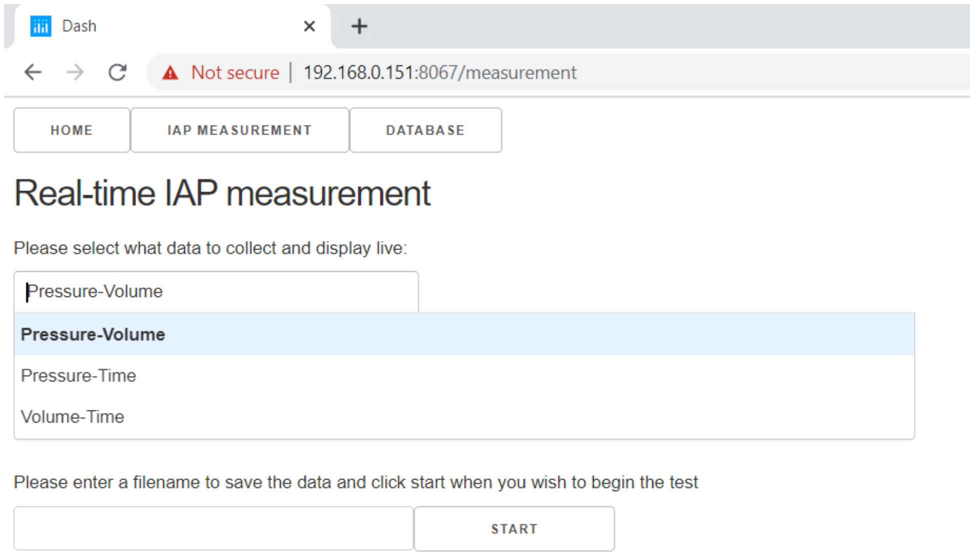


Figure 33: Data collection options

The pressure-volume curve is desired as per the project requirements. However, pressure-time and volume-time curves are also useful to testing and prototyping purposes and may also have potential uses in device application. The user must also input the patient's gender, as there are anatomical differences between men and women which mean that different correlation coefficients are required. Finally, the user is asked to name the test. When the user starts a test, the graph will immediately start plotting sensor data.

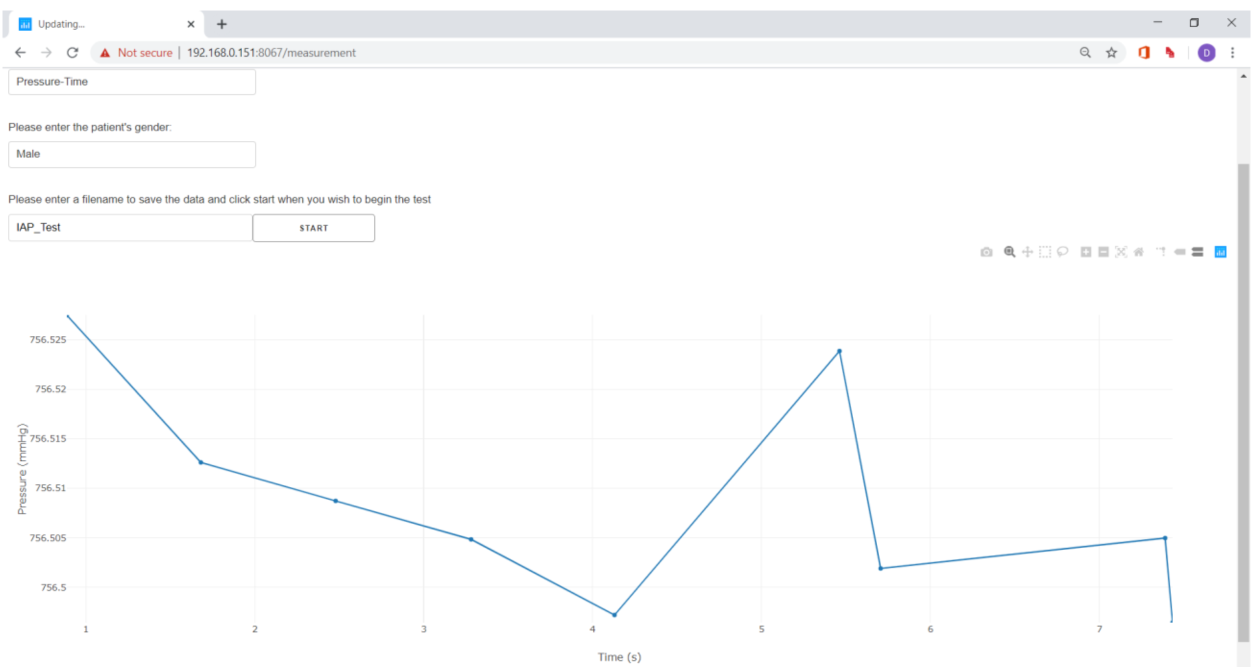


Figure 34: Live-plotting of pressure-time data

Once the test is complete, the user may scroll to the bottom of the page and press the stop button. This brings the user to the posttest page.

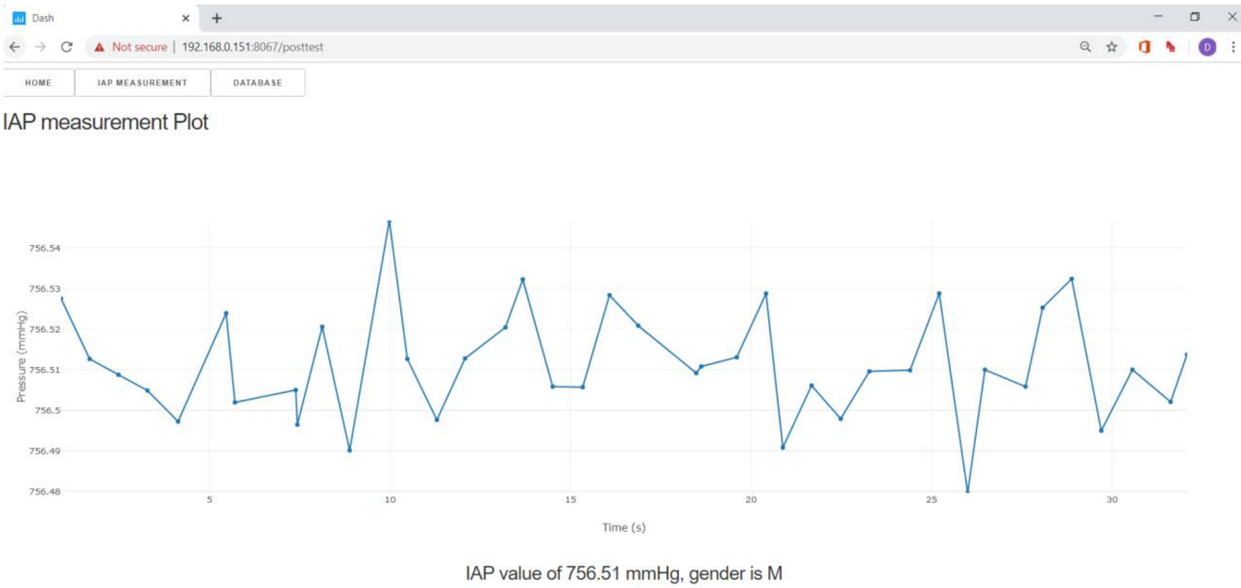


Figure 35: Posttest plot and IAP measurement

The IAP value measured is normally displayed under the graph. For testing purposes, it currently calculates an average of all data points collected. Subsequently, the user may choose to go over previous tests to compare, download or delete data. This can be done in the Database page.

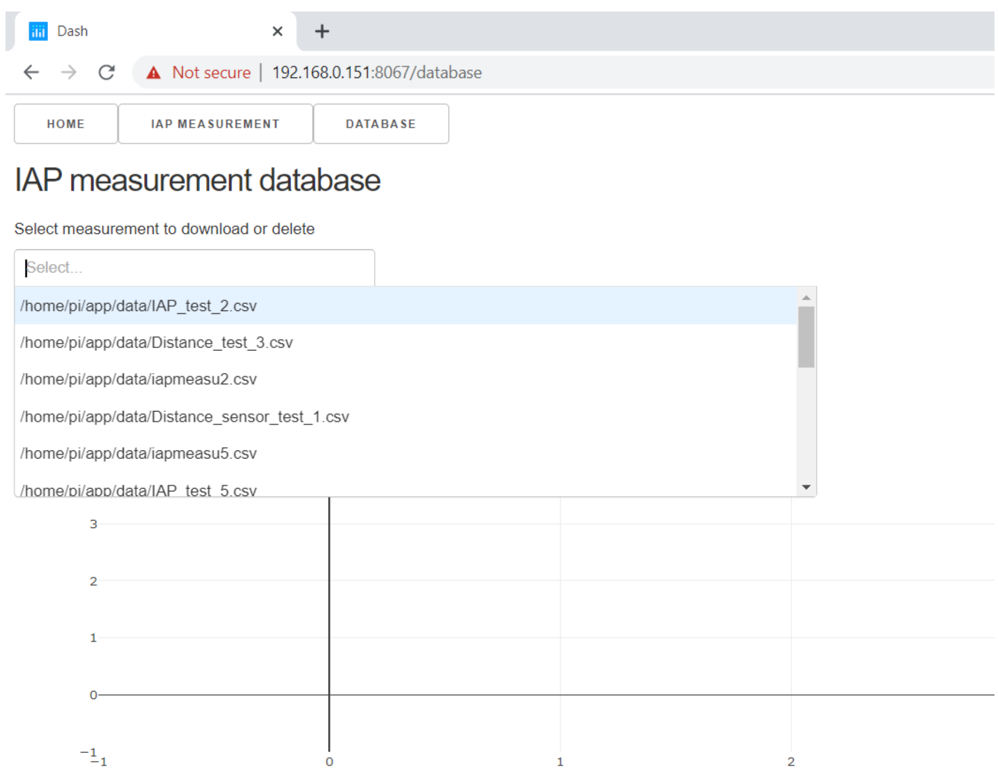


Figure 36: Database test selection

Once the user selects whichever test they would like to download or delete, it is shown, along with the calculated IAP:

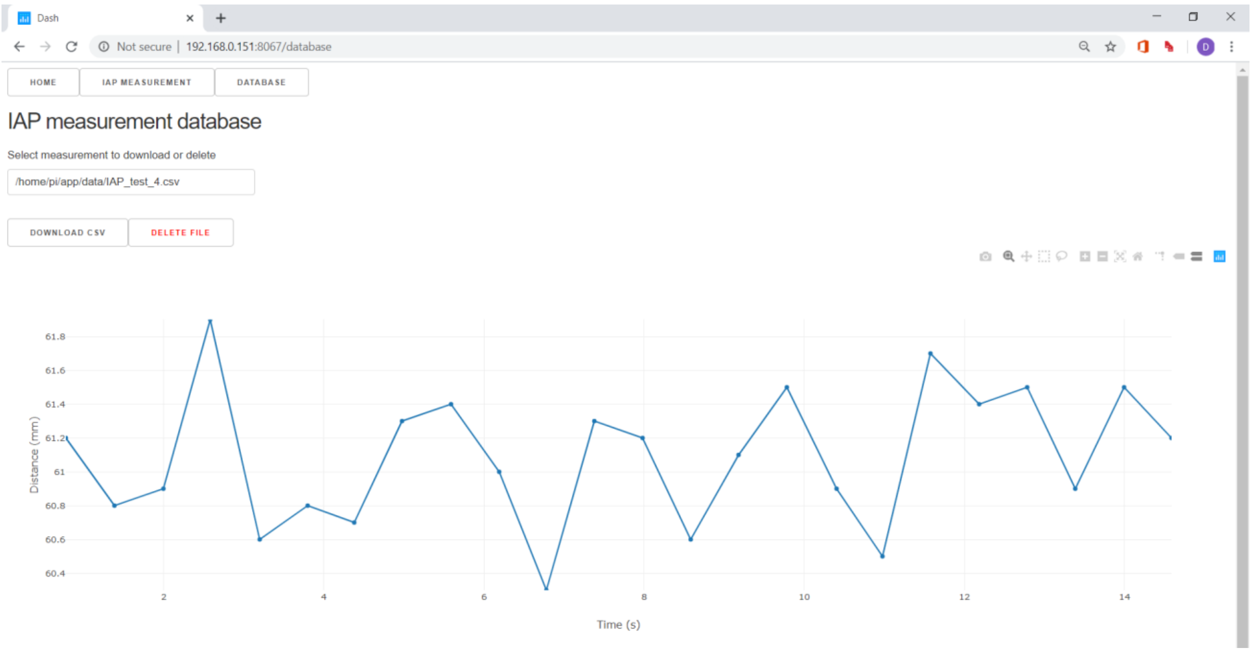


Figure 37: Database file viewing

If the user chooses to delete the file, it is permanently deleted from the database. If the user wishes to download the data, a csv file is downloaded.

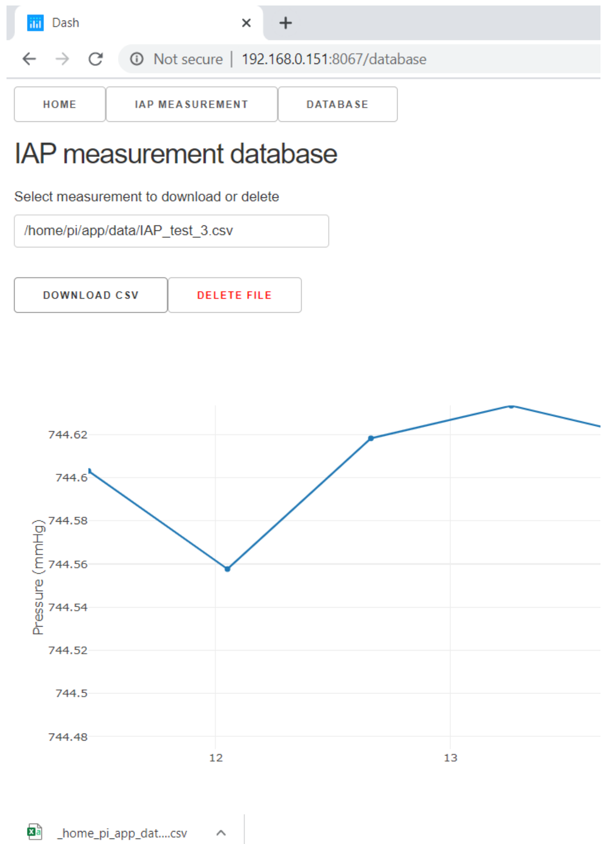


Figure 38: Database file download

	A	B	C
1	Time (s)	Pressure (mmHg)	
2	11.4575	744.6031	
3	12.05014	744.5577	
4	12.6613	744.6183	
5	13.25932	744.6334	
6	13.85449	744.6183	
7	14.45105	744.6258	
8	15.05837	744.5272	
9	15.65201	744.6183	
10	16.25399	744.5955	
11	16.84836	744.5727	
12	17.45362	744.4742	
13			
14			
15			
16			

Figure 39: Downloaded CSV file

Appendix G – Photographs of 3D Printed Casing

Figure 40: Top Part of Housing Without Cap

Figure 41: Top Part of Housing with Cap

Figure 42: Bottom View of Top Part of Housing

Figure 43: Full View of 3D Print

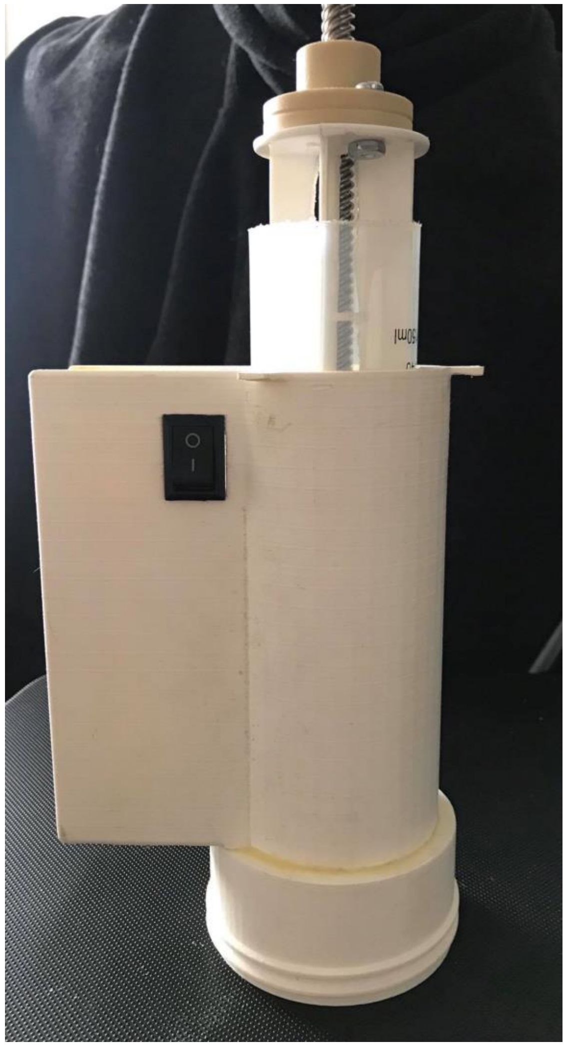


Figure 44: Half assembled casing

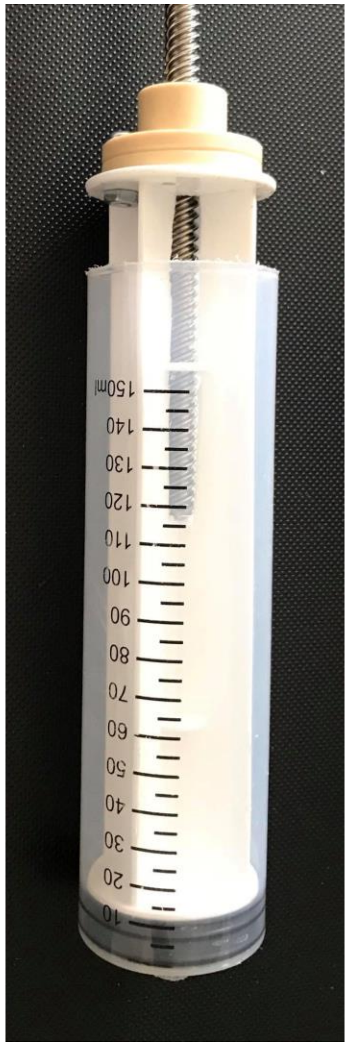


Figure 45: Syringe and Lead Screw

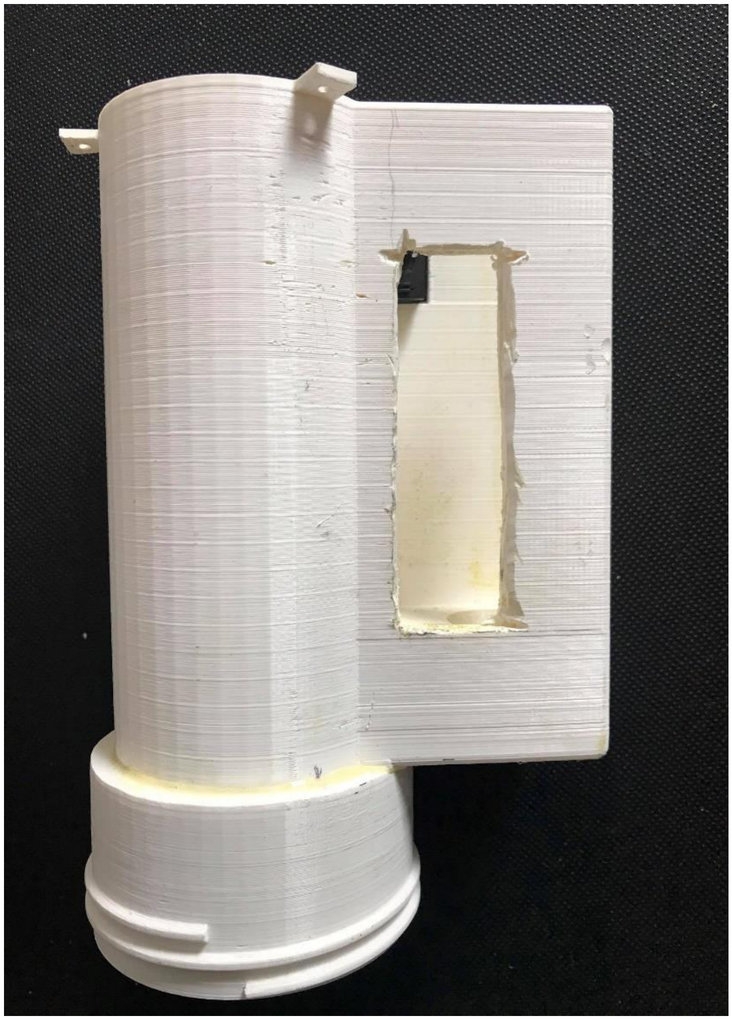


Figure 46: Slot made in 3D printed casing to leave room for wires

Figure 47: Front view, without door

Figure 48: Front view with door

Figure 49: Front view with battery

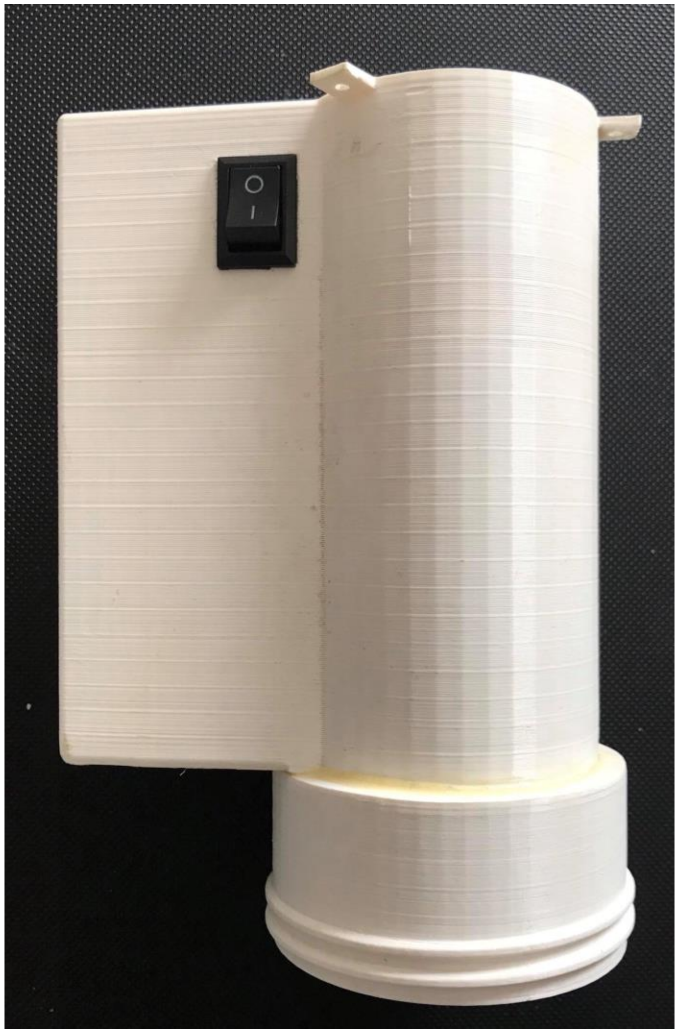


Figure 50: Side view of 3D print

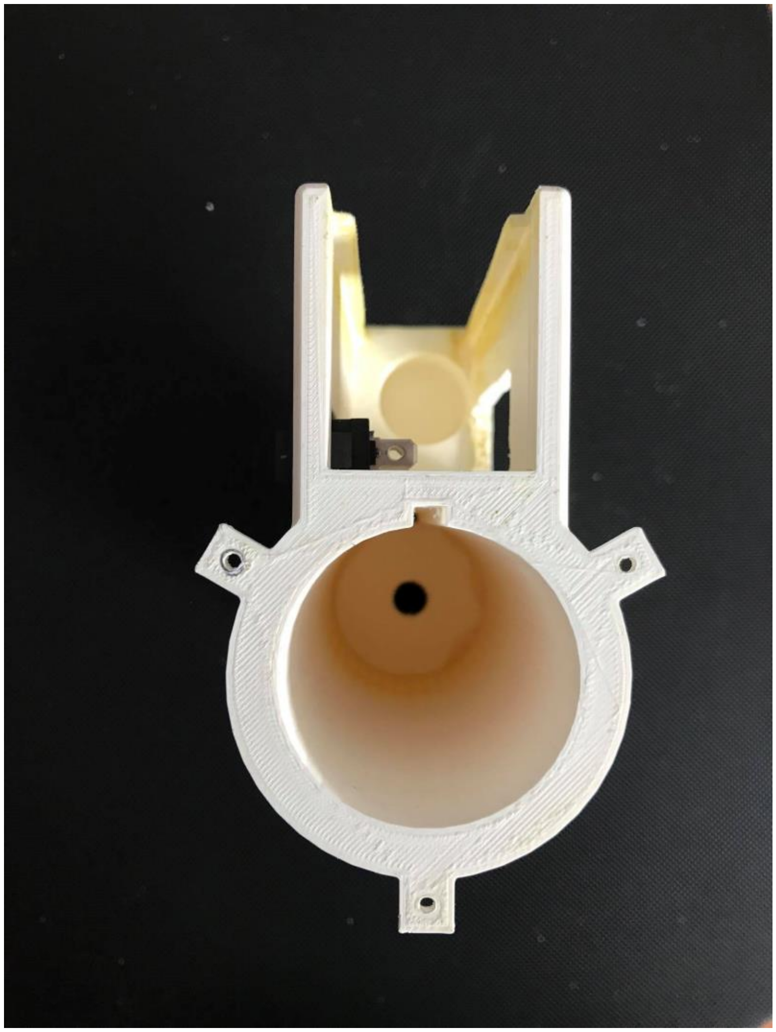


Figure 51: Top view of 3D print

Figure 52: Bottom view of 3D printed casing