Critical Design Review for a Sensor Array to Measure Pressure at the Socket-Stump Interface

Samuel Johnson

Payton McGraw

Chirayu Salgarkar

Client: Dr. Ha Van Vo

Technical Advisors: Dr. Ha Van Vo, Dr. Anthony Choi

Manager: Dr. Stephen Hill

April 8, 2024

Table of Contents

Table of Figures

Table of Tables

Executive Summary

Engineering exists as an avenue to directly improve lives. One such avenue is prosthetic engineering; clinicians and engineers alike have studied ways to minimize harm onto patients during fitting. However, engineered solutions, often addons or patented medical devices themselves, are exorbitant in price, making them inaccessible to clinicians to afford and use. Ironically, that makes the device inaccessible to patients as well.

Our client, Dr. Ha Van Vo, Distinguished Professor of Biomedical Engineering and developer of the Mercer Universal General Prosthetic, requested our team to build a modular stump-socket interface pressure sensory device that is affordable (and therefore accessible to clinicians) and can measure contact pressures at 5 candidate locations on the leg. This device would aid fitting decisions for patients with diabetic neuropathy, who are unable to detect pressure changes due to nerve damage.

Our team of Sam Johnson, Payton McGraw, and Chirayu Salgarkar was chosen to build this device. A force sensitive resistor array system connected to an ESP-32 was chosen as the design for the device. The device is economical: designs used readily available force sensitive resistors (FSRs), economical microcontrollers, handmade multiplexer PCBs, and 3D printed enclosures, rather than proprietary hardware and software. The device cost is less than $275, far less than the thousands of dollars needed for proprietary software like TekScan.

The device was rigorously tested to ensure that it was satisfactory for clinical use. Tests include a sensor press test, a weight scalability test, and an overall clinical test. The device passed all tests.

Thus, the project was deemed successful for meeting all device specifications, feasibility criteria, and merit criteria. Nonetheless, we have written a set of further revisions and recommendations for improvements to the device for future years. For instance, we recommend the integration of resistors used in voltage dividers to a custom printed circuit board in future designs.

Attached herewith is a fully complete critical design review for a prosthetic stump-socket interface, which can measure patient pressure while fitting reliably at a low cost. This device complements Mercer's mission of providing prosthetics at low cost in remote settings, such as in Vietnam and Cambodia under the Mercer on Mission Program.

1 Introduction

Many individuals who require the use of prosthetics have comorbities (multiple chronic conditions), such as Diabetes Mellitus (Walicka et al., 2019). These comorbities have significant ramifications in the context of prosthetic fitting. For instance, neuropathy, or nerve damage, is a prevalent complication for diabetic patients; between 6 to 51 percent of adult diabetics have some form of peripheral neuropathy (Hicks and Selvin, 2019). Such patients may not be able to provide feedback during prosthetic fitting, which in extreme cases may cause sores and bleeding.

1.1 Client Description

To improve fitting procedures for such patients, Dr. Ha Van Vo, Distinguished University Professor of Biomedical Engineering and developer of the Universal General Prosthetic, has requested the development of a device for measuring contact pressure variations between the prosthetic socket and the human appendage. He then wants to use this design for potential further use under the auspices of the Mercer on Mission: Vietnam program, which fits prosthetics pro bono for Vietnamese amputees.

1.2 Problem Statement

Our team aimed to develop a stump-socket interface pressure sensory device that will:

1. Be affordable, so that many can be made for patients at low cost
2. Be able to measure the contact pressures at the tibial tuberosity, femoral head, fibular head, medial and lateral condyle of the femur, and the intersection of socket and stump closest to the ground
3. Be modular, i.e. be easily adaptable for a large group of patients.

2 Summary of Preliminary Design Review

2.1 Project Goal and Specifications

As described in our Preliminary Design Review, our client, Dr. Ha Van Vo wants “a device that works.” As a clinician, he is far more interested in an inexpensive engineered product that can detect pressure variations, rather than a high-level engineered product, often too expensive for a clinical setting. Our team chose a preliminary list of objectives to satisfy this goal:

- i. There must be enough sensor apparatuses to measure pressure at the tibial tuberosity, fibular head, medial condyle, lateral condyle, and bony protrusion at the bottom of the amputee’s stump.
- ii. Device sensors must be able to accurately read pressures up to 50 psi, as that is seen as the maximum contact pressure a patient with a prosthetic can withstand (Pearson et al., 1973).
- iii. Sensor array area must adequately cover each of the anatomical regions listed above.
- iv. Sensing areas are to be, at maximum, 2 millimeters (about 0.08 in) apart.
- v. The device must be integrable with pre-existing prosthetic sleeves provided by the client.

vi. The device compares read pressure values with allowable pressure values for patients with diabetic neuropathy.
vii. The device outputs data into an easy-to-read data application, such as Microsoft Excel.

2.2 Design Criteria and Specifications

2.2.1 Feasibility Criteria

Necessary device qualities were outlined by Dr. Ha Van Vo. They are listed in the table below.

Table 1. List and description of feasibility criteria for potential design

Feasibility Criterion	Description of Criterion
Biocompatibility	The materials used in the device must not cause any adverse effects or reactions to patient skin and body tissue.
Cost of Production	The device must cost less than $300 to produce.
Acceptable Sensor Apparatus Area	Must have sensor area to effectively measure pressure at the Tibial Tuberosity, the Fibular Head, the Medial Condyle, the Lateral Condyle, and the bony protrusion at the bottom of the stump.
Threshold of Measurement	The device must be able to accurately measure up to 50 psi.
Integrability	Must possess ability to attach to human skin and the silicon sleeve provided by Dr. Vo.

Since these criteria were compulsory for our client, the final proposed device design met all requirements.

2.2.2 Merit Criteria

Beyond initial feasibility requirements, project design was evaluated for four other criteria, those being proportion of cost of sensors, sensor area size, number of areas sensed per use, and number of sensors in apparatus. Scores for each merit criteria are calculated from 0-10, through various equations as seen in Table 2.

Table 2. List and description of merit criteria for potential design

Merit Criterion	Weight	Description of Criterion
Proportion of Cost of Sensors	20%	A design's merit score for this category will be calculated using the curve: $10\left(1-\frac{x}{300}\right)$ where x represents the calculated cost of the sensors.
Individual Sensor Area Size	25%	A design's merit score for this category will be calculated using the curve: $10\left(1-\frac{a}{12.25}\right)$ where a represents the total area covered by one sensor. In this case where an individual sensor area is greater than 12.25cm^2, the design will receive a score of 0 in this category due to lack of precision. Additionally, in the case where different sized sensors are used in a design, the average area of the sensors will be used.
Number of Areas Sensed per Usage	35%	A design's merit score for this category will be calculated using the curve: $10(\frac{s}{5})$ where s represents the number of sensed areas that can be measured in one use.
Number of Sensors in Apparatus	20%	A design's merit score for this category will be calculated using the curve: $10-\|10-x\|$ where x represents the number of sensors being used. If the number of sensors per array is within 20-30 sensors, the design will receive a score of 1 in this category. If the number of sensors per array exceeds 30 sensors, the design will receive a score of 0.

2.3 Roles and Responsibilities of Team Members

Our team was composed of three fourth-year undergraduate MUSE students: Samuel Johnson, Payton McGraw, and Chirayu Salgarkar.

2.3.1 Samuel Johnson

Samuel, a computer engineering student, handled the design and manufacturing of the electrical components of the project. They created the wiring schematics and designed the PCB integrating all project components. Additionally, they designed how the system connected to the microcontroller and wrote the code that allowed the microcontroller to read and transmit sensor values. They also assisted with any general needs other members of the project may have had.

2.3.2 Payton McGraw

Payton McGraw, a biomedical engineering student, handled the design and manufacturing of the mechanical parts and performed testing of the design to ensure proper function. For the prototype, Payton modeled both the housing unit base and top. In addition to this, he also modeled the apparatuses that were used to concentrate mass on the sensors during testing of the device. In the testing phase, Payton performed the sensor press test, the weight scalability test, and the clinical test with Chirayu.

2.3.3 Chirayu Salgarkar

Chirayu Salgarkar, a biomedical engineering and mathematics student, developed the experimental design for the sensor press test, weight scalability test, and the clinical test. He developed the array design ultimately used in the CDR, developed the socket used in testing, and conducted the experimentation on his device. He also coordinated meetings and organized the project members' tasks throughout the semester.

2.4 Visual Rendering of PDR Design

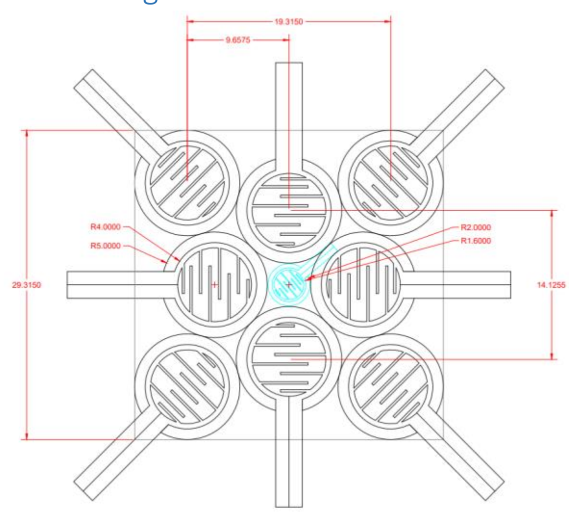

Figure 1. Square Sensor Array (Recommended PDR design).

The above figure shows the recommended array design given by the PDR. This design consisted of 9 circular FSRs, 8 of which had dimensions of 1 cm total diameter, 0.8 cm sensing diameter, while the central sensor had dimensions of 0.4 cm total diameter with a 0.36 cm sensor diameter. Here, the sensing area covers a square (approximately 3 cm in length). This device design was not used in the final build due to procurement issues. Further discussion on this can be seen in section 3.1.

3 Work Accomplished

3.1 Design Revisions

Ultimately, we did not build the recommended design from our PDR. We instead selected and built the 7-sensor array as seen in Figure 10. The main reason for this was an error discovered when ordering parts. The product page for the smaller center sensor listed incorrect measurements for overall size and sensing area. The actual dimensions on the data sheet revealed it was too large to be used in the way we intended. After reevaluation of our options and speaking with the client, our team pivoted to the 7-sensor array. Note that their merit scores were close, and we do not feel it affected the project's overall validity.

3.1.1 Electrical Revisions

Recall section 3.7 of the Preliminary Design Review:
"Each ESP32 will be acting as a WebSocket (a client-to-server communications protocol) using the built in Wi-Fi framework from the Arduino library. These will be connected to the same network as the receiving computer. After a cycle of reading every sensor connected to the microcontroller, it will send this data to the computer using Transmission Control Protocol (TCP). This allows for robust and error correcting communication between the microcontroller and computer. After this data is sent the process will restart for the next cycle of readings."

For our final design this wireless capability was omitted. The team and Dr. Vo agreed that it was not necessary for our project's scope, as it introduced a few confounding problems in complexity and number of microcontrollers. Mainly, for our selected microcontroller, enabling wireless communication deactivated nine of the fifteen ADC (analog to digital) pins. Implementing a solution to keep the project wireless was deemed not necessary as it did not provide much more convenience than using the device with traditional wired communication.

In turn, this decreased the number of microcontrollers from four to just one. In the final design, all 35 of the sensors are connected to one ESP32. The original plan for multiplexing multiple sensors to one ADC pin was still used. To achieve this, an external custom PCB (printed circuit board) was designed and manufactured. The schematic for this and how it connects to the rest of the design will be discussed later.

This greatly simplified our connection to the external PC as well. We were able to use the built-in serial communication output of the microcontroller. Then, we could read directly from that to interpret the data being read and sent. This contrasts with having multiple network sockets open and managing sending receiving data from each simultaneously.

3.1.2 Mechanical Revisions

When considering the amount of wiring that would be necessary for the design, it was decided that a housing unit should be made. This housing unit would serve to keep the excess wiring contained in a compact area. In addition to this, the housing unit would also serve to protect the circuitry used in the design. The design process for the piece is outlined below in section 3.2.2.

3.2 Final Design

3.2.1 Electrical Component Design

We will present the electrical component design in multiple parts to make it easier to digest. This includes:

1. Figure 2, how the sensors are wired to the multiplexer PCB and the ESP32.
2. Figure 3 and 4, the schematic and placement of the custom multiplexer PCB.
3. Figure 5, how all connect to the ESP32.

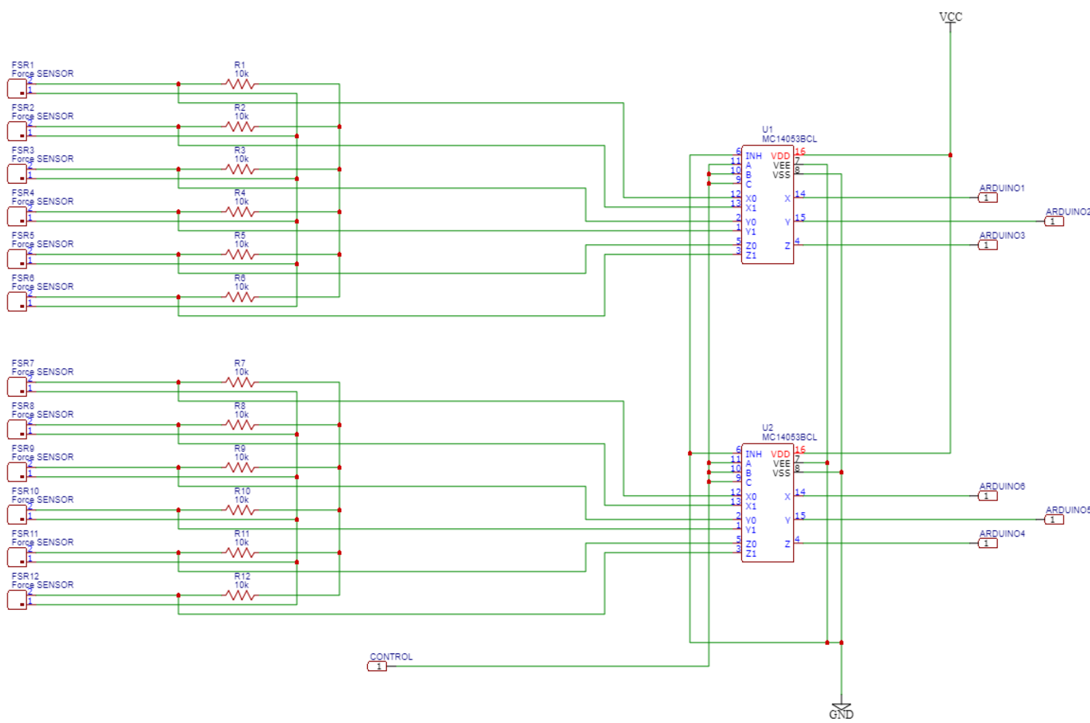

Figure 2. Group of 12 Sensors to MUXs on PCB.

Here we have illustrated how the sensors are wired in our final design. Each are voltage divider circuits, where one terminal of each sensor is connected directly to the 3.3V VCC of the ESP32. The other terminal is connected to an individual 10-ohm resistor that connects to the reference ground of the ESP32. This same connection is also tied to an input pin of a multiplexer on the custom PCB. These inputs are then selected to be passed through for analog reading based on the pin labeled ‘CONTROL’. This reading is controlled by a digital output pin on the ESP32. The figure above illustrates 12 sensors, but in total, we have 35 sensors wired in this configuration to make our five, seven sensor arrays.

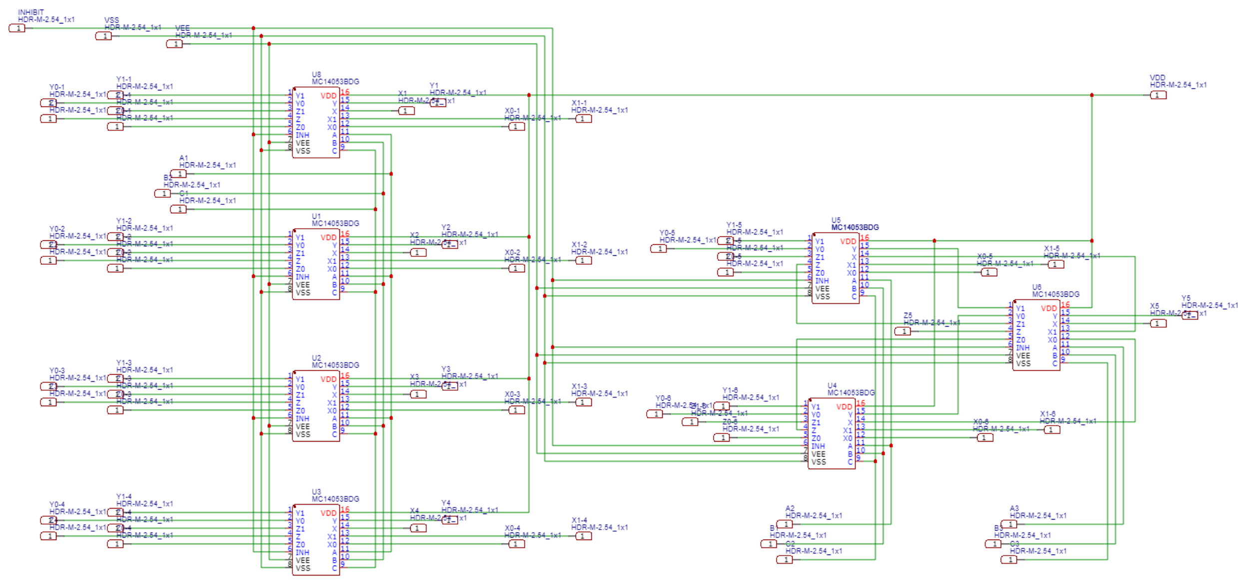

Figure 3. Schematic of PCB.

The figure above is the schematic for the PCB we designed to connect all 35 sensors to one ESP32. It incorporates seven analog multiplexer (MUX) integrated chips (ICs) whose pinout from the datasheet can be seen in Appendix G. Each has three, two to one analog multiplexers. There are six analog input pins and three analog output pins. In simple terms, the MUX selects one input to be sent to the output at a time based on the control lines. They also require connection to power. The VDD of each is connected to the 3.3V VCC of the ESP32. VEE and VSS are the digital and analog reference points respectively. These are both tied to the ground pin of the ESP32. The left 4 MUXs have control lines A1, B1, and C1. In our final design these are all tied to the same digital output pin of the ESP32. The outputs of these are ultimately connected directly to the ESP32's ADC pins. The right three MUXs are wired differently. We had to cascade these MUXs to accommodate the 35 sensors connecting to 15 ADC pins. They have control pins that are individually controlled by digital output pins to accommodate the resulting more complicated switching logic. Here 17 sensors are multiplexed down to just 3 output lines connected to the ESP32. Last, the inhibit feature of the multiplexers we used was not necessary, so it was also tied to the ground pin of the ESP32.

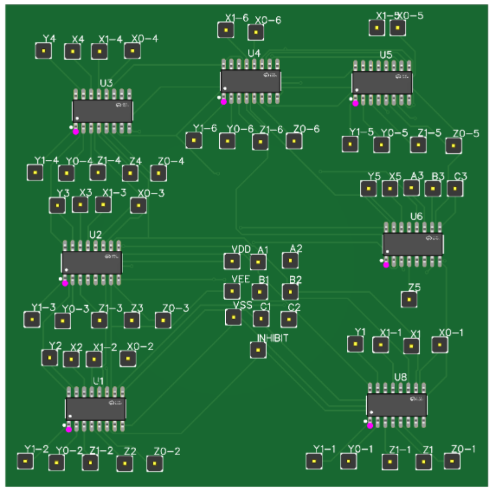

Figure 4. Visual of custom PCB placement.

This is the resulting PCB from the schematic described above. The pins and multiplexers were labeled and placed in a way that made routing and use easy. Due to the small size of the ICs, we ordered the boards already assembled.

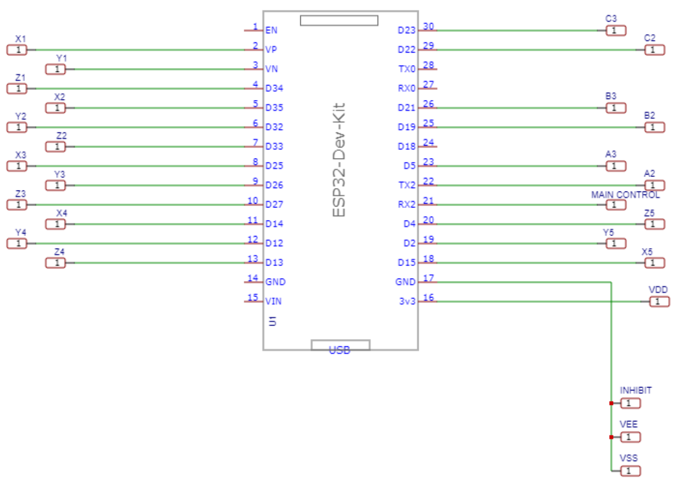

Figure 5. Connections to ESP 32.

Figure 5 illustrates how everything connects to the ESP32. The fifteen analog outputs of the MUXs connect to the fifteen ADC pins of the microcontroller. The control lines for the first four MUXs as described above are tied to pin 21, and the rest of the digital output pins are connected as seen in the figure. Additionally, note the connections to 3.3V and ground. It should also be noted that the pins that are not used could not be used because of other functions like serial communication disabling them.

The device was built using soldering protoboards. This was done to ensure the device was not fragile and maximize device durability in a clinical context. Additionally, as will be discussed later in recommendations, not incorporating the reference resistors into the PCB caused a great deal of wiring clutter. The next series of images will illustrate the process of building the device.

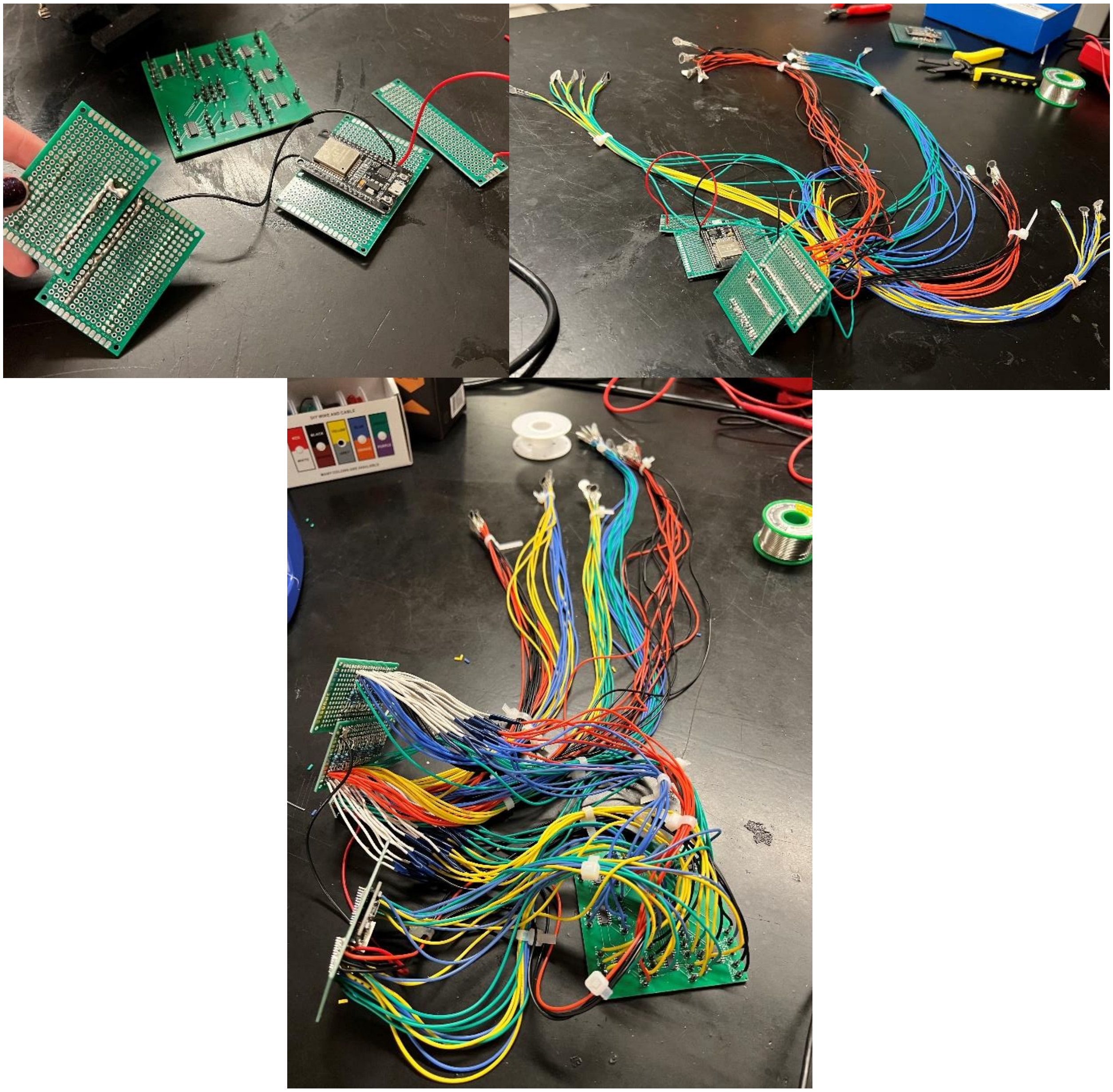

Figure 6. Electrical construction process.

As shown by Figure 6, first the reference resistors were soldered to an auxiliary protoboard and connected to the power pins of the ESP32. Next, each sensor was connected to these resistors according to the wiring diagram. Last, the appropriate connections were made to the MUX PCB and from the PCB to the ESP32. Throughout the building process, bundles of wires were organized using zip ties for wire management and to make debugging easier.

3.2.2 Mechanical Component Design

Prior to designing the housing unit, this team created a set of specifications for the design. When consulting the design of the sensor device, it was decided that the housing unit needed:

1. A volume large enough to hold all circuit boards and wiring used in the device
2. Five circular holes large enough to fit individual sensors through
3. Another hole to fit the micro-USB cable through
4. A detachable top to allow for easy access into the housing unit

With these specifications set, a design for the housing unit and a top was drafted. The 3D modeling for both pieces was accomplished using SolidWorks. Isometric views of both the housing unit and the top can be seen below in Figure 7. Complete engineering drawings of both parts can be found in Appendix B.

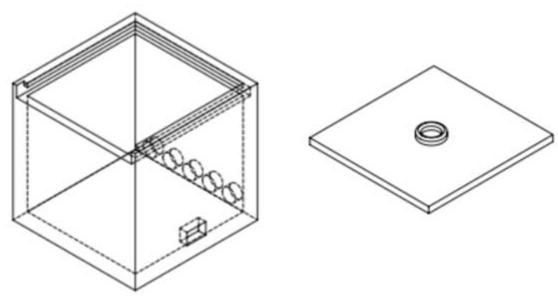

Figure 7. Visual rendering of housing enclosure body and top.

After initial designs were completed, the components were formatted in Prusa Slicer and were set to print with PLA plastic. When the electronics of the design were completed, it was found that all the circuitry and wiring would have to be packed in incredibly tightly to fit in the 3.5-inch by 3.5-inch by 3.75-inch volume inside the unit. To remedy this issue, the original design was scaled up by a factor of 1.43 times in Prusa Slicer. Once the design was finalized, it was again set to print with PLA plastic. An image of the completed box without the device inside can be seen below in Figure 8.

Figure 8. Finalized housing unit for sensing device.

3.2.3 Visual Rendering of Final Design

The following figures below show the visual rendering of the final design. Our device consisted of an ESP32 system connected to a set of sensors, with the sensors protruding out of the box enclosure as well as a USB cable that will connect to the development environment computer. This design can be seen below.

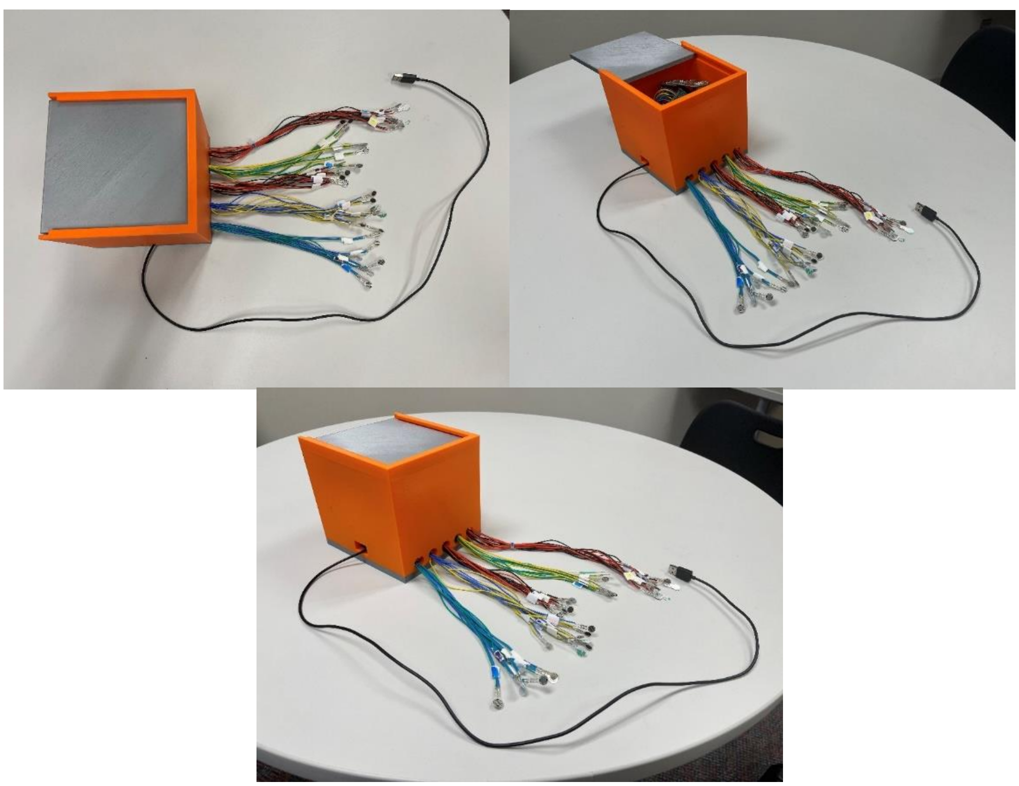

Figure 9. Construction of final design shows electronic components safely inside of box enclosure, with sensors and USB connector emerging out of box holes.

Note that the outside sensors were then arranged into a hexagonal sensor array, which followed the circle-packing optimization strategy as described in section 3.4 of the PDR. Tape was colored to each of the sensors to enhance readability when sensing – having colors allowed us to determine which sensor was which in final testing.

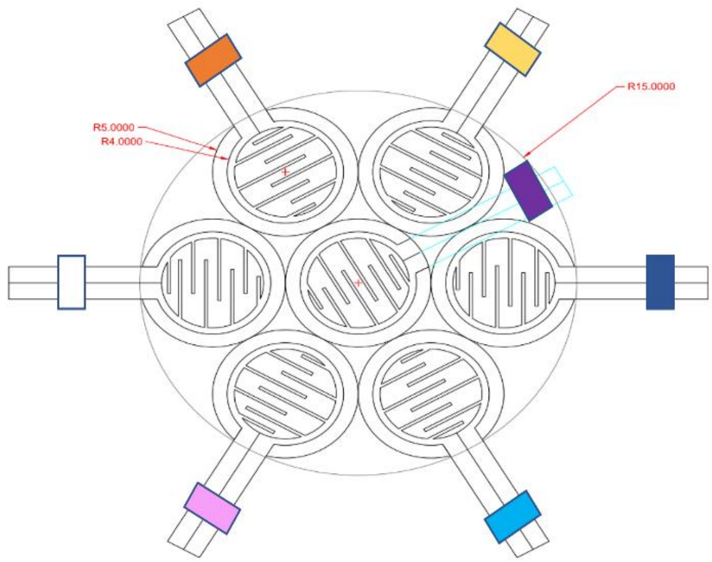

Figure 10. Final sensor packing array used in the device when testing. Note colors represent tape placed on sensors for identification when coding and analyzing data.

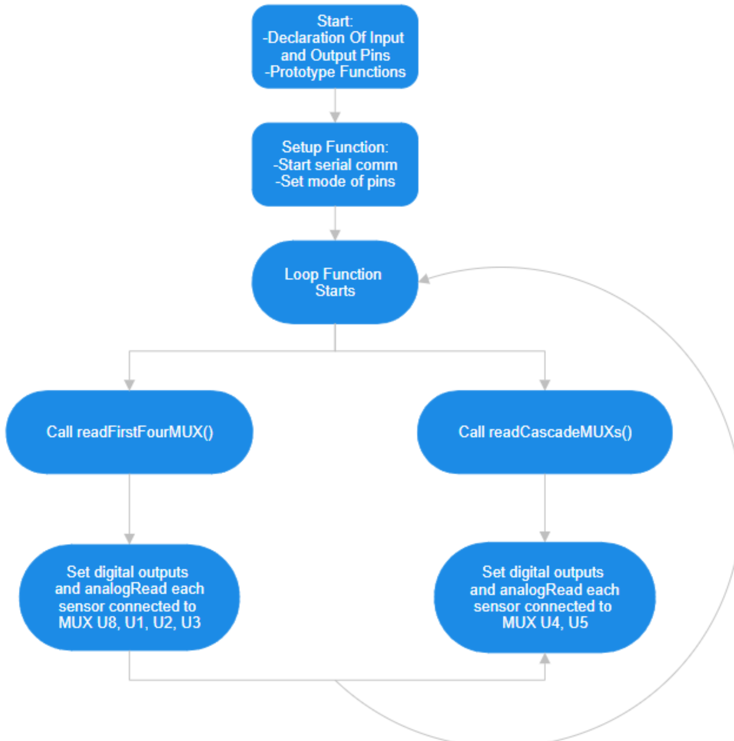

Figure 11. Code diagram for ESP32.

In summary, the ESP32 configures the MUXs and takes a measurement for each sensor in each iteration of the loop. It does this by controlling the output pins connected to the MUXs. This selects each sensor to be passed to its corresponding ADC pin. This loop runs continuously, so the sensors are continually read. These reading are broken into functions for reading the standard and cascaded MUXs. Each of these readings are written over serial communication to the external PC for interpretation. The full code can be found in Appendix E.

4 Results and Discussion

4.1 Test Results

To ensure proper functioning of the design, a sensor press test, a weight scalability test, and a test representative of a clinical test were performed. The results of such tests are outlined in the sections below.

4.1.1 Sensor Press Test

This test was performed to ensure proper function of the individual sensors in each array. During the test, each sensor was pressed in between two fingers, and the presence of an output was observed. Given proper device functionality, each sensor should output a value. The results of this test can be seen in Table 3 below. A comprehensive table of pin outputs of each sensor can be found in Appendix C, Table 7.

Table 3. Results of sensor press test demonstrate a functional array.

Sensor Color Designation	Lateral Condyle Array	Medial Condyle Array	Fibular Head Array	Tibial Tuberosity Array	Bony Protrusion Array
Orange Sensor	✓	✓	✓	✓	✓
Pink Sensor	✓	✓	✓	✓	✓
Yellow Sensor	✓	✓	✓	✓	✓
Dark Blue Sensor	✓	✓	✓	✓	✓
Light Blue Sensor	✓	✓	✓	✓	✓
Purple Sensor	✓	✓	✓	✓	✓
White Sensor	✓	✓	✓	✓	✓

4.1.2 Weight Scalability Test

A prior testing analysis showed a positive correlation between mass and code output. We tested each sensor in each array at masses of 62, 162, 362, and 562 grams using a weightbearing device and weights on top of the device, as seen in Figure 11.

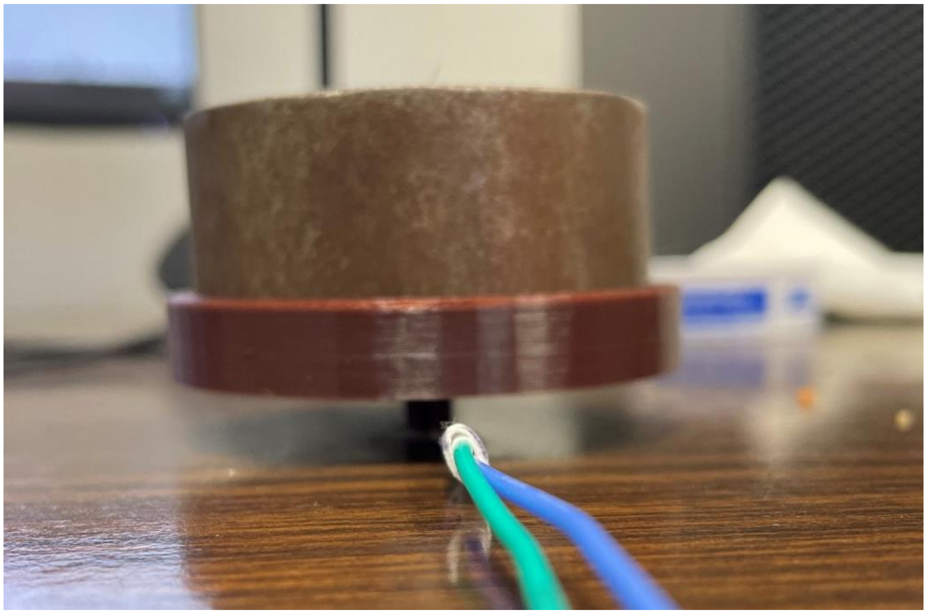

Figure 12. Weight bearing device on top of sensor with standardized weight to measure weight scalability. Weightbearing device is 3D printed and shown in Appendix A.

This test confirms that these sensors can be used as a first impression measurement for pressure analysis. A graphical representation of the results of this test can be seen below in Figure 12 while the raw data obtained in the test can be found in Appendix D.

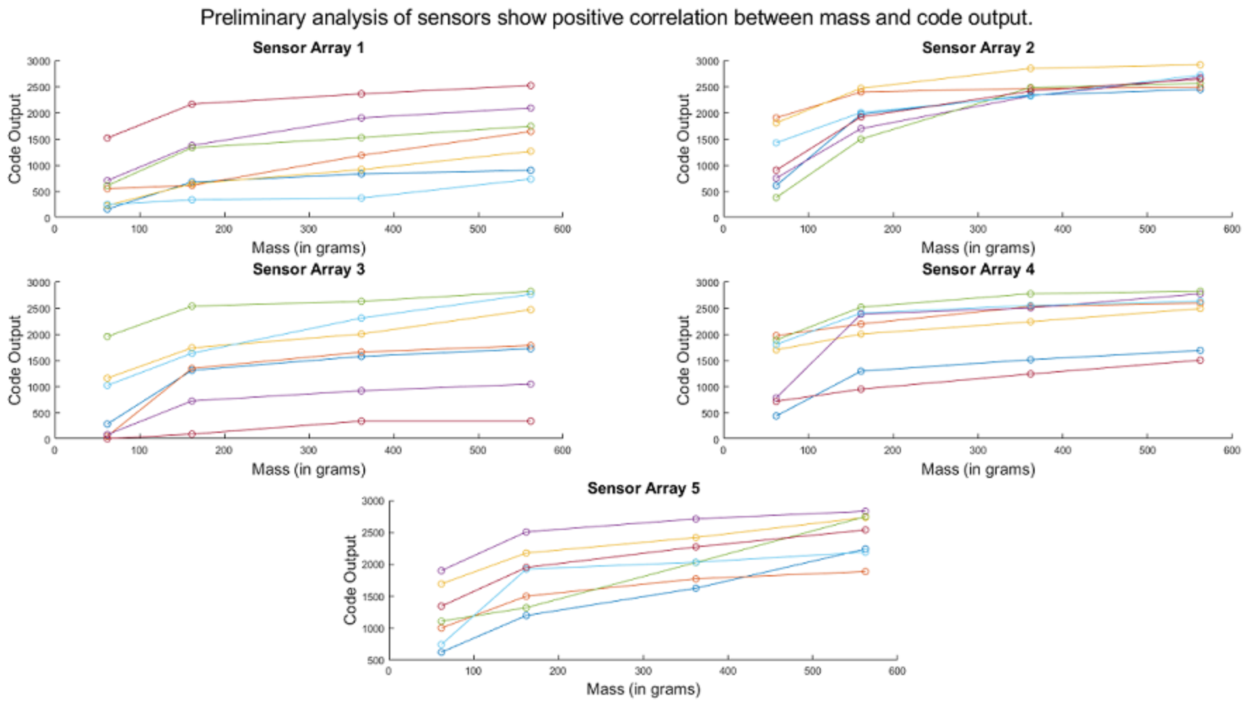

Figure 13. Graphical representation of data obtained from weight scalability test.

Note that in the above figure, orange represents data from orange sensor, yellow represents data from yellow sensor, purple represents data from purple sensor, light blue represents data from light blue sensor, dark blue represents data from dark blue sensor, red represents data from pink sensor, and green represents data from white sensor.

4.1.3 Clinical Test

A final test was performed to test the sensors on a human participant. This test was originally planned to use an amputee as the participant, but the individuals asked declined to participate. As a result, a modified socket was used on Chirayu to test the sensors on a human participant. This test served multiple purposes:

1. Test the efficacy of the sensor packing to adhere to a participant's leg.
2. Test to see whether all sensors could respond together when pressure was applied to the area of interest.
3. Test if 50 PSI was measurable in the given sensor array. This was conducted by having a research assistant press as hard as possible onto a sensor on the patient's leg, and checking if data output was below the maximum output value of 4095.

See the below figure for sensor attachment implementation.

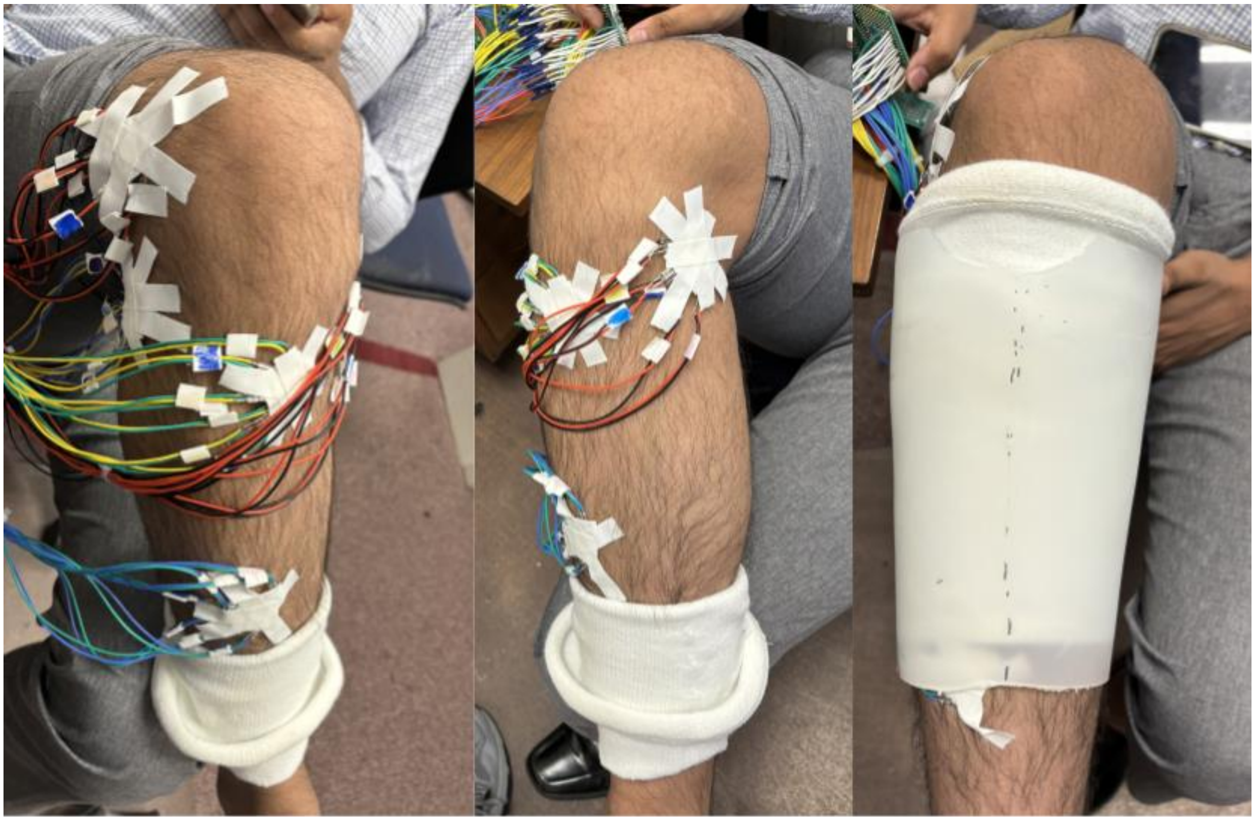

Figure 14. Sensor attachment to patient. Sensors placed in hexagonal array on points of interest (left and middle image), before being covered with sock and socket for testing (right image).

The results of the adherence portion can be seen below in Table 4 while the results of the sensor respondence can be seen in Table 5.

Table 4. Adherence of arrays to bony landmark is confirmed.

Sensor Location:	Adherence:
Tibial Tuberosity Array	✓
Fibular Head Array	✓
Lateral Condyle Array	✓
Medial Condyle Array	✓
Bony Protrusion Array	✓

Table 5. Number of sensor readouts in each array at bony landmarks of interest.

Sensor Location:	Number of Sensor Readouts in Array
Tibial Tuberosity Array	7/7
Fibular Head Array	7/7
Lateral Condyle Array	7/7
Medial Condyle Array	7/7
Bony Protrusion Array	7/7

Note that the maximum pressure value recorded during the maximal press test was approximately 3200, which is less than 4096. We anticipate this pressure to be far greater than 50 psi (average grip strength for men far exceeds that value). Therefore, we anticipate that this device is effective up to and including maximal safe socket pressure values. This data, as well as the conditional formatting in Excel for easier reading, is seen in the figure below.

U53-1	2730
U53-2	2830
U53-3	3750
U61-1	3500
U61-2	3200
U61-3	3100
U81-1	0
U81-2	0
U81-3	0

Figure 15. Screenshot of Excel spreadsheet demonstrating that manual press test is both functional and shows press location under red highlight using conditional formatting.

4.2 Other Factors to Consider

This device is medical in nature; as such, it is important to recognize that device use must be conducted in a safe and sterile fashion. When using this device, clinicians should don gloves and sanitize the areas where testing will occur. We recommend that 3% hydrogen peroxide should be used when sterilizing the testing site, as hydrogen peroxide has both antibacterial and antifungal properties, both of which are potential areas of concern for amputees. For further safety, we recommend that post testing, all sensors are cleaned using a spray solution, and all tape is removed and disposed of – tape may hold residual hairs and skin cells from patients, and we wish to minimize patient-to-patient contamination.

For electronic safety, our team recommends that the electronic boards of the device remain inside of the box enclosure when not in use or during fitting. Water seepage may compromise the integrity of the device, posing potential electrocution risks during patient testing.

IRB approval should be provided when conducting patient testing. Our team was under an IRB provision given by Dr. Ha Van Vo, but other clinicians should request IRB approval before conducting experiments with the device.

When building the device, we also recommend using unleaded solder and operating in a well-ventilated room, to minimize solder fumes. We also recommend prior training on using soldering irons and other electrical equipment before device manufacturing to minimize injury.

4.3 Costs

To ensure satisfaction of feasibility criteria, the project budget was tracked throughout. A breakdown of the costs associated with production of the final design can be seen below in Table 6.

Table 6. Component costs for design.

Item	Individual Cost	Allocation	Total Cost
ESP32 Microcontroller	$5.30	1	$5.30
10 kiloohm Resistors (100ct)	$5.49	1	$5.49
Lead-free Solder	$18.99	1	$18.99
Solder Protoboards	$9.99	1	$9.99
Waterproof First-Aid Tape	$10.95	1	$10.95
USB-A to Micro USB Cable	$7.68	1	$7.68
Custom Printed Circuit Board	$14.25	1	$14.25
Force-Sensitive Resistors (2ct)	$7.99	3	$23.97
Force-Sensitive Resistors (2ct)	$8.19	1	$8.19
Force-Sensitive Resistors (2ct)	$8.72	7	$61.04
Force-Sensitive Resistors (2ct)	$10.99	7	$76.93
3D Printing Costs	$10.00	1	$10.00
Wiring Kits	$12.99	2	$25.98
		Total Cost	**$274.56**

While all FSRs were of the same model and manufacturer, time of purchase varied, which led to price fluctuations. These fluctuations are reflected in the above table.

5 Summary and Conclusions

5.1 Project Summary

At its onset, this team was tasked by Dr. Ha Van Vo, Distinguished Professor of Biomedical Engineering at Mercer University, to create a device to measure pressure at the socket-stump interface. After much research and drafting of potential designs, this team settled on a device that utilized 7 force sensitive resistors in a hexagonal pattern coupled with basic voltage dividers to sense pressure values at key bony landmarks on an amputee's stump. An ESP32 microcontroller was then used to process these signals and convert them to discrete values.

Finally, code was written to transfer these values into a Microsoft Excel spreadsheet, which displayed a color indicating the acceptability of the pressure measurements.

After construction of the device concluded, a base test indicated that each sensor would output a value representative of pressure to a pin. While the weight scaling test indicated that these values varied for each sensor, a general trend was evident in that higher masses corresponded with higher readouts from each of the sensors. A final test on Chirayu showed safe adherence and acceptable function of the device on a human participant. With the completion of this project, the team is ready to turn over this prototype device to Dr. Vo (for improvement in the future).

5.2 Recommendations for Future Work

Given the small budget and timeframe for this project, many different aspects of this design can be improved for the future. Firstly, to make the design more convenient, the design should be able to connect to a computer via a wireless connection. Given a lack of resources and the status of this design as a prototype, this team was focused on providing a functional design. Due to such parameters, this team opted for the usage of a wired design to ensure consistent functionality. However, future iterations of this design should seek to incorporate wireless connectivity to make the design more usable in areas where there is not a computer in the immediate vicinity.

In addition to the wireless capabilities, new sensors should be selected for this design. When initially developing the design, force-sensitive resistors provided a cost-effective method for measuring pressure values. When testing the finished product, the trend of increasing values with increasing masses showed promise, but the values output by the sensors were variable. There is research stating that FSR's are not optimal for prosthetic measurements, and capacitive sensors can be used as more accurate weight bearing measurement tools (Swanson et al., 2019). Given this, more accurate and consistent sensors, such as capacitive sensors, should be chosen for future iterations.

Another area of improvement is the number and size of sensors present in each array. When testing, the 7 sensors with a circular (4-millimeter radius) sensing area led to the generation of a comprehensive set of data values, there is always room for improvement. The usage of sensors with smaller sensing areas would allow for more accuracy in identifying where excess pressure is located at a landmark. Additionally, more sensors would allow for the array to cover a larger area around the bony landmarks, which would lead to the generation of a more encompassing set of data values. Both changes would lead to the development of a more accurate design but would most likely require a budget far exceeding the budget allotted for this project.

Yet another area of improvement would be the development of a better, preferably reusable, backing. Given the limitation on budget, the team used skin-safe first-aid tape commonly found in first-aid kits. While the tape could hold the sensors in the array formation on their own, when adhering to the human subject during clinical testing, the sensors would detach from the backing. Additionally, after some time, the adhesive on the tape would fail and the array would not adequately adhere to the subject. To remedy these issues, a more permanent backing that secures the sensors in the array formation should be designed. In addition to this, a better, biocompatible adhesive should be used securely attach the sensor arrays to the bony landmarks outlined above.

The final recommendation for improvement of this design would be integration of resistors used in voltage dividers to a custom printed circuit board. While the method of using solder to create the circuit worked, it made the circuitry messy and cumbersome. Integrating the resistors into a printed circuit board would clean up some of the messy wiring present in this prototype.

References

ESP32 Pinout Reference: Which GPIO pins should you use? (n.d.). https://randomnerdtutorials.com/esp32-pinout-reference-gpios/

Hicks, C. W., & Selvin, E. (2019). Epidemiology of peripheral neuropathy and lower extremity disease in diabetes. *Current Diabetes Reports*, *19*(10), 86. https://doi.org/10.1007/s11892-019-1212-8

Johnson, S., McGraw, P., Salgarkar, C. (2023) Preliminary design review for a sensor array to measure pressure at the socket-stump interface. *Mercer School of Engineering.*

Pearson, J. R., Holmgren, G., March, L., & Oberg, K. (1973). Pressures in critical regions of the below-knee patellar-tendon-bearing prosthesis. *Bulletin of Prosthetics Research*, *10*(19), 52–76.

Swanson, E. C., Weathersby, E. J., Cagle, J. C., & Sanders, J. E. (2019). Evaluation of force sensing resistors for the measurement of interface pressures in lower limb prosthetics. *Journal of Biomechanical Engineering*, *141*(10), 1010091–10100913. https://doi.org/10.1115/1.4043561

Walicka, M., Raczyńska, M., Marcinkowska, K., Lisicka, I., Czaicki, A., Wierzba, W., & Franek, E. (2021). Amputations of lower limb in subjects with diabetes mellitus: Reasons and 30-day mortality. *Journal of Diabetes Research*, *2021*, 1–8. https://doi.org/10.1155/2021/8866126

Appendix A: Schematic for weight bearing device

When performing the weight scaling test, all weight needed to be concentrated on the sensing area. To accommodate, the apparatus below was designed.

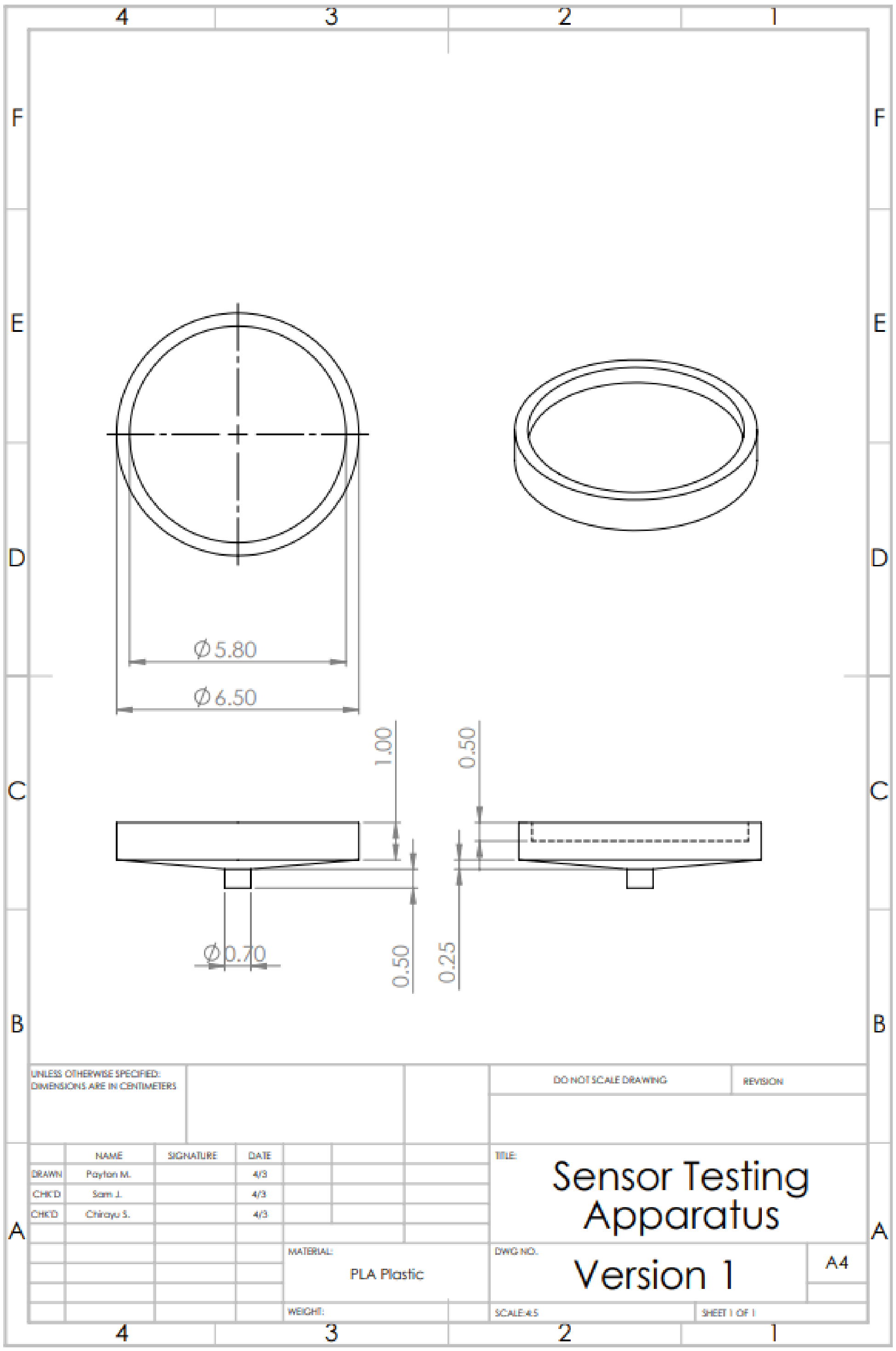

Figure 16. Schematic of sensor testing apparatus which bears weight.

Appendix B: Schematics for housing units

The following pages will include schematics for the housing unit for the circuitry of the design. Initial printing yielded a design that was too small, so both parts were scaled by a factor 1.43.

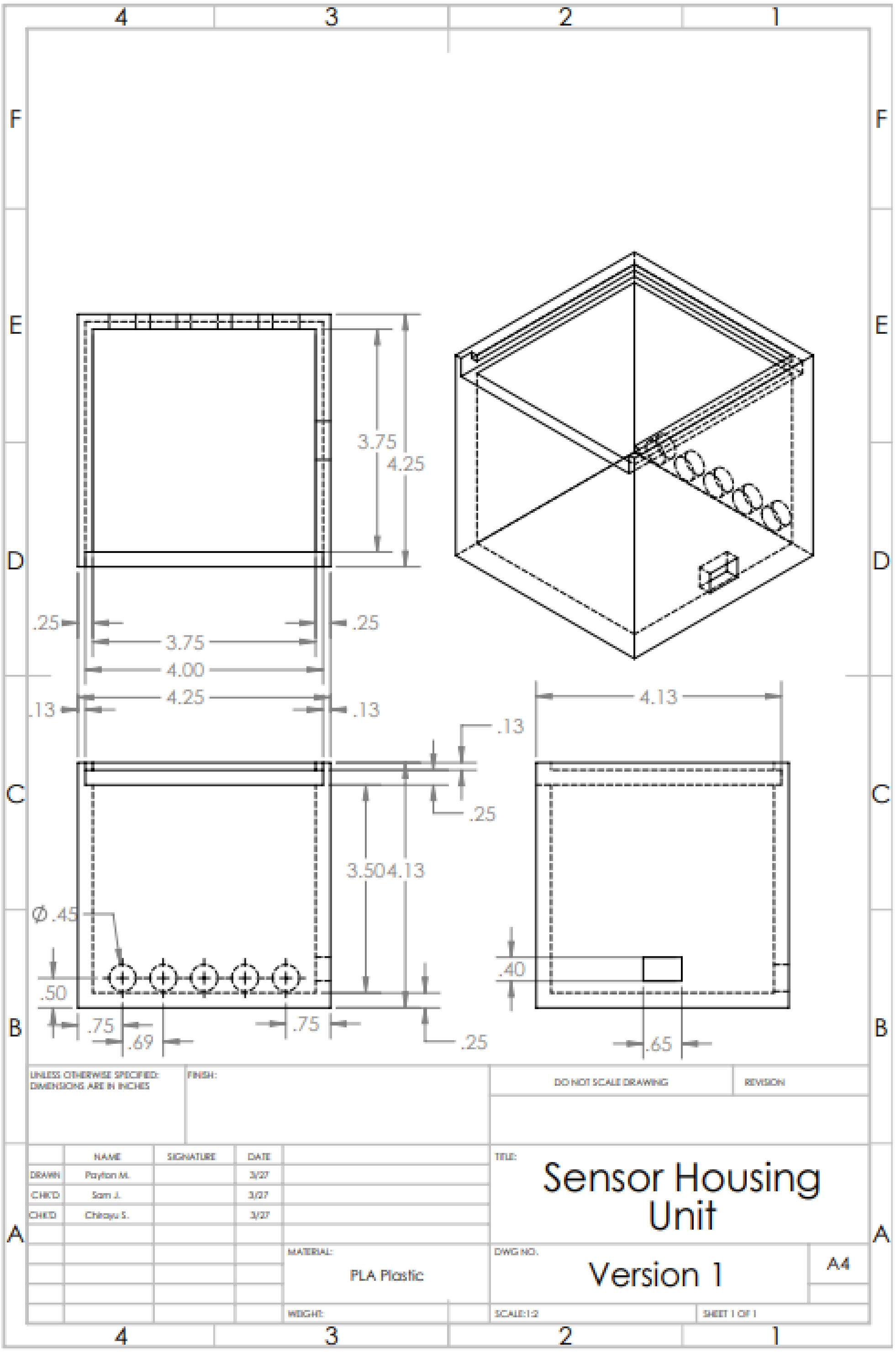

Figure 17. Sensor housing unit diagram.

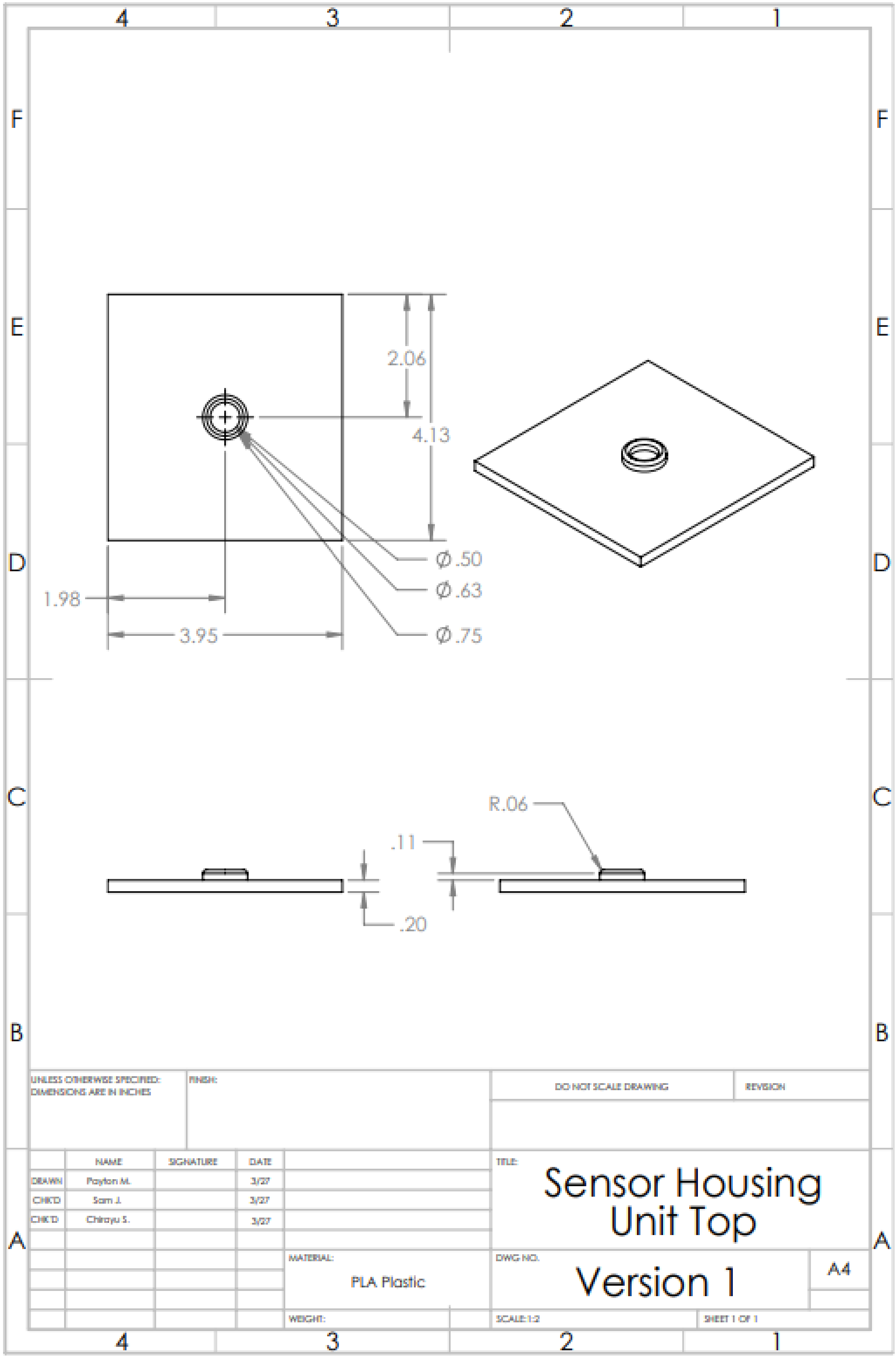

Figure 18. Sensor housing top diagram.

Appendix C: Output pins to sensor analysis

When testing each of the functionality of the individual sensors of the array, the output pin for each sensor was noted. Given the proper function of the design, each sensor should correspond to a single, unique output pin. A list of the output pins for each sensor can be seen in the table below.

Table 7. Output pin to sensor table.

Sensor Color Designation	Lateral Condyle Array	Medial Condyle Array	Fibular Head Array	Tibial Tuberosity Array	Bony Protrusion Array
Orange Sensor	U:6, 4	U-2:2, 1	U-5, 2	U-2:2, 2	U-1:1, 1
Pink Sensor	U-3:2, 3	U-2:1, 1	U-5, 5	U-2:1, 3	U-8:1, 3
Yellow Sensor	U-3:1, 3	U-1:2, 2	U-6, 3	U-2:2, 3	U-8:2, 3
Dark Blue Sensor	U-3:2, 2	U-1:2, 3	U-5, 3 & U-5, 6	U-3:2, 1	U-8:2, 1
Light Blue Sensor	U-6, 2	U-1:1, 3	U-5, 1	U-3:1, 2	U-8:1, 1
Purple Sensor	U-6, 1	U-1:2, 1	U-6, 3	U-3:1, 1	U-8:2, 2
White Sensor	U-6, 5	U-1:1, 2	U-5, 4	U-2:1, 2	U-8:1, 2

The codes listed above represent the pin output location. The number following the U- represents the general pin the output was located on. The number following the colon represents the subsection of the pin the output was located on if such a subsection exists. Finally, the number following the comma represents the position of the output on that pin section or subsection.

An important thing to note is that in one array, two sensors output to one pin and one sensor output to two different pins. This was suspected to be caused by a slight mix-up in wiring, but when testing, this mix-up did not present a serious issue. When observing the dark blue sensor, the team did all recordings with respect to the U-5, 3 position, meaning this should be the observed output pin when looking at pressures at this position on the landmark. Additionally, even though two sensors output to the same pin, the output read the highest pressure experienced between the two sensors, meaning that if the two sensors were placed next to each other on the array, both areas could be adjusted according to that highest value. Thus, while the device was not working as intended in this case, the team was able to work around it and provide a functional prototype.

Appendix D: Raw data for weight scalability testing

Table 8. Raw data table, graphical representation seen in Figure 12.

Sensor \ Mass	62 grams	162 grams	362 grams	562 grams
TT Orange	625	1200	1628	2243
TT Yellow	1005	1503	1775	1889
TT Purple	1696	2179	2423	2738
TT Light Blue	1904	2511	2715	2834
TT Dark Blue	1112	1323	2031	2751
TT Pink	745	1927	2035	2193
TT White	1346	1953	2275	2541
FH Orange	158	676	831	898
FH Yellow	549	609	1184	1639
FH Purple	227	642	912	1262
FH Light Blue	704	1373	1899	2089
FH Dark Blue	607	1331	1523	1741
FH Pink	247	336	369	730
FH White	1515	2162	2358	2518
LC Orange	282	1309	1572	1720
LC Yellow	47	1344	1655	1781
LC Purple	1157	1737	2000	2464
LC Light Blue	80	723	917	1045
LC Dark Blue	1953	2533	2623	2815
LC Pink	1019	1634	2304	2759
LC White	0	90	337	337
MC Orange	607	1970	2335	2444
MC Yellow	1900	2393	2461	2485
MC Purple	1803	2464	2844	2913
MC Light Blue	748	1695	2317	2672
MC Dark Blue	378	1494	2478	2559
MC Pink	1423	2002	2317	2719
MC White	900	1918	2411	2642
BP Orange	435	1290	1510	1687
BP Yellow	1970	2191	2527	2590
BP Purple	1698	1999	2235	2486
BP Light Blue	779	2379	2499	2768
BP Dark Blue	1885	2512	2770	2817
BP Pink	1800	2400	2544	2627
BP White	715	947	1238	1500

Note that TT refers to Tibial Tuberosity, FH refers to Fibular Head, LC refers to Lateral Condyle, MC refers to Medial Condyle, and BP refers to Bony Protrusion at bottom of stump.

Appendix E: Code for the ESP32 To take readings, send data over serial

Attached below is the main driver code for the ESP 32 output reading.

src\main.cpp

```
/*
 * This ESP32 code is created by esp32io.com
 *
 * This ESP32 code is released in the public domain
 *
 * For more detail (instruction and wiring diagram)\n visit https://esp32io.com/tutorials/esp32-force-sensor
 */
#include <Arduino.h>


//**************************SENSOR READING PINS*********************************
#define FORCE_SENSOR_PIN_X1 36 // ESP32 pin GPIO36 (ADC0): the FSR and 10K pulldown are connected to A0
#define FORCE_SENSOR_PIN_Y1 39
#define FORCE_SENSOR_PIN_Z1 34
#define FORCE_SENSOR_PIN_X2 35
#define FORCE_SENSOR_PIN_Y2 32
#define FORCE_SENSOR_PIN_Z2 33
#define FORCE_SENSOR_PIN_X3 25
#define FORCE_SENSOR_PIN_Y3 26
#define FORCE_SENSOR_PIN_Z3 27
#define FORCE_SENSOR_PIN_X4 14
#define FORCE_SENSOR_PIN_Y4 12
#define FORCE_SENSOR_PIN_Z4 13
#define FORCE_SENSOR_PIN_X5 15
#define FORCE_SENSOR_PIN_Y5 2
#define FORCE_SENSOR_PIN_Z5 4

//**************************CONTROL PINS*********************************
#define OUTPUT_PIN_ALL1 16

#define OUTPUT_PIN_A2 17
#define OUTPUT_PIN_B2 19
#define OUTPUT_PIN_C2 22

#define OUTPUT_PIN_A3 5
#define OUTPUT_PIN_B3 21
#define OUTPUT_PIN_C3 23

void readCascadedMUXs();

void readFirstFourMUX();
void readU8();
void readU1();
void readU2();
void readU3();

void setup() {

  //CONFIGURING DIGITAL OUTPUT PINS
  pinMode(OUTPUT_PIN_ALL1, OUTPUT);

  pinMode(OUTPUT_PIN_A2, OUTPUT);
```

```
  pinMode(OUTPUT_PIN_B2, OUTPUT);
  pinMode(OUTPUT_PIN_C2, OUTPUT);
  pinMode(OUTPUT_PIN_A3, OUTPUT);
  pinMode(OUTPUT_PIN_B3, OUTPUT);
  pinMode(OUTPUT_PIN_C3, OUTPUT);


  //STARTING SERIAL COMMUNICATION
  Serial.begin(9600);

  while(!Serial);

}

void loop() {
  readFirstFourMUX();
  readCascadedMUXs();
}

void readFirstFourMUX() {
  readU8();
  readU1();
  readU2();
  readU3();
}

//reading X5, Y5 Z5 for ALL 11 channels
void readCascadedMUXs() {
  int reading;

  delay(10);

  //configure to read sensors connected to U5
  Serial.print("U5 \n");

  digitalWrite(OUTPUT_PIN_A3, LOW);
  digitalWrite(OUTPUT_PIN_B3, LOW);
  digitalWrite(OUTPUT_PIN_C3, LOW);

  digitalWrite(OUTPUT_PIN_A2, LOW);
  digitalWrite(OUTPUT_PIN_B2, LOW);
  digitalWrite(OUTPUT_PIN_C2, LOW);

  //READING XO-5, Y0-5, Z0-5
  reading = analogRead(FORCE_SENSOR_PIN_X5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z5);
  Serial.print(reading);
  Serial.print(":\n");

  delay(10);

```

```
  digitalWrite(OUTPUT_PIN_A2, HIGH);
  digitalWrite(OUTPUT_PIN_B2, HIGH);
  digitalWrite(OUTPUT_PIN_C2, HIGH);

  //READING X1-5, Y1-5, Z1-5
  reading = analogRead(FORCE_SENSOR_PIN_X5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z5);
  Serial.print(reading);
  Serial.print(":\n");

  //configure to read sensors connected to U6
  Serial.print("U6 \n");

  digitalWrite(OUTPUT_PIN_A3, HIGH);
  digitalWrite(OUTPUT_PIN_B3, HIGH);
  digitalWrite(OUTPUT_PIN_C3, HIGH);

  digitalWrite(OUTPUT_PIN_A2, LOW);
  digitalWrite(OUTPUT_PIN_B2, LOW);
  digitalWrite(OUTPUT_PIN_C2, LOW);

  //READING XO-6, Y0-6, Z0-6
  reading = analogRead(FORCE_SENSOR_PIN_X5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z5);
  Serial.print(reading);
  Serial.print(":\n");

  delay(10);

  digitalWrite(OUTPUT_PIN_A2, HIGH);
  digitalWrite(OUTPUT_PIN_B2, HIGH);
  digitalWrite(OUTPUT_PIN_C2, HIGH);

  //READING X1-6, Y1-6
  reading = analogRead(FORCE_SENSOR_PIN_X5);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y5);
  Serial.print(reading);
  Serial.print(":\n");
}


//reading X1, Y1 Z1 for both channels
void readU8() {
  int reading;
```

```
  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, LOW);

  Serial.print("U8 - 1 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X1);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y1);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z1);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, HIGH);

  Serial.print("U8 - 2 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X1);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y1);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z1);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

}

//reading X2, Y2 Z2 for both channels
void readU1() {
  int reading;

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, LOW);

  Serial.print("U1 - 1 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X2);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y2);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z2);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, HIGH);

```

```
  Serial.print("U1 - 2 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X2);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y2);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z2);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

}

//reading X3, Y3 Z3 for both channels
void readU2() {
  int reading;

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, LOW);

  Serial.print("U2 - 1 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X3);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y3);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z3);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, HIGH);

  Serial.print("U2 - 2 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X3);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y3);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z3);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

}

//reading X4, Y4 Z4 for both channels
void readU3() {
  int reading;

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, LOW);
```

```
  Serial.print("U3 - 1 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X4);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y4);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z4);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

  delay(10);

  digitalWrite(OUTPUT_PIN_ALL1, HIGH);

  Serial.print("U3 - 2 \n");
  reading = analogRead(FORCE_SENSOR_PIN_X4);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Y4);
  Serial.print(reading);
  Serial.print(":\n");
  reading = analogRead(FORCE_SENSOR_PIN_Z4);
  Serial.print(reading);
  Serial.print(":\n--------------\n");

}
```

Appendix F: ESP32 pinout information

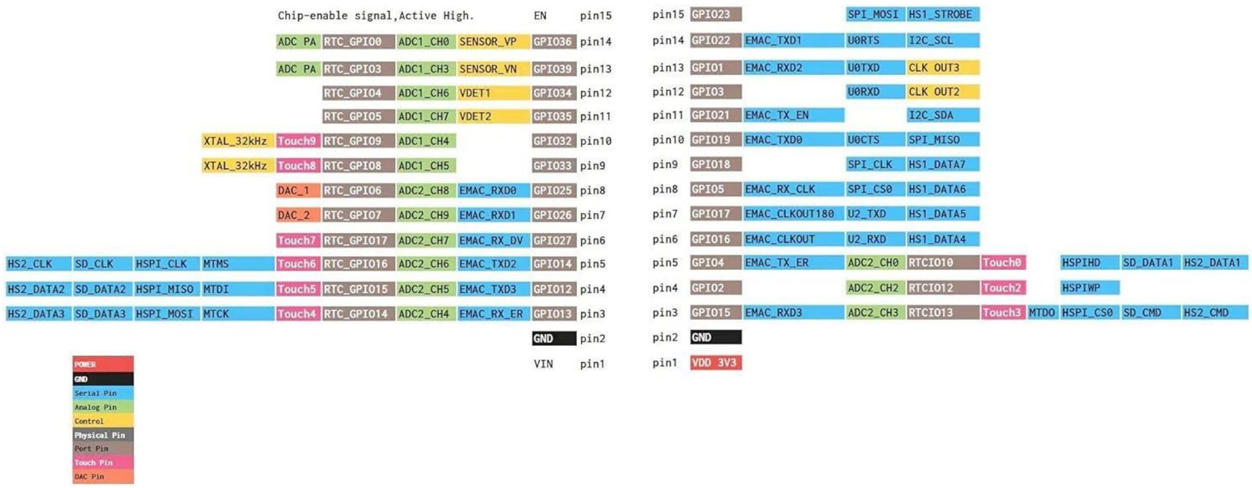

Figure 19. Diagrammatic view of ESP32 pinout.

Appendix G: Operation instructions

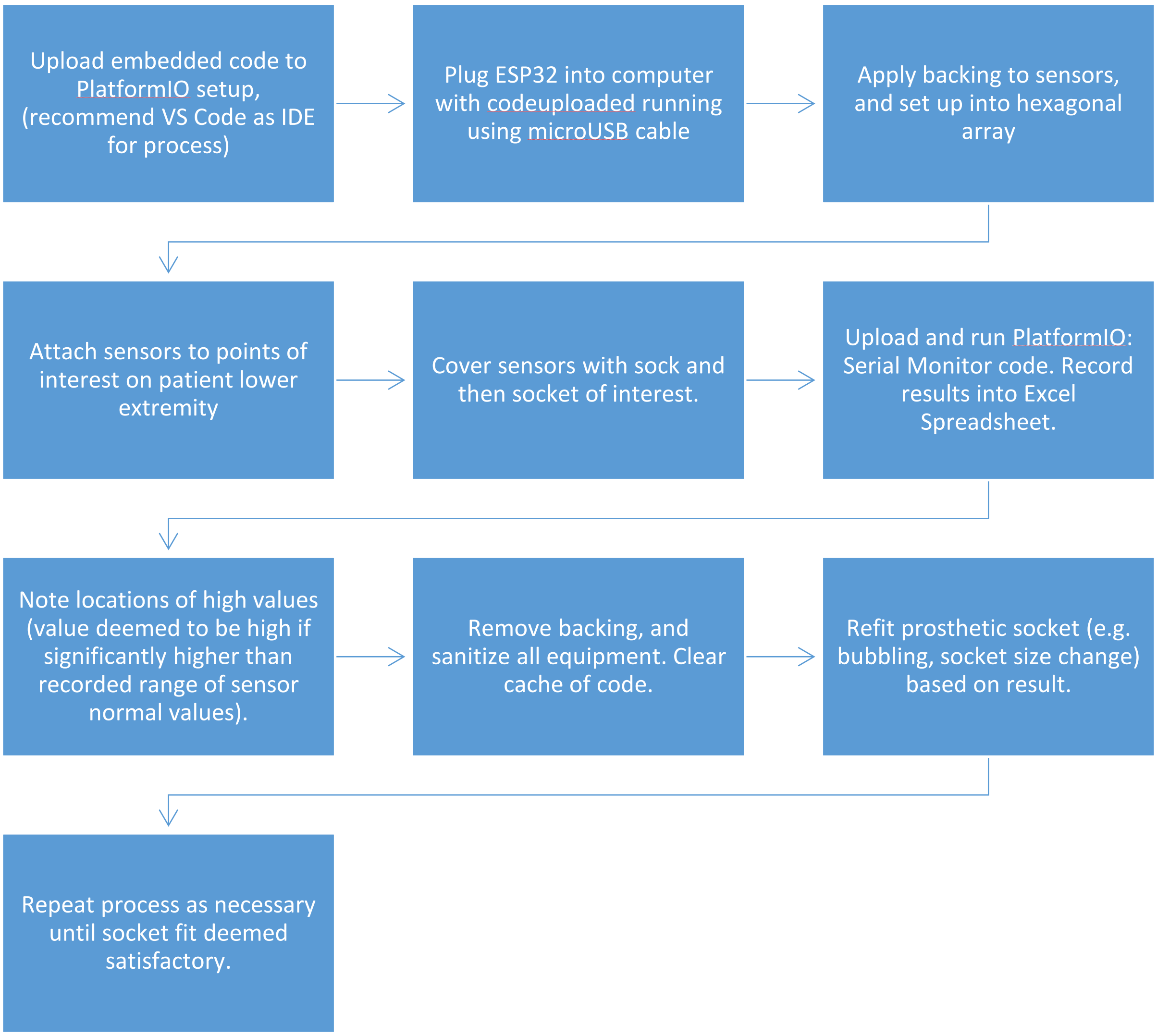

Figure 20. Operation instructions for clinicians using device.

Appendix H: Pinout of Analog Multiplexers

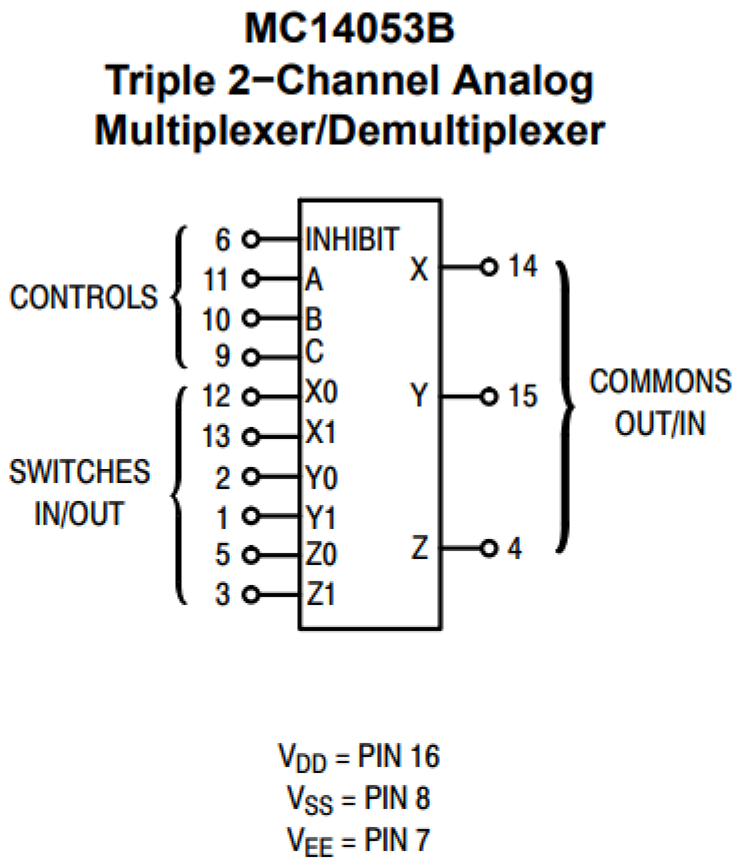

Figure 21. Analog multiplexer pinout diagram.

Appendix I: Annotated test plans

Sensor Press Test
Test Objective: Test the function of each of the FSRs
Equipment Needed: Completed design
Location(s): Prosthetic Lab
Date(s)/Total Time: 1 hour
Personnel: Payton McGraw and Chirayu Salgarkar
Criteria for success: Each sensor reads a data value to an output pin
Procedure:

1. Download the code for the design and plug the design into a computer via a microUSB cable.
2. Apply pressure to a single FSR
3. Record whether a signal was output and where the signal output to
4. Repeat until all FSRs have been tested.

Weight Scalability Test
Test Objective: Conduct a weight scaling for each sensor to output a range of values over
Equipment Needed: Completed design, apparatus to concentrate weight onto sensor area, 50g weight, 100g weight, 200g weight, and 500g weight.
Location(s): Prosthetic Lab
Date(s)/Total Time: 3 hours
Personnel: Payton McGraw and Chirayu Salgarkar
Criteria for success: Observe a positive correlation between increasing mass applied to sensor area and output created by the sensor.
Procedure:

1. Download the code for the design and plug the design into a computer via a microUSB cable.
2. Position the apparatus over the center of the FSR.
3. Place the 50-gram weight onto the apparatus and record the output
4. Place the 100-gram weight on top of the 50-gram weight already on the apparatus and record the output
5. Place the 200-gram weight on top of the 50-gram and 100-gram weights and record the output.
6. Remove both the 100-gram and 200-gram weights from the apparatus and place the 500-gram weight on top of the 50-gram weight. Record the value.
7. Repeat this process until each sensor has been tested on.

Clinical Test
Test Objective: To observe whether the adhesive used in the array backing adheres to patient and to observe values captured in a clinical environment. Additionally, this test was conducted to determine whether sensors could handle 50psi without hitting max sensor output value.

Equipment Needed: Completed design, white medical tape, patient, prosthetic sleeve, and prosthetic socket
Location(s): Prosthetic Lab
Date(s)/Total Time: 2 hours
Personnel: Payton McGraw and Chirayu Salgarkar
Criteria for success: Observe successful adhesion to patients during testing period and observe realistic values during pressure test on each array. When applying 50 psi pressure to an area, the output of the sensors in an array should not hit the max value of approximately 4000
Procedure:

1. Download the code for the design and plug the design into a computer via a microUSB cable.
2. Apply the backing to each of the sensors to form the hexagonal array.
3. Place the arrays at the points of interest on the stump
4. Place the sleeve over the arrays and put the socket over the sleeve.
5. Observe values when applying pressure to areas of interest. Note outliers.