Project Smart Shoes

By: Tyler Rupp

IEEE Susquehanna Section Regional Capstone competition

Abstract

Many people in America have access to a body scale and a doctor that can help them improve their health. Despite this access, people tend to only check their weight and go to the doctors a few times a year. This inattentiveness to their health can lead to obesity and other unknown health issues. For example, an individual may be placing more weight on one foot when walking. Over time, the uneven weight distribution may lead to back problems or the risk of falling due to imbalance. The issue can be solved by giving people a simpler, more frequent, and traceable way to measure their weight and balance.

This is now possible with Smart Shoes that can measure weight and weight distribution. Smart Shoes are designed using an Arduino nano 33 IoT microcontroller for its Wi-Fi and Bluetooth capabilities. Weight and Weight distribution in the shoes are measured by Wheatstone bridge modules that measure weight from the flex of the module. The Smart Shoes also have GPS capabilities and internal memory storage to track distance and store data from the user. Smart Shoes were developed to regularly gather weight and weight distribution data anywhere the user wears the shoes. The ability to measure weight is validated against an average household digital scale. At the conclusion of the development and testing processes, the shoes were able to measure weight within 10% of a digital scale. The shoes can measure overall weight and the weight on each shoe which can be used to tell if the user is balanced or not. The future hope is that these shoes can allow people to improve their balance, measure their weight more regularly, help people improve their stride, and track distance without their phone.

Table of Contents

Table of Figures

Table of Tables

Table of Equations

1. Introduction

A. Introduction

Mary was an older adult who lived a normal life until she fell walking down the steps of a bus. She fractured her hipbone and scraped her face and body. She recovered quickly but the experience remained with her. The impact of her fall limited her activity because she was afraid that she might fall again. Suddenly her normal life was more challenging. Activities such as going to the movies or riding the bus weren't possible because of the fear of falling. Her fall and injuries had a huge impact on her quality of life.

According to the Centers for Disease Control and Prevention, about one-third of adults over 65 fall each year. This is primarily because balance tends to start declining midlife. Despite the decline, balance can be improved, and falls prevented. Insoles exist that can be inserted into shoes that measure gait and let the user know how to improve their balance [9]. The biggest issue with this method is that these insoles use pressure sensors to measure gait. No data is recorded using weight distribution, only pressure. While this may be helpful in determining what parts of the foot the induvial uses more, it tells nothing in terms of weight distribution. That is why the Smart Shoes were created. These shoes use weight sensors to measure weight and weight distribution. Users can adjust their stride so that they walk with half of their weight on the right leg and half of their weight on the left leg. Since weight is being recorded, this can also help individuals track their weight regularly and get to the weight they desire to be. The Smart Shoes use an Arduino Nano 33 IoT microcontroller to process the data. They also use a GPS to track how far someone has walked and has a micro-SD card to save user data. All these devices are used inside a pair of shoes to help people improve their health, and specifically, for older adults to prevents falls.

B. Human Impact

The Smart Shoes have a positive impact on public health. Today, not many people regularly check their weight. This ignorance can contribute to the prevalence of obesity in America. Since the Smart Shoes can measure weight and are worn every day by the user, they help to increase the user's awareness of their weight. This knowledge can help individuals get to their desired weight whether it be gaining or losing weight. Another public health impact is with the shoes' ability to track distance and steps. While smart phones and devices can do this already, this gives users the ability to take a step away from their smart phones and simply use their shoes to measure steps and distance. The Smart Shoes can also more accurately measure steps compared to how smart phones calculate steps. The Shoes measure the change in weight on the sensors with each step. A phone can only measure a step when the phone detects enough movement, almost like a “shake.” The Shoe’s method is more accurate. The last public health impact that Smart Shoes offer is their ability to measure weight distribution. Knowing the weight on each shoe can help individuals be aware of their imbalance, make the appropriate adjustment, and practice walking more balanced. Furthermore, for people recovering from a stroke, fall, or other injury/disease, Smart Shoes can be a great tool in helping them relearn how to walk.

C. Technical Literature

Many Smart Shoes have been developed to measure certain data. In some cases, Smart Shoes have been developed to recognized daily activities [2]. These Shoes have an accelerometer, gyroscope, and pressure sensors to receive data from the user and determine what activity they are performing. Other Smart Shoes, with similar hardware, are used to estimate energy expenditure and measure a person’s heart rate [3]. This data can “provide appropriate feedback in function of users’ exercise information and health status.” Smart Shoes can also be used for energy harvesting from the user walking or running [4]. The energy that is generated can be used to charge portable devices and batteries. In addition to Smart Shoes, Smart Insoles have been used for similar purposes such as monitoring frail older adults walking speed [1]. This data is used to help identify frail older adults and what can be done to help them. Finally, when it comes to the medical field, Smart Shoes have even been used to assess mobility and monitor diseases [5]. According to an article by *applied sciences*, “Smart shoes offer the possibility to support prevention, diagnostic work-up, therapeutic decisions, and individual disease monitoring with a continuous assessment of gait and mobility.”

There are many different types of Smart Shoes that are made for different reasons. In every case, the shoes are a wearable technology that allow us to begin the integration of electronics and clothing.

D. Similar Products

Nike Hyper Adapt 1.0- Price Range: $200-$500

These shoes don't connect to the internet, but they have sensors in them that when someone put on the shoes, it tightens the laces for them. The first version of the HyperAdapt 1.0 was released in the United States on November 28, 2016. It was originally priced at $720.00. In comparison to the capstone Smart Shoes, the only advantage these shoes have is their smart lacing ability. The capstone Smart Shoes will be priced around $120-$180 which is much cheaper than the Nike Hyper Adapt 1.0. They will also have abilities to connect to the internet, a phone via Bluetooth, record weight, steps, foot strike, cadence, distance, and possibly pronation. The audience of the Nike Hyper Adapt 1.0 are more for people looking for convince where the audience of the capstone Smart Shoes are more for people who want to track their weight and/or improve in their walking form. More information available in [6].

Under Armour's HOVR Sonic sneakers- Price Range: $110

Under Armour's HOVR Sonic sneakers has Bluetooth capabilities in one of the shoes. The sneakers also have the ability to connect to an app called MapMyRun which is used by many people to track their runs. The shoes along with a smart phone can track time running, distance, steps, stride length, and cadence. In comparison to the capstone Smart Shoes, these products are very similar. Under Armour's HOVR Sonic sneakers are more designed for runners and people who wish to record their time and distance while running. Therefore, it has more accurate capabilities than the capstone Smart Shoes like measuring distance, time running, and stride length. The Smart Shoes may be more expensive ($120-$200) but they can measure weight, foot strike, and connect to the internet which makes it worth it. Under Armour's HOVR Sonic sneakers are more of a sports version of the capstone Smart Shoes and are for runners rather than people trying to learn how to walk again. More information available in [7].

The FeetMe insoles- Price Range: Demo only

The FeetMe insoles aren’t for sale yet, but they are very similar to the capstone Smart Shoes. Instead of actual shoes, this product is just two insoles that will fit in any shoe. They have capabilities like wireless charging, can store 26 days of movement collection, Bluetooth, 180-minute charge time, and 18 pressure sensors. They can measure cadence, stance duration, swing duration, step duration, support duration, and stride velocity. In comparison to the capstone Smart Shoes, The FeetMe insoles are more compact and can be used in more than one pair of shoes. They, however, cannot measure weight, distance traveled, or have displays. The FeetMe insoles are considered a medical product and has a target audience of people who are also learning how to walk again. Overall, the capstone Smart Shoes will try to be a cheaper price than the FeetMe insoles. More information available in [8].

Nurvv Run Smart insoles- Price Range: $179.99

The Nurvv Run Smart insoles are similar to The FeetMe insoles but are actually on the market and target the runner community. They weigh 22 grams, have 16 pressure sensors on each shoe, Bluetooth connectivity, connect to Nurvv Run app, a GPS chip, and a 5-hour battery life. The Nurvv Run Smart insoles can measure distance, run time, average pace, cadence, balance, step length, pronation, and foot strike. In comparison to the capstone Smart Shoes, they have similar features except that they are insoles and have a well-developed app to work with the product. The Nurvv Run Smart insoles do not measure weight and are mainly used for running rather than every day walking measurements. Since they only have a 5-hour battery life, this product wouldn’t work well for people who want to track their progress throughout the day. Overall, the Nurvv Run Smart insoles would be a good competitor against the capstone Smart Shoes, but since they target different audiences, both products will succeed in the market. More information available in [9].

2. Experimental Method

A. Engineering Requirements Development

(i) Customer Requirements

Customer Requirements:

1. Smart Shoes must be wirelessly charged (R)
2. Must track steps taken (R)
3. Must track distance traveled (R)
4. Be able to Measure and track a person's weight (R)
5. Have a long battery life (R)
6. Have Bluetooth connectivity (R)
7. Can show stride and how to improve it (R)
8. Can send data to a phone for viewing (R)
9. Can measure weight distribution (R)
10. The Smart Shoes are comfortable (R)
11. Must have accurate measurements (R)
12. The shoes are inexpensive (D)
13. Should be lightweight (D)
14. Should be water resistant (D)
15. Should be durable (D)

Standards:

16. Title 47 Section 15.247 of Code of Federal Regulations "Operation within the bands 902–928 MHz, 2400– 2483.5 MHz, and 5725–5850 MHz"
17. IEEE Std 802.11
18. ISO/IEC/IEE 42010:2011

Constraints:

19. System must be designed, built, and tested no later than April 23rd, 2020
20. System must not exceed budget of $300.00 USD
21. Must contain internal power source
22. Must provide a 5-volt source of power for internal electronics
23. System must fit inside a size 10 men's shoe.

Figure 1: Customer Requirements

The customer requirements were obtained by the customer survey results. If a specification had an average value of 4 or higher, on a scale from 1-5, then it became a customer requirement. Any requirement in Figure 1 with an (R) is required and any requirement with a (D) is desired in the design. Desired requirements were also obtained from the customer survey results as additional comments or greater than 3 average value. The engineering requirements in Table 1 are based on the customer requirements as well as the constraints to the project and federal standards for Bluetooth and Wi-Fi.

(ii) Standards

Title 47 Section 15.247 of Code of Federal Regulations “Operation within the bands 902–928 MHz, 2400– 2483.5 MHz, and 5725–5850 MHz” is a standard that deals with using Wi-Fi and Bluetooth. It states that at least 15 channels should be used and that the hopping channel carrier frequencies are separated by 25khz for bands of 2400– 2483.5 MHz. This standard was met by using the Arduino Nano 33 IoT microcontroller which meets the standard.

IEEE Std 802.11 is a standard that explains how to use wireless local area networks and MAC addresses. The standard was met by using the Arduino Nano 33 IoT microcontroller which only has one medium access control address.

ISO/IEC/IEEE 42010:2011 is a standard that addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions. Many IoT devices use this standard. The standard requires that architecture descriptions use viewpoints, which are addressed by architecture views. The NINA-W10 used on the Arduino Nano 33 IoT microcontroller uses viewpoints in its architecture descriptions. Thus, the standard is satisfied.

(iii) Environmental and Safety

The Smart Shoes are designed to be safe for the environment and the user. Further revisions of the Smart Shoes will be done to make them waterproof and safe for the user to walk in the rain. When it comes to the environment these Smart Shoes will be worn by a user and eventually thrown away. The goal is for many components of the Smart Shoes to be recycled as the shoes may be worn but the electronics are still functioning. Lead-free solder was used while making the Smart Shoes and other initiates will be taken in order to lower the greenhouse gas emissions produced by the Smart Shoes.

(iv) Legal, Political or Ethical Concerns

If the capstone Smart Shoes become a sellable product there are some liabilities that need to be accounted for. While using the Smart Shoe someone could get electrocuted from shoes, they could break after the consumer wears them for a period of time, or they could catch on fire from charging. These outcomes will try to be negated through testing, but the possibility will always be there. If these outcomes do happen, then the company will have to pay for it. To further ensure that these things do not happen it would be wise to get the product approved by Underwriter Laboratories and the Federal Communications Commission. For further legal action, a lawyer or expert in the law should look over the product and check to make sure that people cannot sue the company for minor defects in the product.

The capstone Smart Shoes will help people recover better from injuries or diseases that impact their ability to walk properly. The weight sensors in the shoes will be able to detect imbalances and foot strike while the user is wearing them. This information will be sent to the user’s smart phone and allow them to change how they walk based on the information. The capstone Smart Shoes will also be able to track people’s weight for them as they go about their day. No more will someone have to check their weight in the morning and write it down on a piece of paper. The Smart Shoes will record the user’s weight periodically through the day (when they stand up). This information will also be sent to the user’s smart phone, and they can make health adjustments based on the collected data. Allowing humans to monitor their weight more could help decrease obesity or other weight related issues. The capstone Smart Shoes are more of an everyday shoe rather than a sport shoe.

(v) Sustainability

To make the capstone Smart Shoes, many materials will be used. In the shoe itself there may be plastic, foam, leather, rubber, and other synthetic materials. Then in the circuit portion there will be metals like copper and aluminum, plastic, microchips, and possibly even glass for the displays. Eventually, the Smart Shoes will wear out and their users will throw them away. This could potentially cause a lot of waste in landfills or to a Waste-to-Energy facility. Using lead solder could also be potentially dangerous to the environment. With this known information, choices can be made to protect the environment such as using lead-free solder. Other choices could be to use recyclable material in the Smart Shoes so that they can be returned after being worn out and then recycled.

B. Engineering Requirements

Marketing Requirements	Engineering Requirements	Justification
11,4,9	The weight sensors and A/D converter will measure a person's weight within +/- 2%	To obtain an accurate weight within 8 pounds or less
2,3,11	The system will be able to record steps and distance traveled within +/- 5%	An accurate measurement of steps taken, and distance traveled
5,15,21,22,	The system can perform for 24 consecutive hours	This will allow users to charge the shoes while they sleep and wear them all day without losing power
6,8,7,16,17	The system will be able to send all stored data through Bluetooth to a smart device.	This is how the measurements will be viewed
12,20	The Smart Shoes will have a production cost of less than $300	Many smart shoes are between $100-$300
10,23	The system will fit inside a pair of size 10 men's shoe	The electronics should not hinder walking or foot movement
1,14	The system will only receive power wirelessly	This makes charging easier and the system will be inherently more water resistant
13	The shoes will weigh less than 5 pounds	Most shoes are under this weight and not far from it
2,3,4,7,9	The system will be able to store at least 4GB of data	For storing measurement data
4,11,9	The weight sensors will be able to measure weight from 0 to 400 pounds.	Most humans are inside of this weight range
5	The microcontroller will turn off after 1 minute of the system not being in use	To save battery life

Table 1: Engineering requirements

C. Level 1 and 2 Functional Decompositions

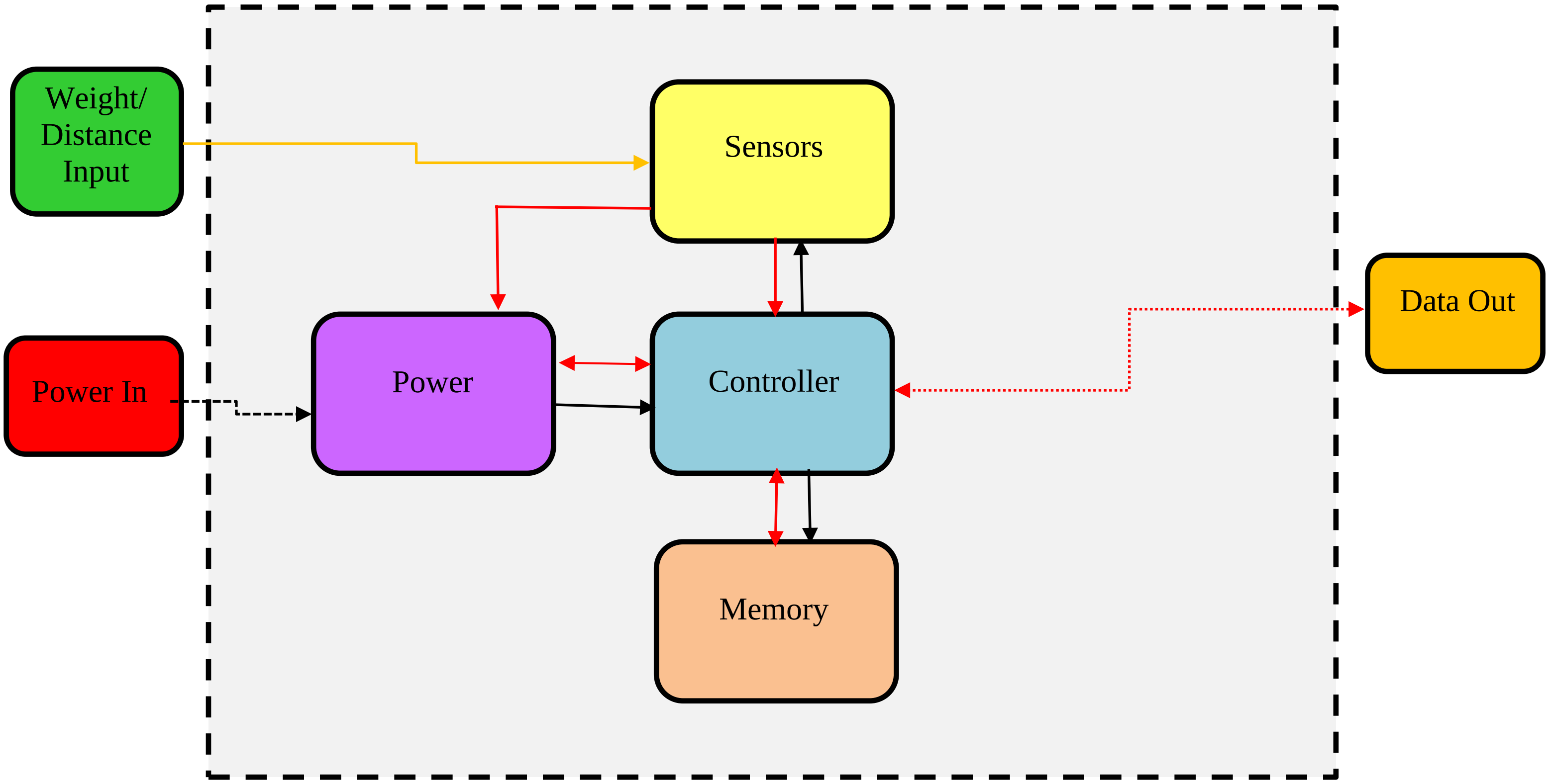


Figure 2: Level 1 Decomposition

The Smart Shoes level I decomposition is shown in Figure 2 above. Each shoe is comprised of four sub-systems. These sub-systems are the sensors being used, the power of the system, stored memory, and the controller that manages all the data. The sub-systems are connected in a way that will allow the Smart Shoes to work properly. In Figure 2, orange lines are for sensors inputs, black lines are for power, red lines are for data, and dashed lines are for wireless transmission. One shoe will have "Data Out" going to a smart device via Bluetooth and the other shoe will be a slave that sends data to the master controller of the first shoe. All the blocks inside the grey box will be developed as part of the Smart Shoes and everything outside the box is either an input or output to the system.

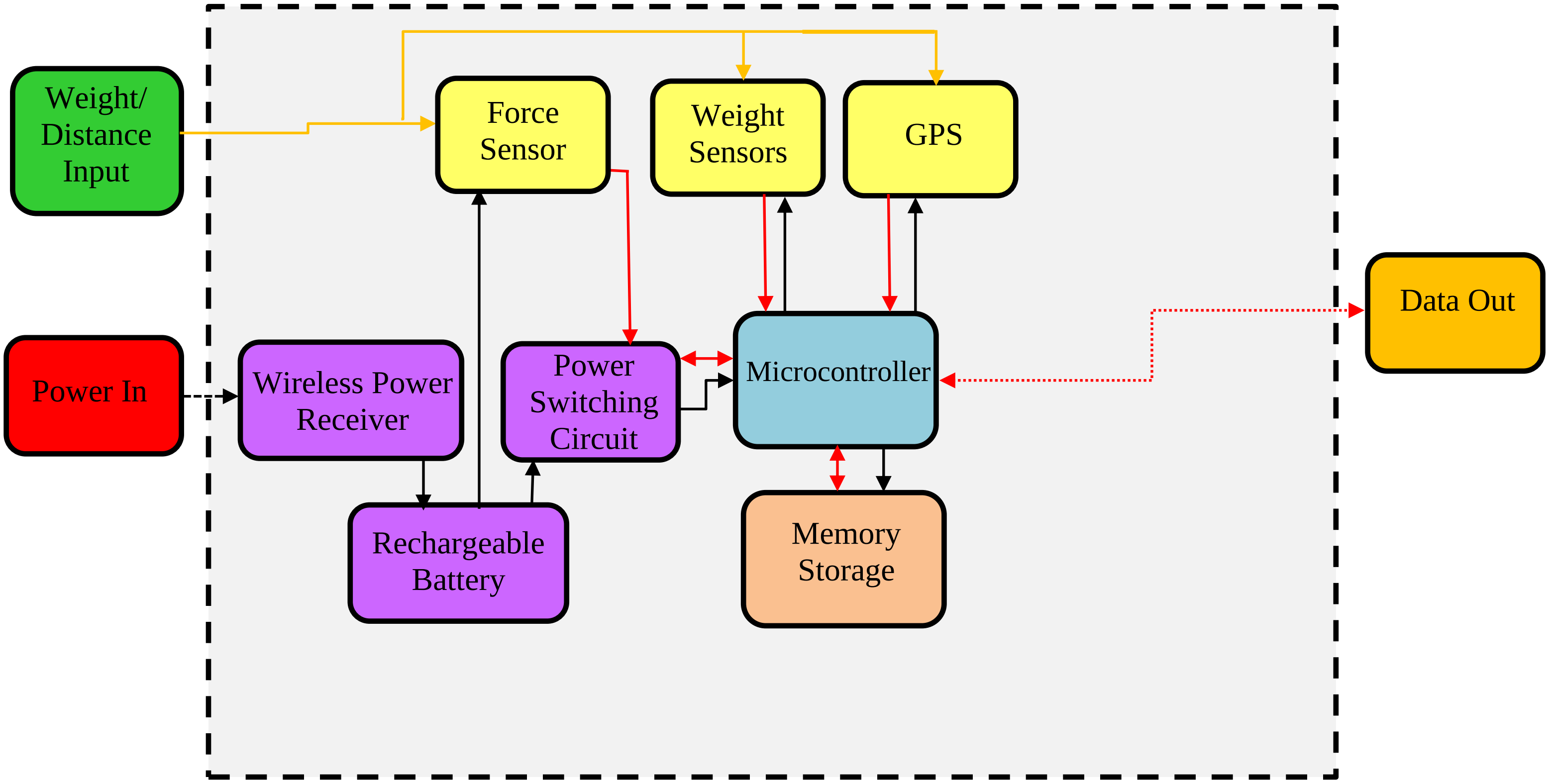


Figure 3: Level 2 Functional Decomposition

The Smart Shoes level II decomposition is shown in Figure 3 above. As shown in Figure 1, power from a wireless charger will be received by the "Wireless Power Receiver." This module will have an output of 5V DC that will charge the "Rechargeable Battery." From the battery, power will pass through the "Power Switching Circuit." The module will act as a switch that will let current pass to the "Microcontroller" when there is weight on the shoes and will switch off the current when no weight is measured for a minute. The "Microcontroller" will receive data from the "Weight Sensors", "GPS", "Memory Storage", and possibly "Data Out". The "Microcontroller" will process this data and send it to a smart device so that the data can be view. It will also send data to the "Memory Storage" to be stored. The "Weight Sensors" will measure the weight on the Smart Shoes and send a voltage value to the "Microcontroller" for processing. They will also send a voltage value to the "Power Switching Circuit" so that it can switch properly. The "GPS" module will receive coordinate information from satellites and send this information to the "Microcontroller" for processing. Lastly, the "Memory Storage" module will receive data from the "Microcontroller" and store it for memory. It will also send data to the "Microcontroller" when the microcontroller requests for it.

(i) Decomposition Level I and II I/O Tables

Module	Weight Sensors
Inputs	• Weight from user • 5V DC from microcontroller
Outputs	• Analog data in mV to the microcontroller • Analog data in mV to the power switching circuit
Functionality	• Produces a small analog voltage that is related to the weight on the sensors

Table 2: Weight Sensors I/O

Module	GPS
Inputs	• Signal from GPS satellites • 5V DC from microcontroller
Outputs	• Digital data for microcontroller
Functionality	• Receives location from GPS satellites and sends this information to the microcontroller

Table 3: GPS I/O

Module	Force Sensor
Inputs	• Weight from user • 5V DC from Battery • Data from Power Switching Circuit
Outputs	• Data to Power Switching Circuit
Functionality	• Turns the microcontroller on when the Force sensor is pressed

Table 4: Force Sensor I/O

Module	Wireless Power Receiver
Inputs	• Power from wireless charger
Outputs	• 5V DC to rechargeable battery
Functionality	• Receives power from a wireless charger and is used to recharge the battery

Table 5: Wireless Power Receiver I/O

Module	Rechargeable Battery
Inputs	• 5V DC from wireless receiver
Outputs	• 5V DC to power switching circuit • 5V DC to weight sensors
Functionality	• Used to store energy and power the whole system

Table 6: Rechargeable Battery I/O

Module	Power Switching Circuit
Inputs	• 5V DC from rechargeable battery • Analog data in mV from weight sensors
Outputs	• 5V DC to microcontroller
Functionality	• Turns the microcontroller off when the system is not in use and on when in use

Table 7: Power Switching Circuit I/O

Module	Memory Storage
Inputs	• 5V DC from microcontroller • Data from microcontroller
Outputs	• Data to the microcontroller
Functionality	• Stores measurement data from the microcontroller and sends it back when needed

Table 8: Memory Storage I/O

Module	Microcontroller
Inputs	• **5V DC from power switching circuit** • **Data from weight sensors** • **Data from GPS** • **Data from memory storage** • **Data from smart device (data out)**
Outputs	• **Data to memory storage** • **5V DC to GPS** • **5V DC to memory storage** • **Data to smart device (data out)**
Functionality	• **Receives data from sensors and memory storage, processes the data, and sends it to a smart device via Wi-Fi; Sends data to memory storage from sensors**

Table 9: Microcontroller I/O

D. Subsystem Design

(i) Explanation of the Simulation Circuit and Software Used

Cadence OrCad was used to simulate the Power Switching Circuit for the Smart Shoes. In this software, the Pspice tool was used to create the schematic layout of the circuit. Libraries of electrical components in Pspice made it possible to digitally simulate a component with all of its electrical properties. In the schematic, the components are connected using wires and when the circuit is complete it can be simulated in Pspice. After it is simulated, voltages and currents can be read at different nodes and evaluated.

A diagram of the simulated circuit can be seen in Figure 4. The diagram has different sections (marked by dashed lines) that make up the Power Switching Circuit. Going through the sections, the 'Reference Voltage (3V)' section is a voltage divider that uses 300K and 200K ohm resistors to create 3V from a 5V source. This reference voltage is used in the 'Comparator (>3V)' section that uses an op-amp to check if the pressure sensor is less than 3 volts. If the pressure sensor is greater than 3 volts, the 'Pull-Up Resistor' will keep the output of the op-amp at 5V and if it is less than 3 volts then the output will change to 0 volts. The ‘Pressure Sensor’ in the diagram is a pulse generator that generates a pulse that goes from 5V down to 2.9V every 20 seconds. This represents the same voltage as a physical pressure sensor that operates as a variable voltage divider by the resistance changing due to pressure applied to the sensor. The ‘Buffer for Arduino analog input’ section uses an op-amp as a buffer to generate the same voltage as the pressure sensor without creating a voltage drop. The output of this op-amp will be used as the input to an analog pin on the Arduino microcontroller. The ‘SR Flip Flop’ section is the main logic of the circuit. Using three NAND gates, the flip flop is SET and outputs 5V when 0V is received from the ‘Comparator (>3V)’ section. It will continue to output 5V until it is RESET by the Arduino output. The ‘Arduino Output’ in the diagram is a pulse generator that generates a pulse that goes from 0V up to 5V every 40 seconds. This represents the same voltage that an Arduino microcontroller would output on a digital pin. When the Arduino output pulses 5V the SR Flip Flop is RESET and outputs 0V. The ‘GND Connection From Battery to Arduino’ section in the diagram uses an N-channel MOSFET to create a path from 5V to a resistor and finally ground. This represents using an N-channel MOSFET to create a path from the ground of a battery to the ground of an Arduino microcontroller. Whenever the output of the SR Flip Flop is 5V, there will be a connection from the battery ground to the microcontroller ground. Whenever the output of the SR Flip Flop is 0V, the ground connection will break, and the Arduino microcontroller will not be powered. The final section is the ‘5V Battery’ section. This represents a 5V battery. Anywhere on the diagram that reads “Battery” in red is connected to the battery in the ‘5V Battery’ section.

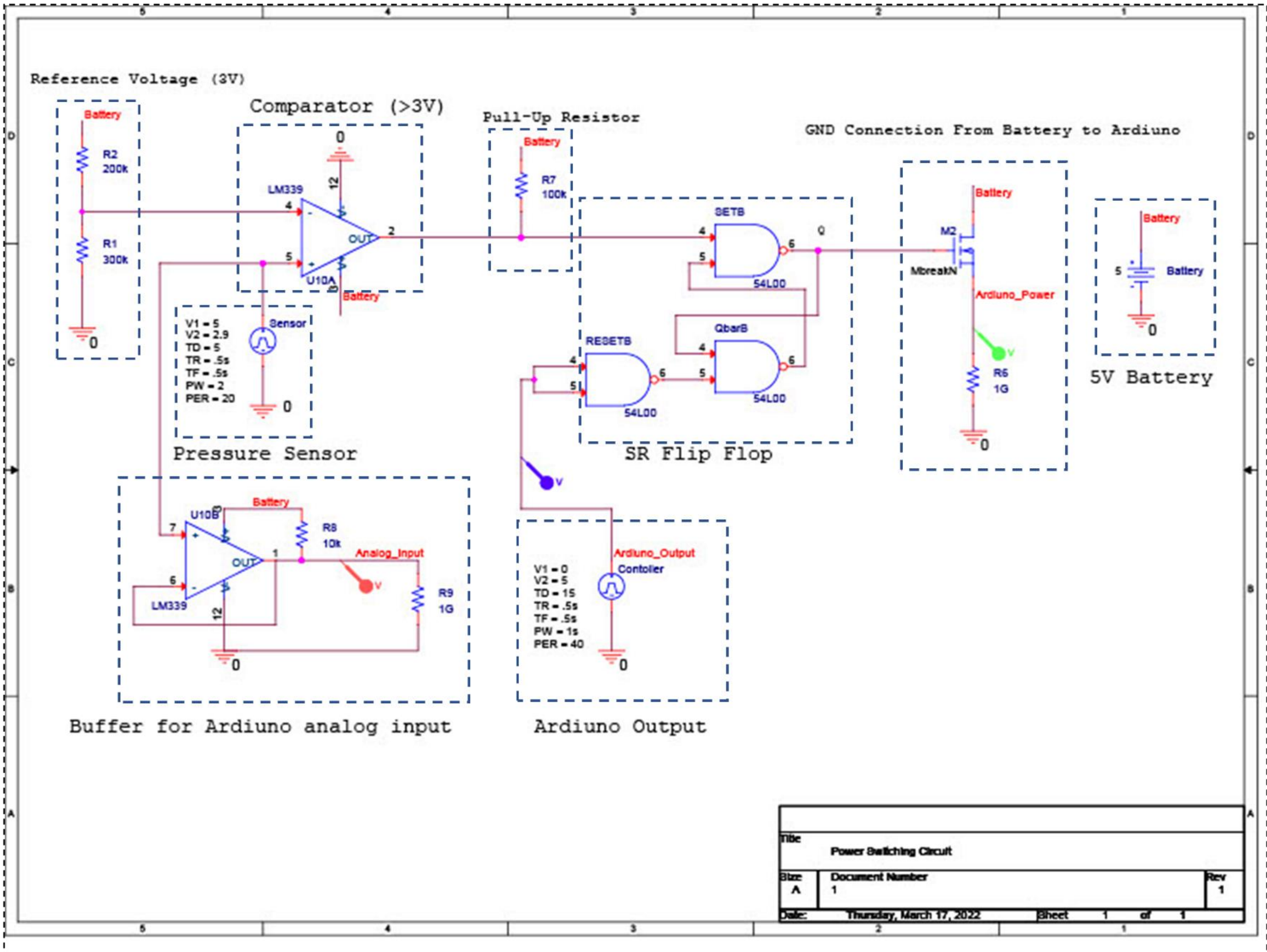


Figure 4: The Pspice diagram for the Power Switching Circuit in the Smart Shoes. The circuit is broken down into sections (marked by dashed lines) to better understand its functionality.

(ii) Theory and Equations

- Reference Voltage (3V)-

  Equation 1: Voltage Divider{ $Vout = (V^{+} - V^{-})(R_2/(R_2 + R_1))$}

  Equation 2: Ohm's Law { $V = I * R$ }

  - Actual Values: $Vout = (5V - 0V)(300K/(200K + 300K)) = 3V$
  - Explanation of Operation: The voltage is dropped between 2 resistances. According to the equation, the 200K ohm resistance will have a 2 volt drop and the 300K ohm resistance will have a 3 volt drop.

- Comparator (>3V)-
  - Theory:

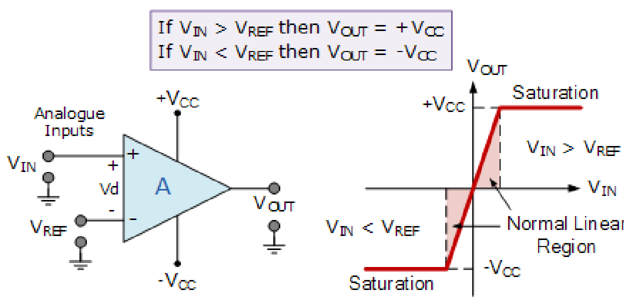


Figure 5: Op-amp Theory

  - Actual Values: $+Vcc = 5V, -Vcc = 0V, VREF = 3V, VIN = 2.9V - 5V$
  - Explanation of Operation: When the pressure sensor is pressed to lower than V_{REF} (3V) then $Vout = -Vcc = 0V$. When the pressure sensor is not pressed ($VIN = 5V$) then $Vout = +Vcc = 5V$.

- Buffer for Arduino analog input-
  - Theory: Inverting input voltage = non-inverting input voltage. Currents going into Inverting input and non-Inverting input are ideally 0A.
  - Actual Values: non-Inverting input voltage = pressure sensor voltage. Inverting input voltage = Vout.

- Explanation of Operation: Since the Inverting input voltage is equal to the non-inverting input voltage, the non-Inverting input is connected to the pressure sensor voltage, and the Inverting input voltage is connected to Vout, Vout is equal to the pressure sensor voltage. Also, since the input currents are ideally 0A, there will be no voltage drop to the pressure sensor voltage.

- Pull-Up Resistor-
  - Theory: The LM339 Op-Amp has an open-collector at the output. It will need a pull-up resistor so that the collector does not pull the output to ground.

Equation 3: LM339 Op-Amp Pull Up Resistance $\{R_{pull-up} \geq ((V_{cc} - .2\text{V})/I_{Max})\}$

  - Actual Values: $RPull - up\ >=\ (5V\ -\ .2V\ /\ 4ma)\ >=\ 12K\ ohms.$
    $RPull - up\ =\ 100K\ ohms$

  - Explanation of Operation: When the comparator is active (Vin < 3V) the transistor at the output of the LM339 will turn on and will make the node go to 0V creating a 5V voltage drop over the pull-up resistor (.05ma). when the comparator is inactive (Vin >3V) the transistor of the LM339 will be off and there will be no current traveling through the pull-up resistor and the node will go to 5V.

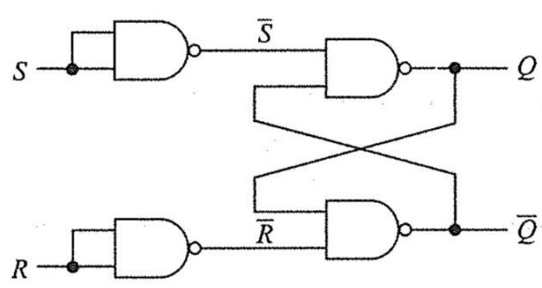
- SR Flip Flop-
  - Theory:
  - 

Figure 6: SR flip flop

- Truth Table:

R	S	Q
0V (Low) 0	0V (Low) 0	Q
0V (Low) 0	5V (High) 1	5V (High)
5V (High) 1	0V (Low) 0	0V (Low)
5V (High) 1	5V (High) 1	N/A

Table 10: SR flip flop truth table

- Explanation of Operation: Once the SR flip flop is Set (01) it will continue to output 5V even if 00 or 01 happen. Once it is Reset (10) the output will be 0V until the SR flip flop is set again. Since the output of the comparator is inverted logic, the S NAND gate is not needed.

- <u>GND Connection From Battery to Arduino-</u>
  - Theory: When voltage is applied to the gate of the MOSFET, electrons are pulled up from the substrate and create a path for electrons to travel from the source to the drain and vice-versa.

    Equation 4: Resistance of a Wire $\{R = \frac{\rho l}{A} \quad \rho = 1.7x10^{-8}\ \Omega m\ \}$

  - Actual Values: $Vturn - on \ = \ 2V$

  - Explanation of Operation: When the SR flip flop outputs 5V the MOSFET will create the connection between the Arduino ground and the ground of the battery. This will turn the microcontroller on since there is a ground connection. Once the SR flip flop outputs 0V, the MOSFET will not have a connection between the grounds and the microcontroller will be off. Since $V_{turn-on}$ has a value in between 0 and 5 volts, this operation will work as expected.

Equation 5: Distance Calculation (Haversine Formula)

$$a = sin^2(\Delta\varphi/2) + cos\,\varphi_1 \cdot cos\,\varphi_2 \cdot sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R \cdot c$$

φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km)

Equation 5 was used to calculated distance using latitude and Longitude from the GPS [10].

(iii) Simulation Results

The engineering requirements for the Smart Shoes state that the microcontroller will turn off 1 minute after not being used. The Power Switching Circuit fulfills this requirement. Figure 7 shows three voltmeters on the Power Switching Circuit. One will show the Pressure Sensor voltage(red), another the digital pin output voltage of the microcontroller(Blue), and one will show if there is a connection between the Arduino ground and the battery ground(Green).

Figure 8 shows all three of these voltages over some time. The results show that by pressing the pressure sensor below 3V(Red), the microcontroller will turn on(green) and pressing it again will do nothing to the state of the microcontroller. The results also show that when the digital pin on the microcontroller (Blue) goes High (5V) the microcontroller will turn off(green). This can also be seen even more clearly in Figure 9, which separates the voltages between the pressure sensor with the ground connection and the digital pin with the ground connection. The simulation is not a perfect representation since the digital pin does not wait 1 minute to go High after the last time the pressure sensor has been pressed. This is due to limitations in the Pspice software and not being able to have the Arduino microcontroller represented in the software. However, the overall function of the Power Switching Circuit is validated by this simulation and the waiting 1 minute function will be completed by software in the microcontroller. The functions validated by this simulation are that pressing the pressure sensor turns the microcontroller on; pressing the pressor sensor multiple times will not affect the state of the microcontroller; and that when the digital signal goes High (5V) the microcontroller will turn off and stay off until the pressure sensor is pressed again.

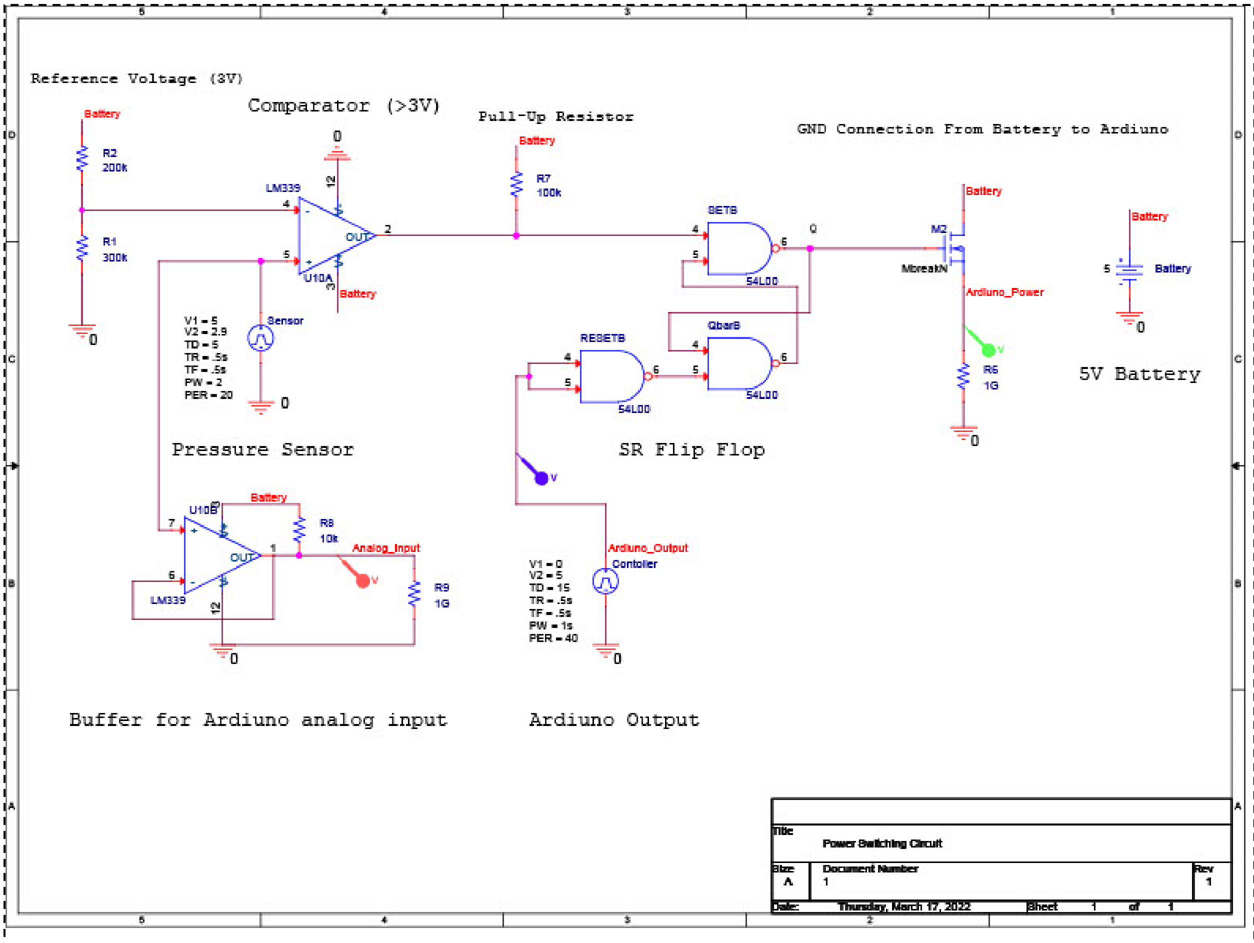


Figure 7: The Pspice diagram for the Power Switching Circuit in the Smart Shoes without sections. Colors of voltmeters match up with the graphs in Figures 3 and 4.

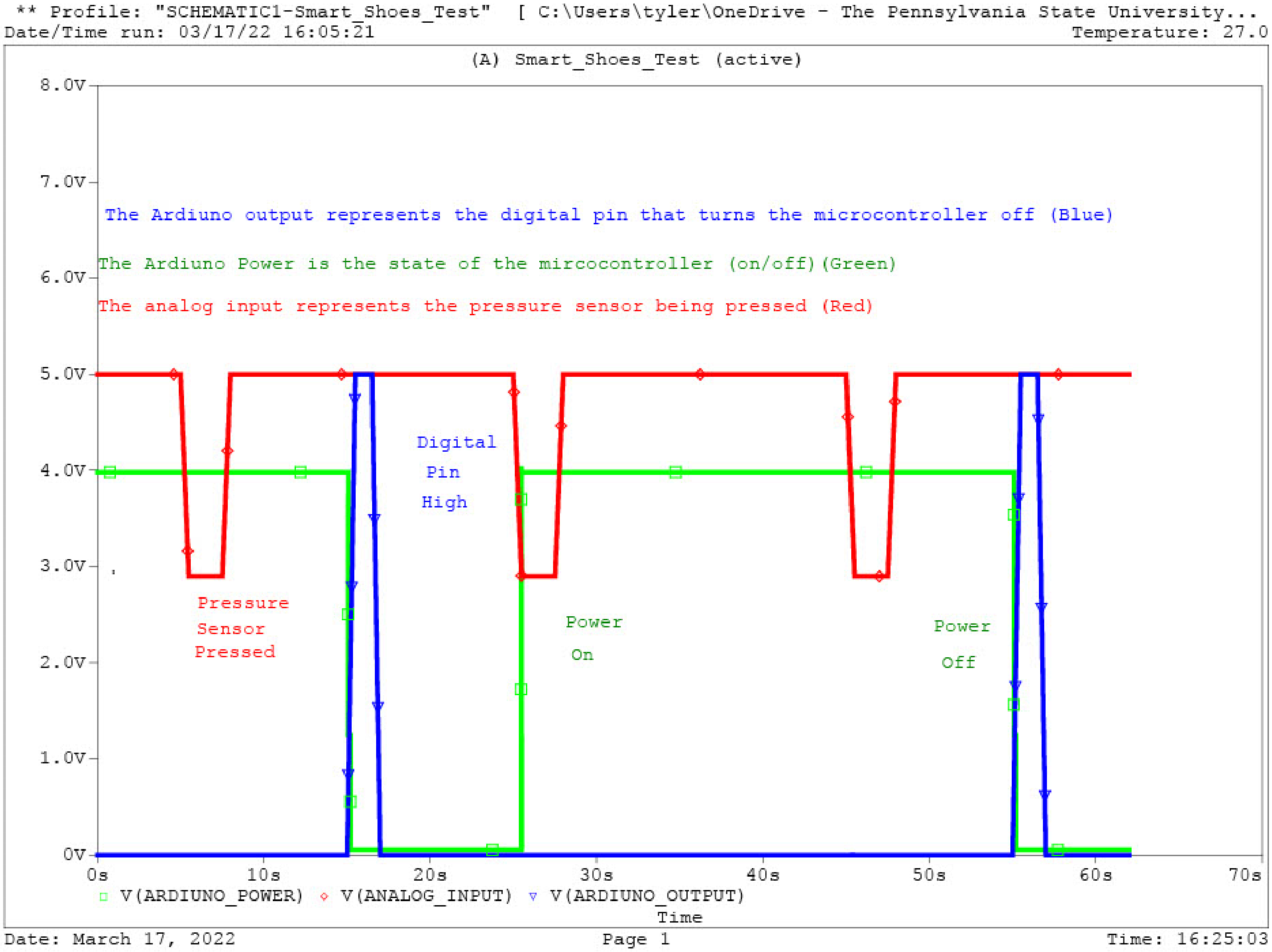


Figure 8: The simulation results for the Power Switching Circuit

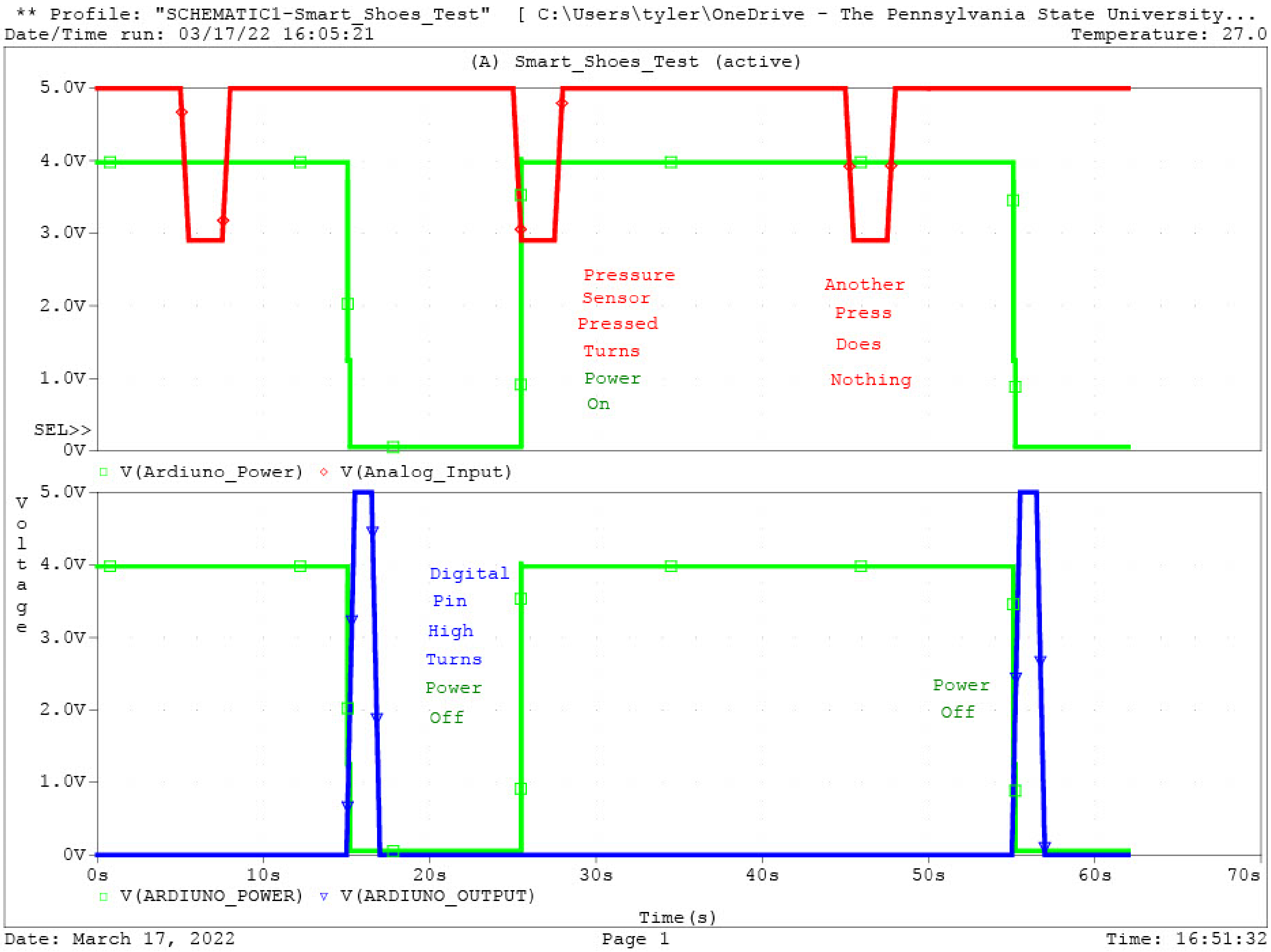


Figure 9: The simulation results for the Power Switching Circuit in Figure 3 broken into two graphs.

E. PCB Design

(i) General Overview

Circuit Maker was used to create the PCB design for the power switching circuit. The software works by creating a schematic where all your components are connected by wires. This can be seen in Figure 10. Each component has a footprint that will be used for the physical PCB. In addition to the circuit schematic, Circuit Maker has a layout where you design how the physical PCB will be connected. This can be seen in Figure 11. The physical path of the wires and footprints of components are laid out and connected by the user.

The current revision of the power switching circuit can be seen in Figures 12-14. The circuit has 2 IC’s, 6 resistors, 1 MOSFET, and 8 vias to connect to other circuitry. Through hole components were used to bread-board the circuit. However, these components were very large compared to surface mount and drew more current. Thus, surface mount IC’s and resistors were chosen in the final PCB design. A surface mount MOSFET was chosen in the first revision of the PCB but had a gate-source threshold voltage that was too high and caused the circuit not to work. A new MOSFET was chosen with a gate-source threshold voltage that worked with the circuit. Because of this change, a second revision PCB was made.

(ii) Designing the PCB

The power switching circuit schematic can be seen in Figure 10. In the schematic there are 8 vias that will be used to connect the power switching circuit to other circuitry. The vias are VCC, PSGND, PSVCC, A0, D10, NanoGND, Vin, and GND. VCC and GND will connect to a 5-volt battery. PSGND and PSVCC will connect to the ends of a pressure sensor. A0 will connect to an analog pin on the Arduino nano 33 IoT to read the pressure sensor voltage. D10 will connect to a digital pin on the Arduino nano 33 IoT to turn the microcontroller off. Finally, NanoGND and Vin will connect to the power and ground of the microcontroller.

The first IC (LM339) has two op-amps that are being used (one for a comparator and one for a buffer). The Second IC (SN74LS00NS) did not have a footprint so one was created using the footprint of a similar IC package. It uses 3 NAND gates in which one output goes to the gate of the MOSFET. The physical layout of the wires can be seen in Figure 11. In Circuit Maker, the footprints of each component were placed to create a compact PCB. Then every node was connected with wires on the top layer or the bottom layer. 2-layers were needed to keep the PCB compact and so wires didn't cross over each other and create a short. To get a node from the top layer to the bottom a via was needed and multiple can be seen in Figures 11 and 14. To keep things smaller, these vias did not have holes in the middle of them. After all the nodes were connected, an outline of the PCB was established (the blue outline in Figure 13) and a Gerber file was created. This Gerber file was used to send to a manufacturer so they could manufacture the designed PCB. The 1st revision of the power switching circuit PCB can be seen in Figure 15.

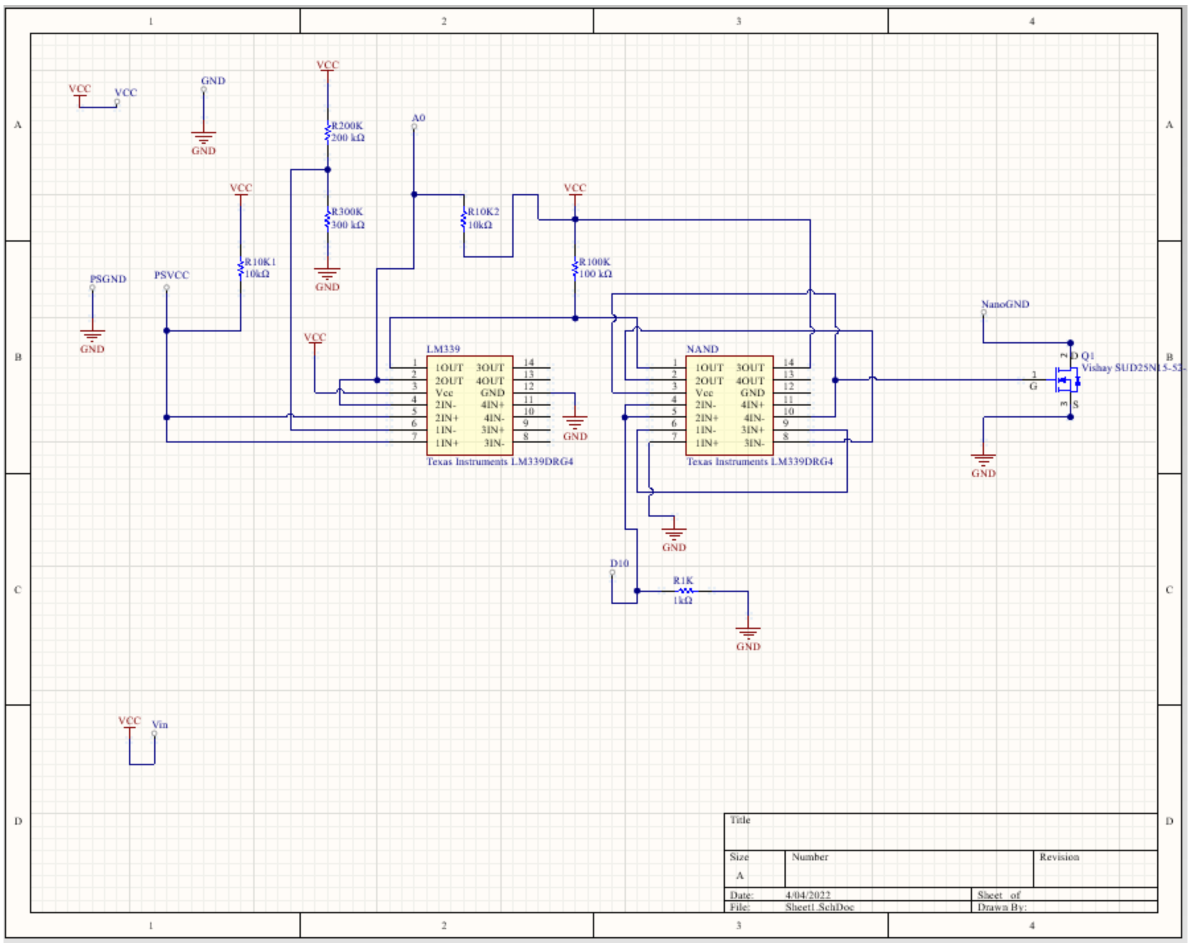


Figure 10: The power switching circuit (1st revision) schematic in Circuit Maker

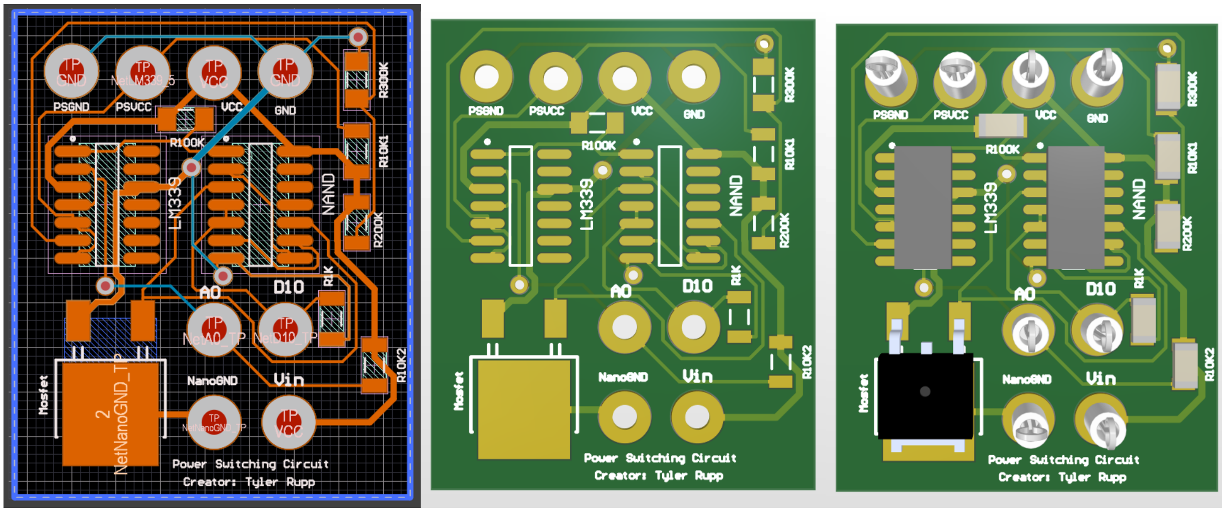


Figure 11: The power switching circuit (1st revision) physical 2D layout in Circuit Maker (Left) and the PCB 3D image (populated and nonpopulated).

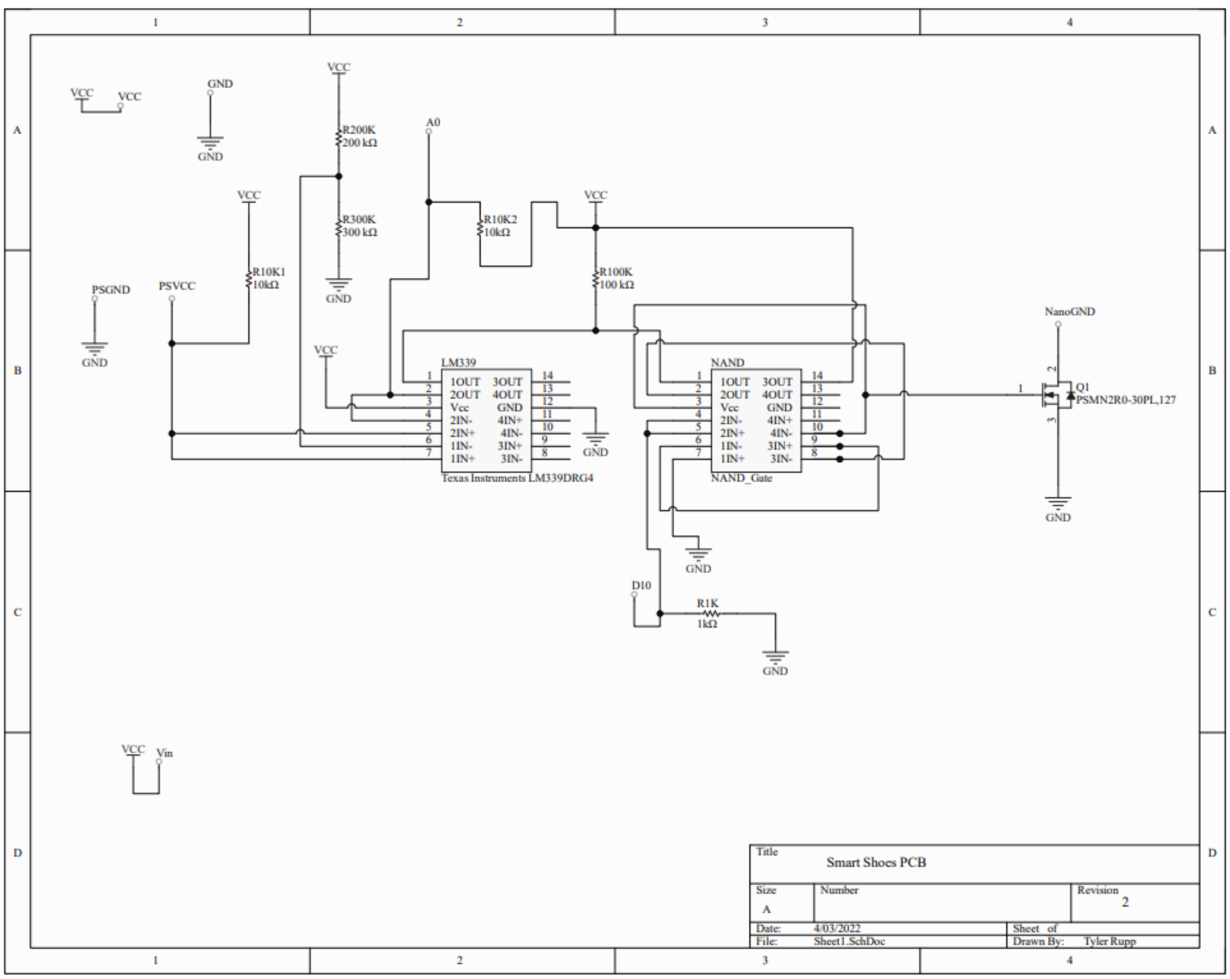


Figure 12: The power switching circuit (2nd revision) schematic in Circuit Maker

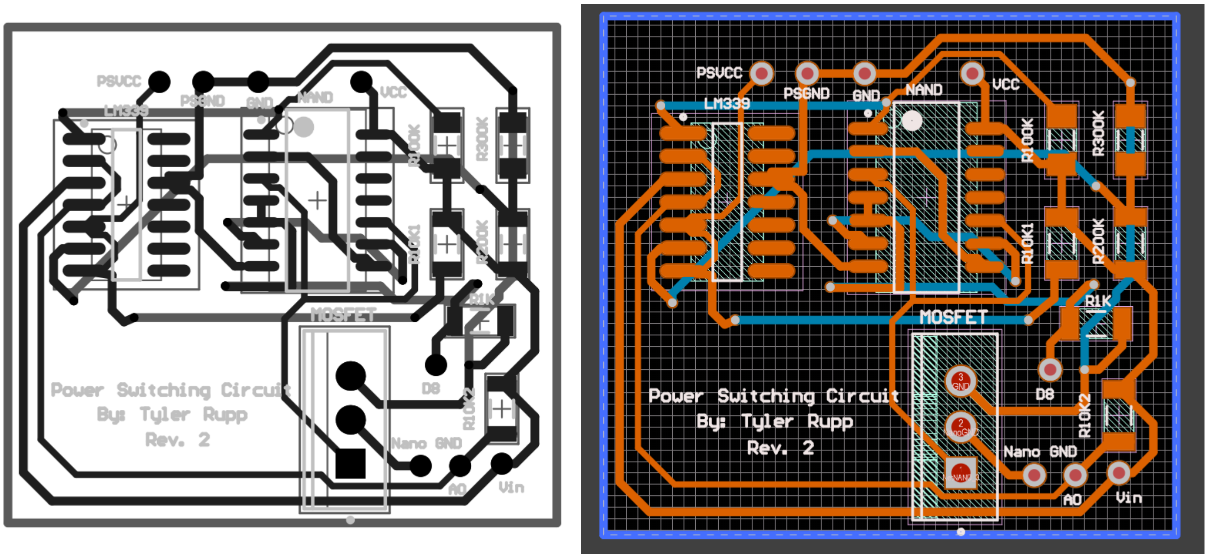


Figure 13: The power switching circuit (2nd revision) physical 2D layout in Circuit Maker.

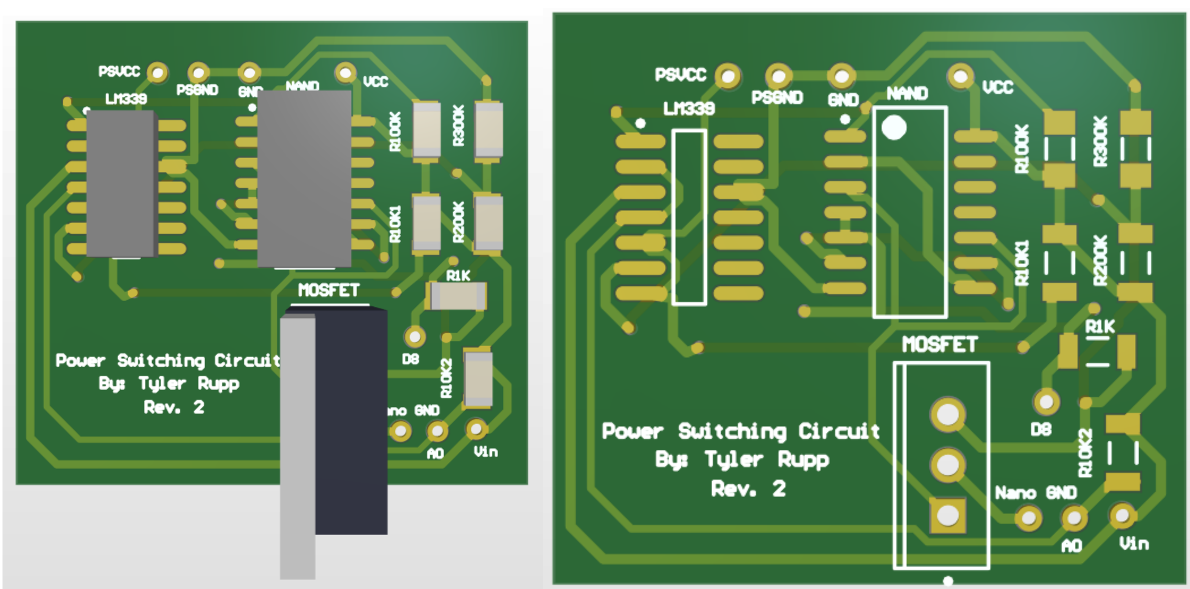


Figure 14: The power switching circuit (2nd revision) physical 3D image in Circuit Maker (populated and nonpopulated).

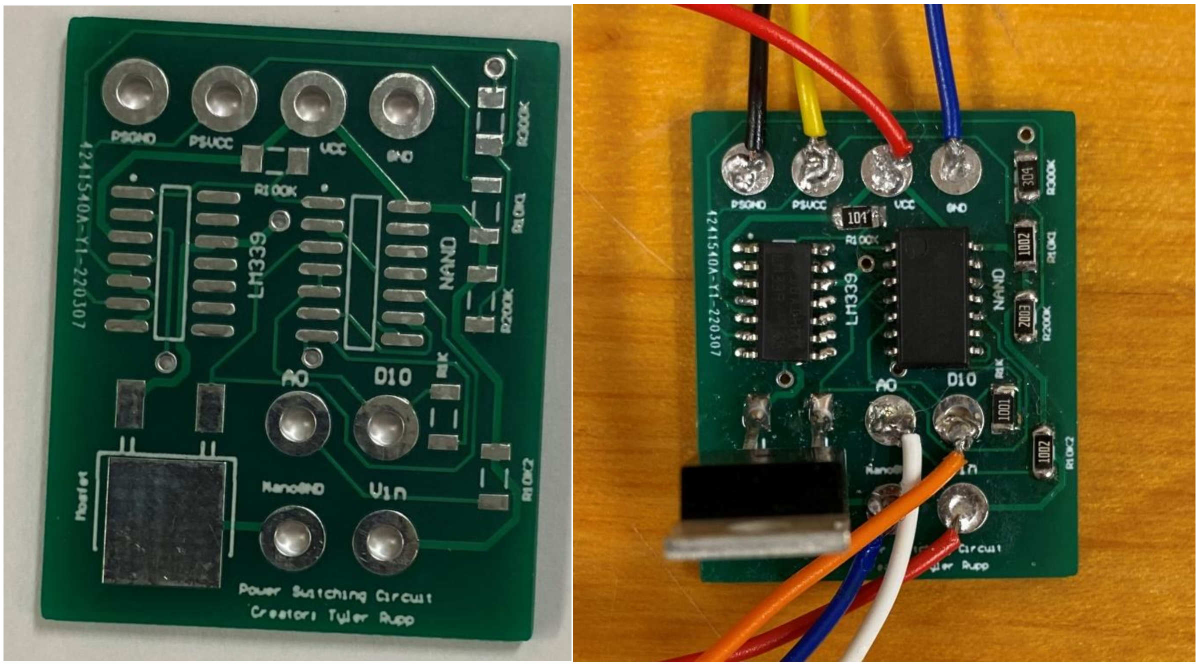


Figure 15: The ordered power switching circuit (1st revision) PCB (populated and nonpopulated).

(iii) Testing the PCB

The power switching circuit PCB was tested using a Tektronix MSO 2012 Mixed signal oscilloscope and a Fluke 87 Multimeter. The setup of the test can be seen in Figures 16 and 17. The goal of the test was to get oscilloscope measurements of the digital pin, analog pin, and the 3.3V reference pin of the microcontroller connected to the power switching circuit PCB. The results can be seen in Figures 24-30. The other goal was to measure the current draw of the PCB and microcontroller when the microcontroller was off and on. These results can be seen in Figures 18-23. The results from the oscilloscope will be compared to the simulation results in the next section.

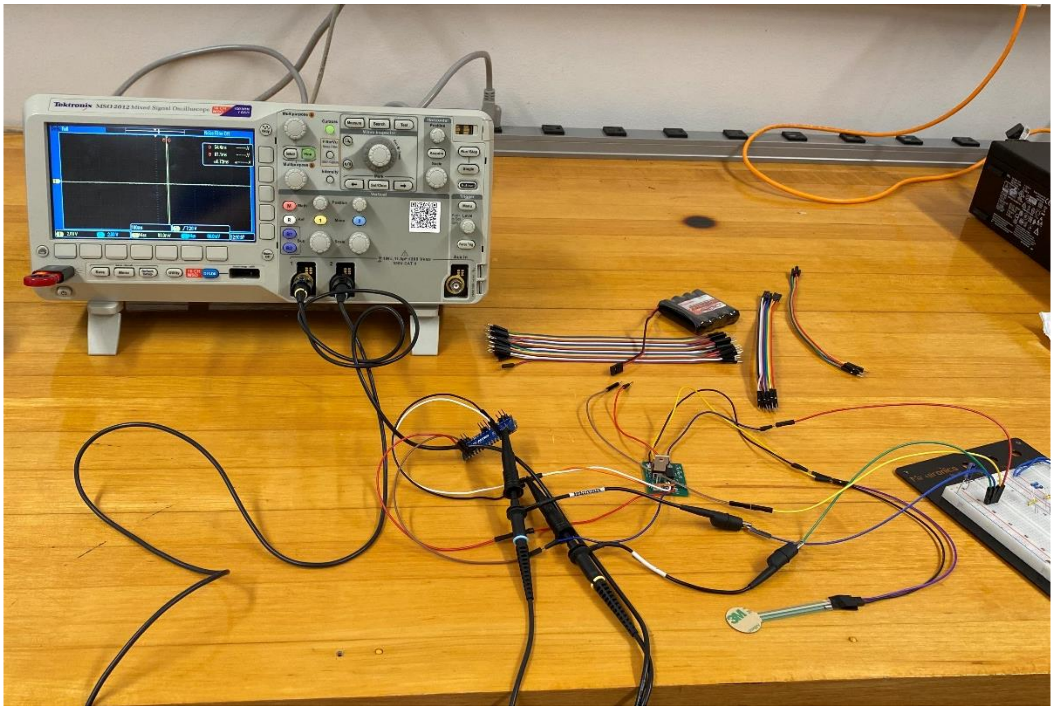


Figure 16: The setup to test the power switching circuit PCB with the Arduino nano 33 IoT.

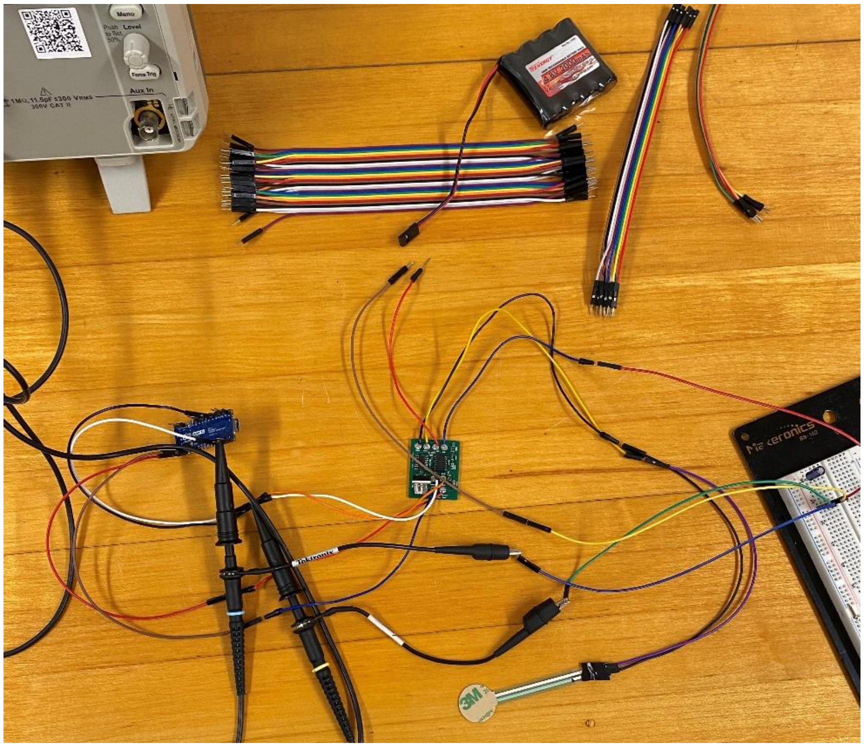


Figure 17: The setup to test the power switching circuit PCB with the Arduino nano 33 IoT.

Figure 18: The current draw for when the power switching circuit has the microcontroller on (26.43 mA) and when it has it off (2.87mA)

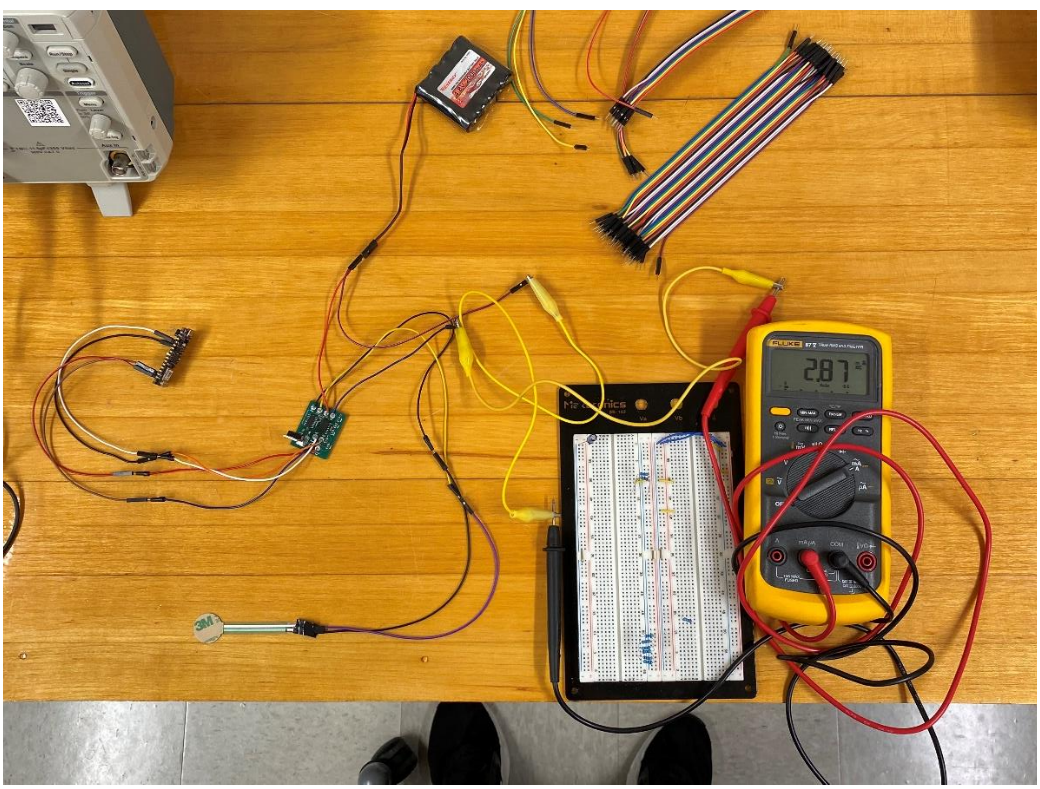


Figure 19: The ampere reading taken at the ground side of the battery for when the microcontroller is off.

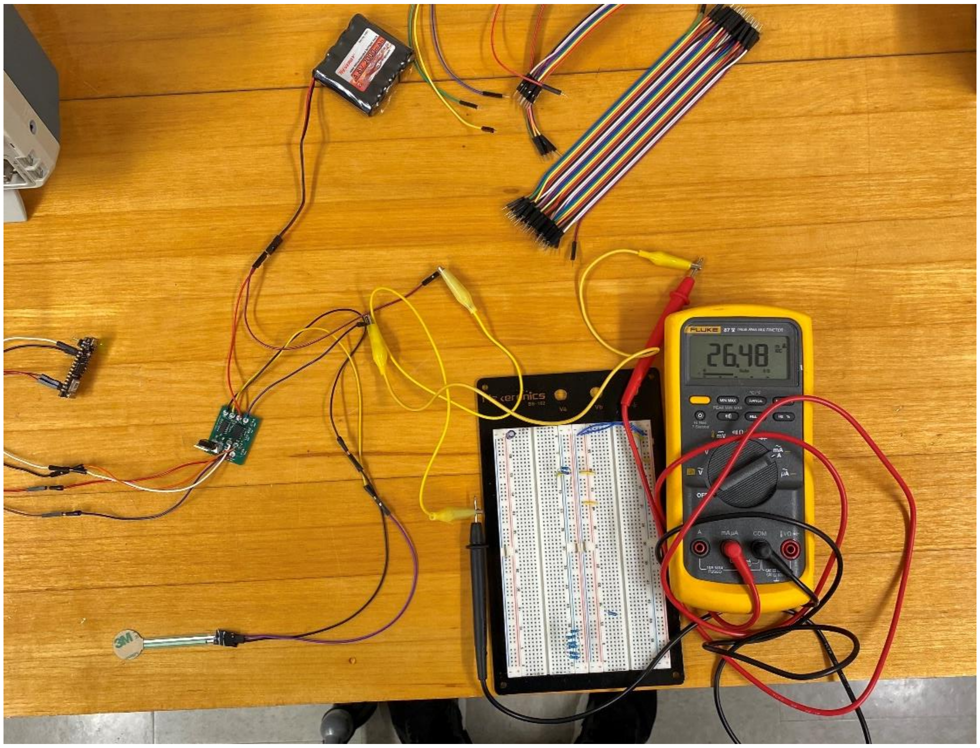


Figure 20: The ampere reading taken at the ground side of the battery for when the microcontroller is on.

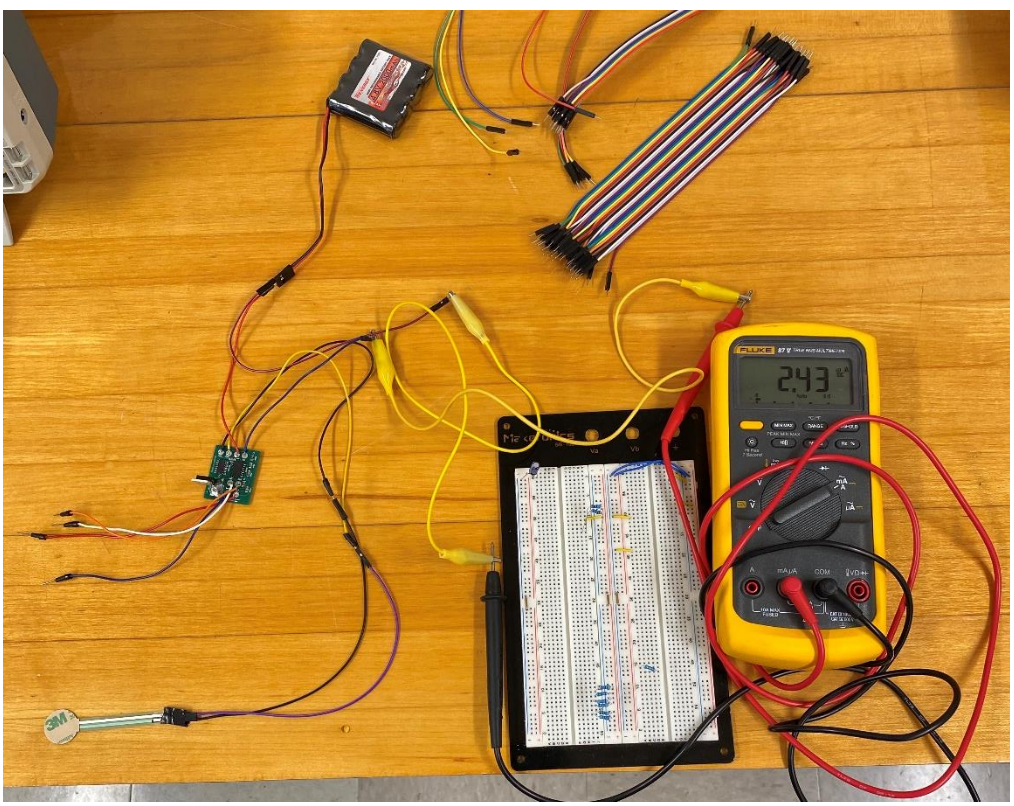


Figure 21: The ampere reading taken at the ground side of the battery for when the microcontroller is completely disconnected from the power switching circuit.

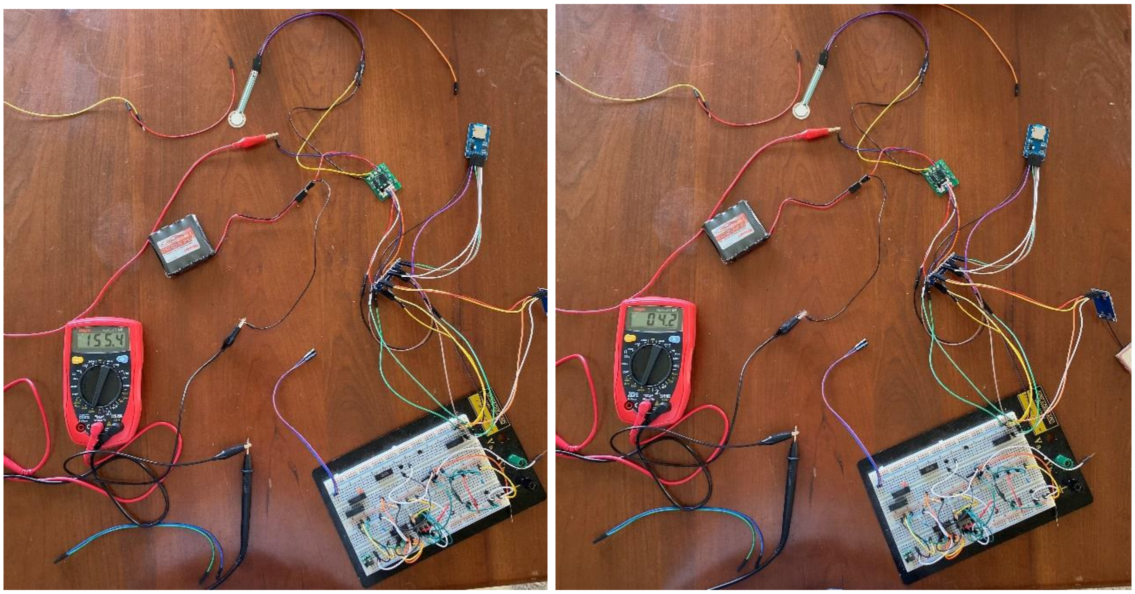


Figure 22: The ampere reading taken at the ground side of the battery for when the microcontroller is on(left) and off (right). This is with most of the subsystems connected to the micro controller.

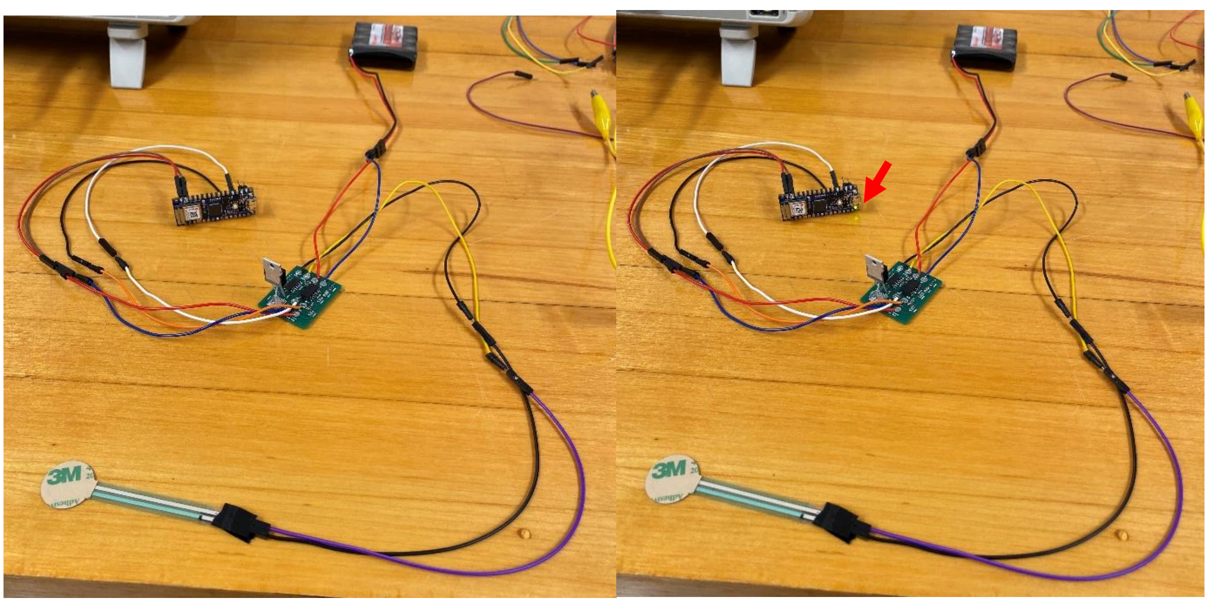


Figure 23: The off state of the microcontroller (left) and the on state of the microcontroller (right). An indicating LED will turn on when the microcontroller is on.

The smart shoes engineering requirements state that “the microcontroller will turn off after 1 minute of the system not being in use” and that “the system can perform for 24 consecutive hours.” The power switching circuit will be used to fulfill both requirements. For the first requirement, the power switching circuit will have to work with the microcontroller. Figures 26 and 27 show the oscilloscope graph of the microcontroller turned on from pressing the pressure sensor. If the pressure sensor is pressed again the only thing that will happen is that the 1-minute timer to turn off will reset. The microcontroller is reading the pressure sensor from an analog pin and can tell when it is pressed. This data is used in the microcontroller to reset the 1-minute timer. Once the timer is up, the microcontroller will send a logic ‘high’ through a digital pin and this will reset the SR flip flop in the power switching circuit. Since the SR flip flop is connected to the MOSFET, the MOSFET will lose its gate voltage and the microcontroller will stay off until the pressure sensor sets the SR flip flop again. The microcontroller turning off can be seen in Figures 24 and 25.

The second engineering requirement’s goal is saving power consumption. In the PCB test, the current draw is 26.48 mA when the microcontroller is on and 2.87 mA when off (Figures 20 and 21). This means that by turning the microcontroller off, 23.61 mA are being saved. If the system was on all the time, it would last 75.5 hours (about 3 days) and if the system were off all the time, it would last 697 hours (about 29 days) with a 2000mAh battery. Both the system on and off satisfy the engineering requirement. This would make the power switching circuit seem unnecessary, but the current draw changes once the whole system is connected. Figure 22 shows the current draw of most of the system on (155.4 mA) and off (4.2 mA). If the whole system was on all the time, it would last 12.87 hours and if the whole system were off all the time, it would last 476 hours (about 20 days) with a 2000mAh battery. This shows that the power switching circuit is necessary to turn off the microcontroller off in order to save power consumption.

(iv) PCB Testing Vs. Simulation Results

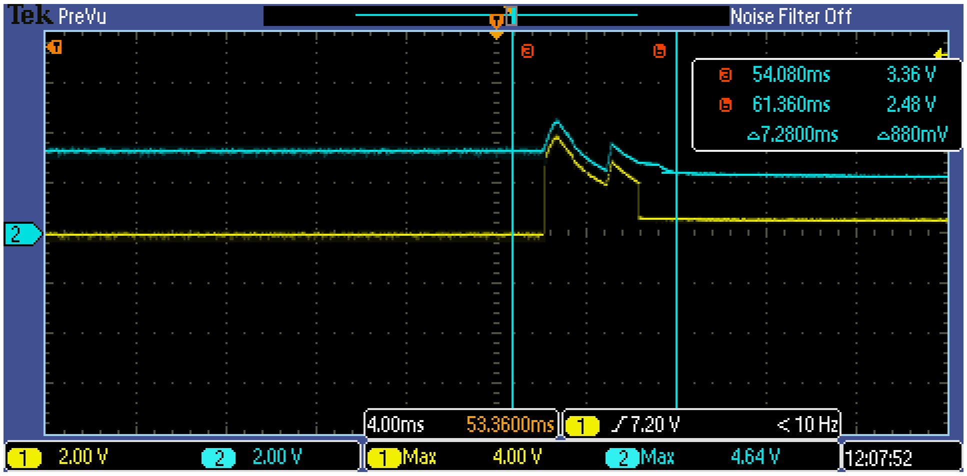


Figure 24: The oscilloscope graph of the digital pin (yellow) turning off the microcontroller (blue). The blue line is the 3.3V output voltage of the microcontroller because Vin is connected directly to the 5V of the battery.

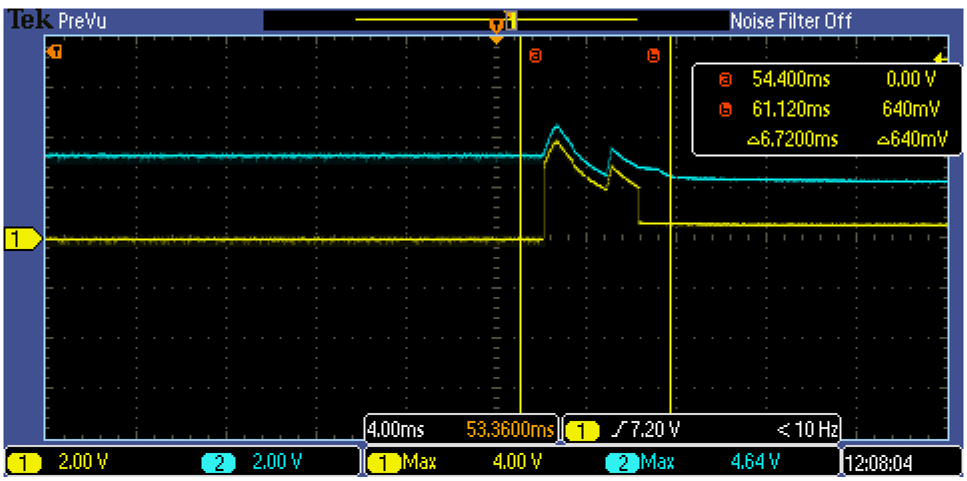


Figure 25: The oscilloscope graph of the digital pin (yellow) turning off the microcontroller (blue). The yellow line represents the digital pin that goes high to turn the microcontroller off. Once the microcontroller turns off the digital pin goes low again.

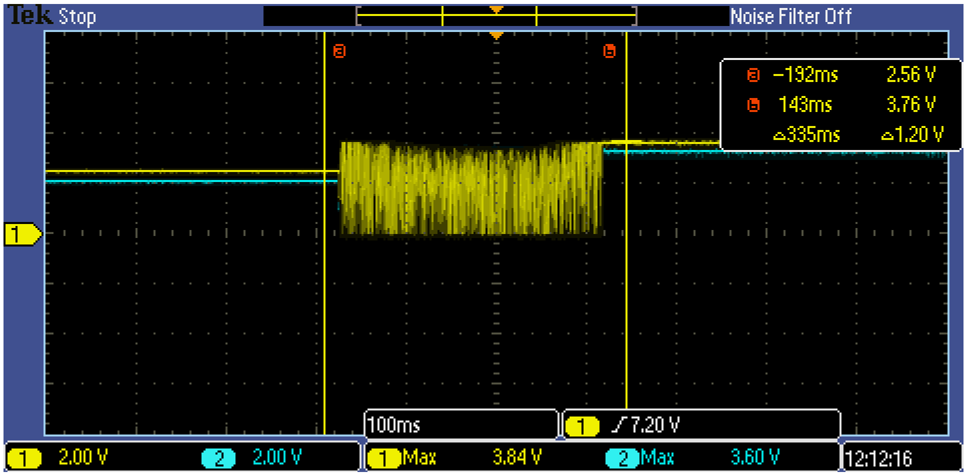


Figure 26: The oscilloscope graph of the analog pin (yellow) turning the microcontroller on (blue). The yellow line is taken at the output of a buffer (op-amp) which is why the signal bounces between 0V and 4V (the op-amps max output voltage).

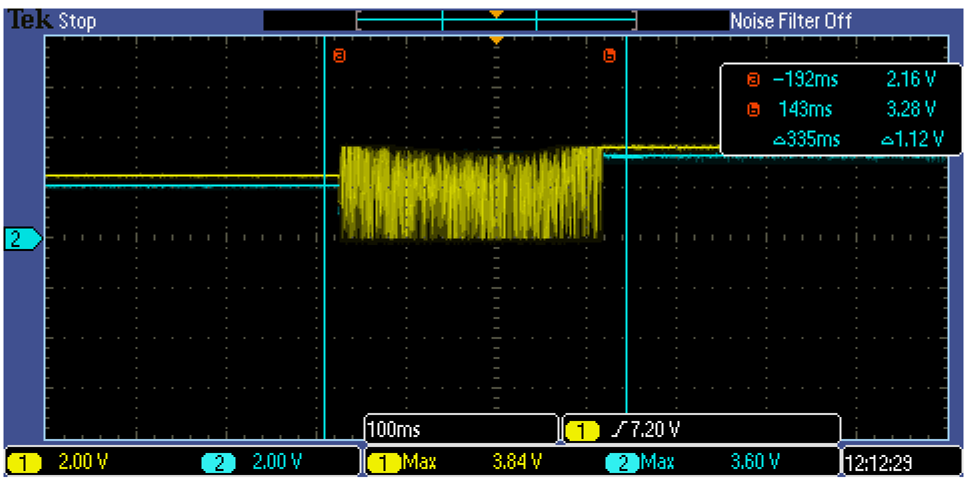


Figure 27: The oscilloscope graph of the analog pin (yellow) turning the microcontroller on (blue). The blue line is the 3.3V output voltage of the microcontroller because Vin is connected directly to the 5V of the battery.

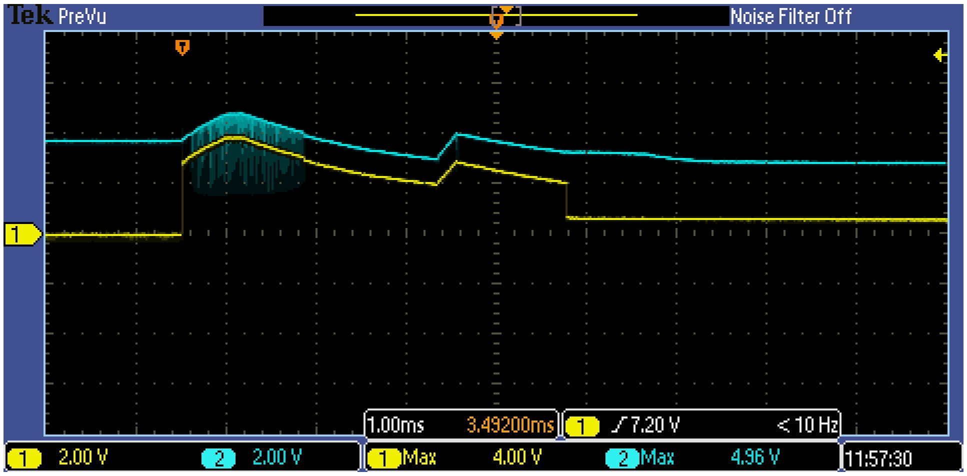


Figure 28: The zoomed in oscilloscope graph of the digital pin (yellow) and the analog pin (blue) when the microcontroller is being turned off.

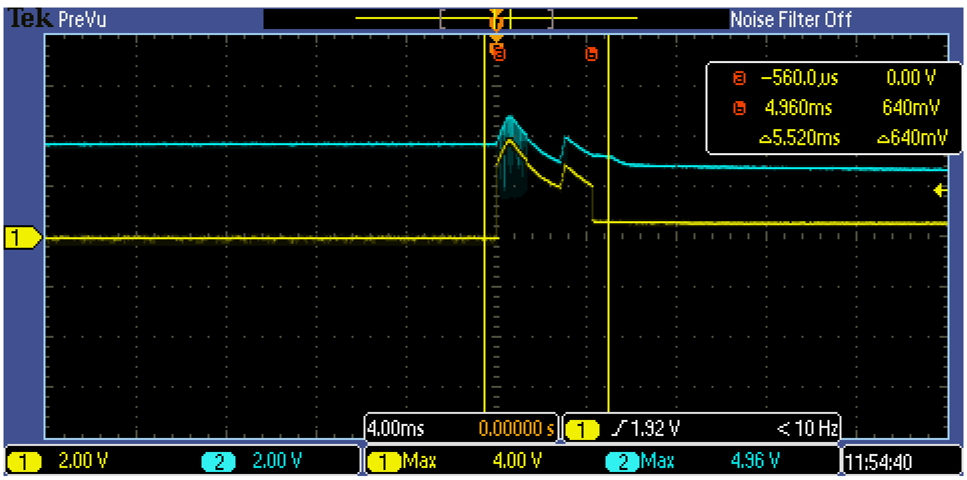


Figure 29: The oscilloscope graph of the digital pin (yellow) and the analog pin (blue) when the microcontroller is being turned off. This graph shoes the voltages and timing of the digital pin

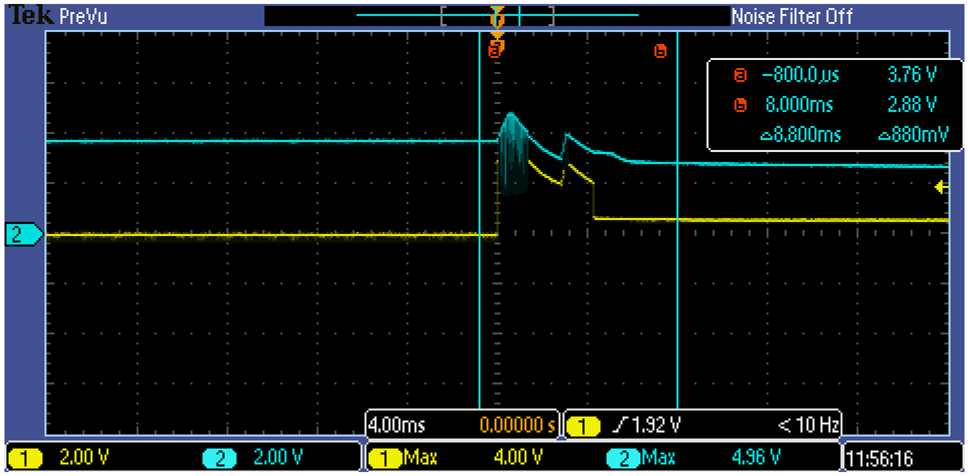


Figure 30: The oscilloscope graph of the digital pin (yellow) and the analog pin (blue) when the microcontroller is being turned off. This graph shoes the voltages and timing of the analog pin.

The comparison between the Pspice simulation results and the PCB results is slightly complicated. In the Pspice simulation results (Figure 7), the MOSFET connected the VCC of the microcontroller and the VCC of the battery. In the PCB results, the MOSFET connects the GND of the microcontroller to the GND of the battery. With this in mind, the functionality of the circuit works in both. Figures 24 and 25 show the PCB turning the microcontroller off after 1-minute. The digital pin pulse is certainly not as smooth as the simulation results but works just the same. The 3.3V reference voltage drops from 3.36V to 2.48V. obviously, this does not match the simulation results because of the MOSFET connection difference mentioned earlier. However, this does show that the GND connection between the battery and microcontroller is broken. Figures 26 and 27 show the PCB turning the microcontroller on by pressing the pressure sensor. There is a lot of noise and oscillation in the analog signal due to the buffer being used. However, if this signal was smoothed out by a capacitor, the simulation results and PCB results would look very similar. Lastly, Figures 28-30 show the digital pin and analog pin signals while the PCB is turning the microcontroller off after 1-minute. While the analog pin is being turned off, there is also some noise and oscillation.

In conclusion, the simulation results and PCB test results are in agreement. There is a lot of oscillation in the analog pin reading the pressure sensor, but the microcontroller takes samples and converts it to digital so this should not affect the functionality of the circuit. The digital pin in the PCB test is not a smooth pulse but this will also not affect the functionality of the circuit. The current readings from the Fluke 87 multimeter also show that the current draw is significantly less when the microcontroller is off vs being on. This means that the power switching circuit PCB has fulfilled its purpose of saving power for the whole system.

3. System Integration and Results

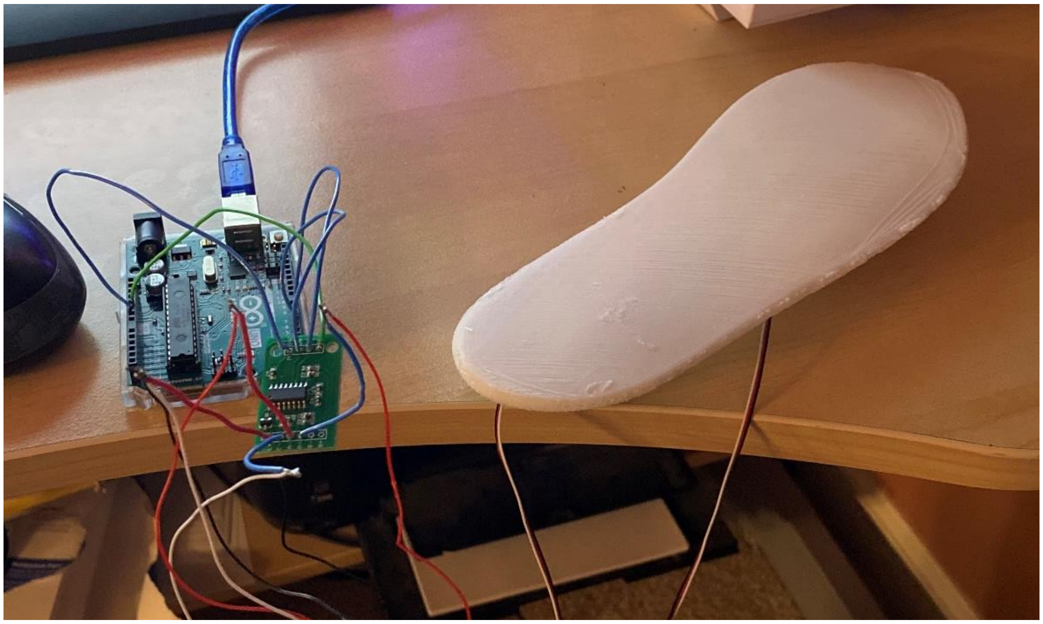


Figure 31: Testing the weight sensors and A/D converter

Figure 32: An example weight from the weight sensors

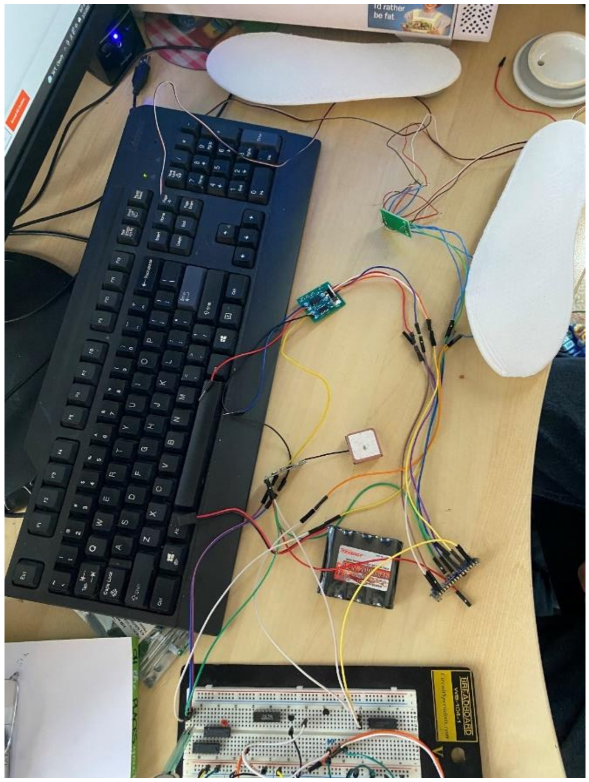


Figure 33: Integrating the GPS module and the Power Switching Circuit

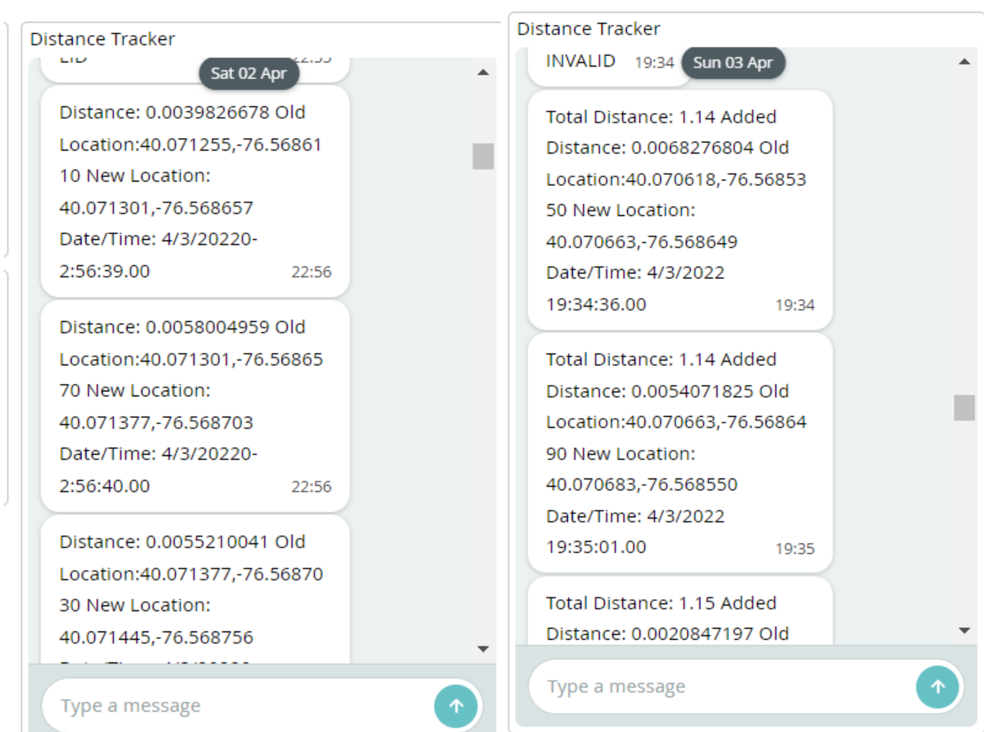


Figure 34: Example data from the GPS module

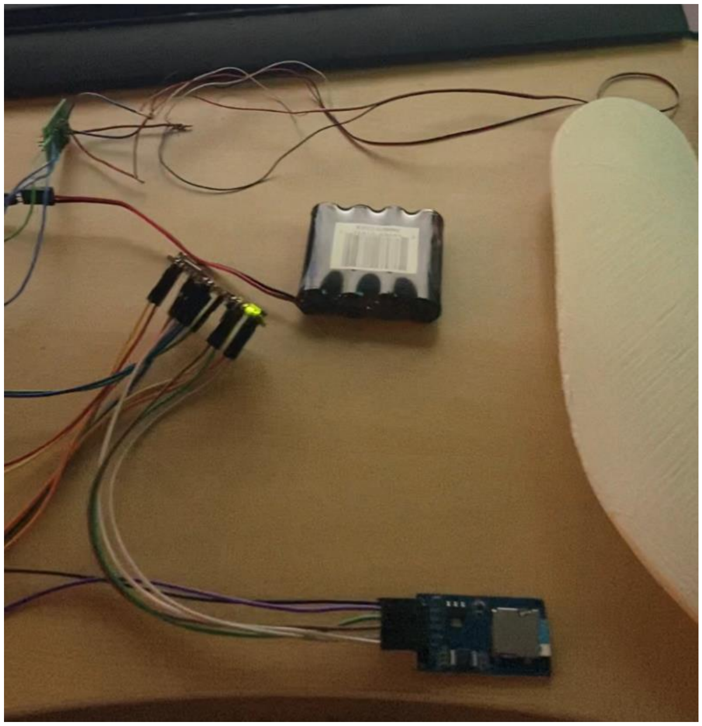


Figure 35: Testing and Integrating the micro-SD card and module

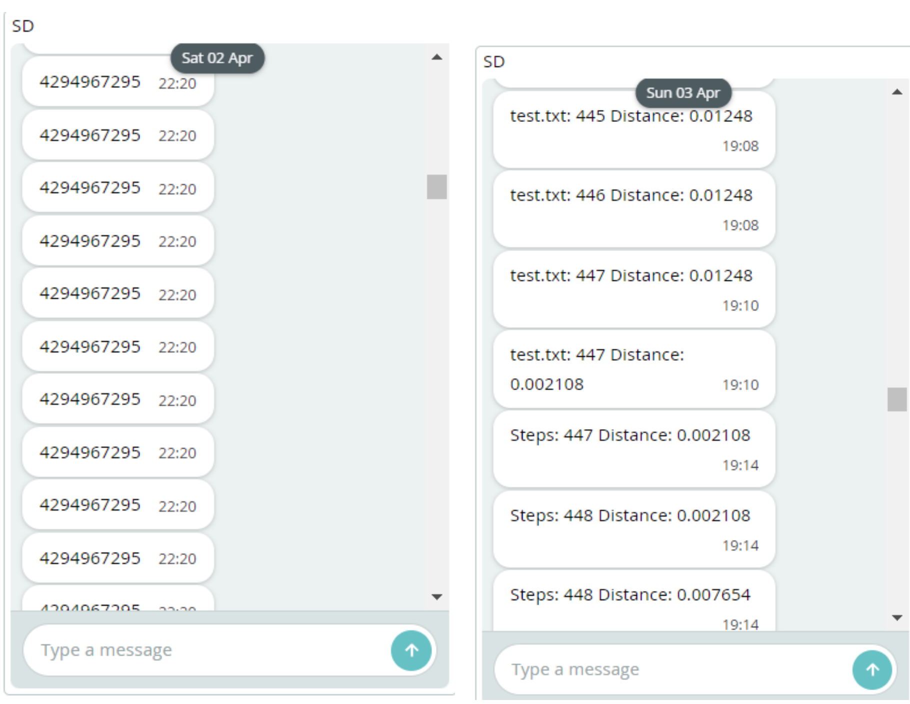


Figure 36: Example data from the micro-SD card and module. Data on the left is serial and data on the right is the serial converted to readable data

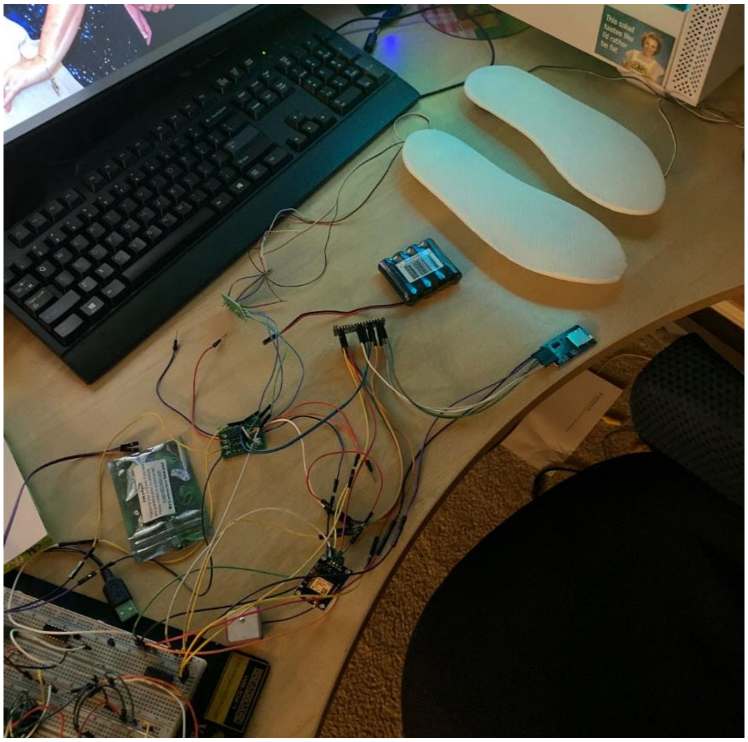


Figure 37: Integrating the whole system together

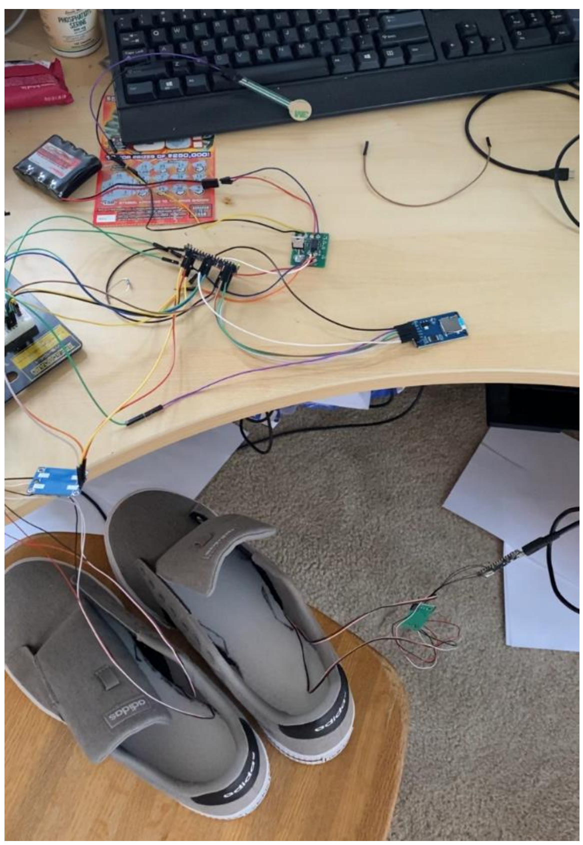


Figure 38: Integrating the system into a pair of shoes

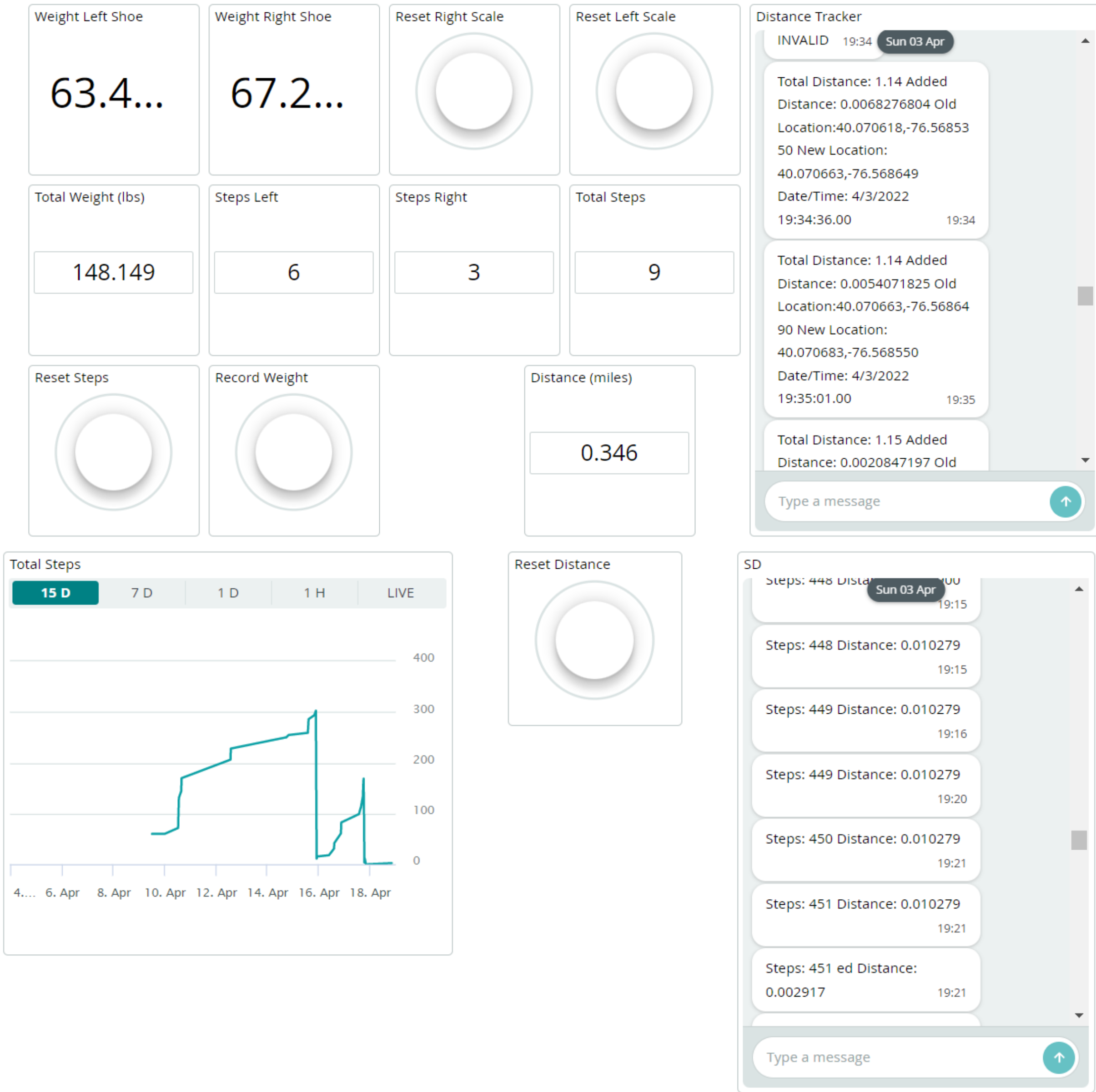


Figure 39: Example data of the fully integrated Smart Shoes

Figure 40: Making the Smart Shoes wireless

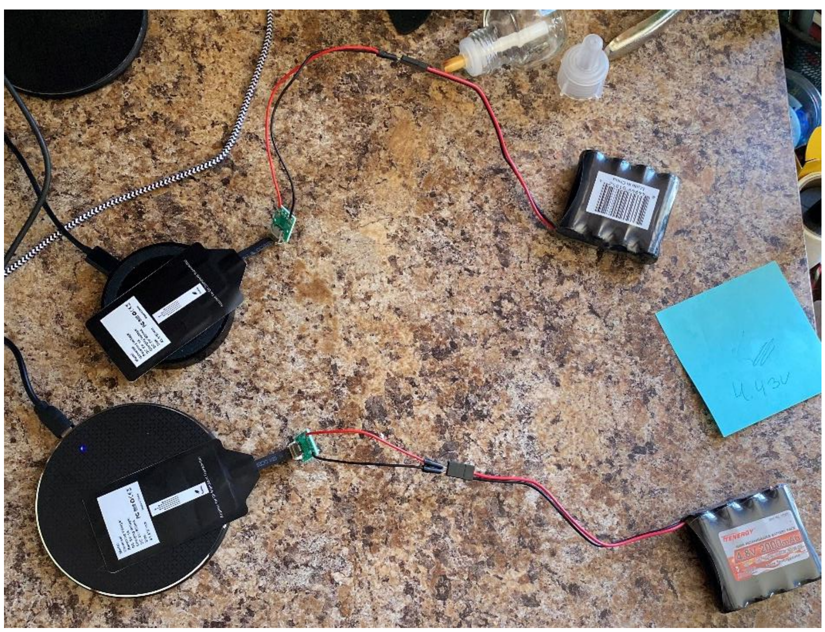


Figure 41: Testing wireless charging

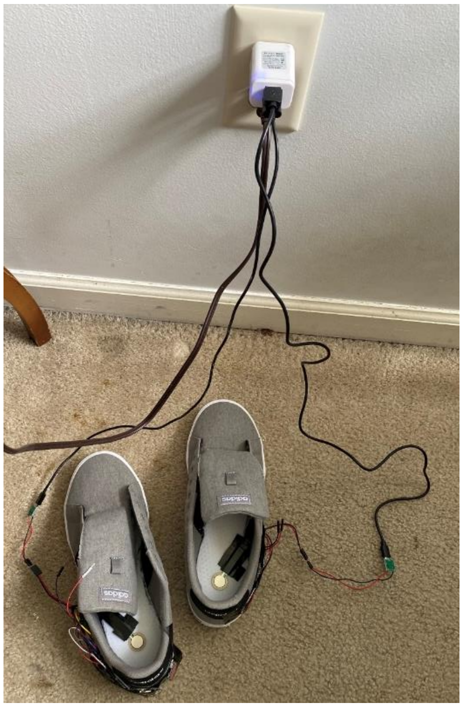


Figure 42: Testing wired charging as an alternative to wireless

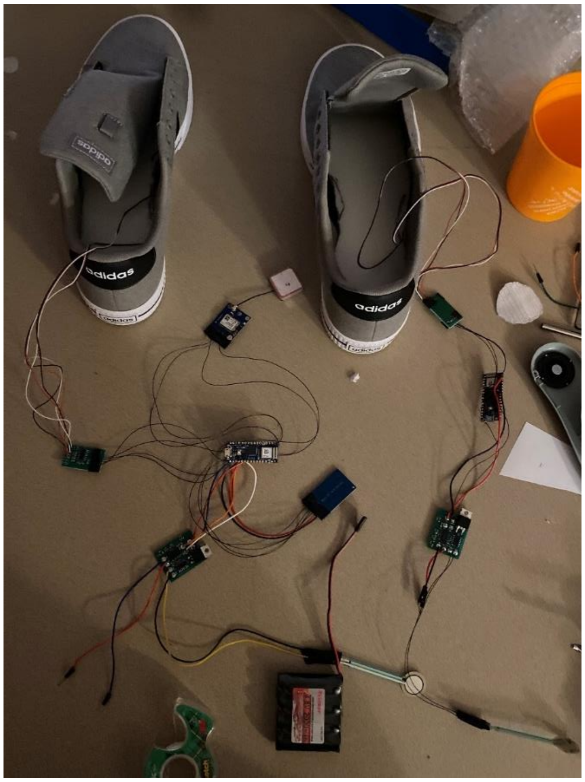


Figure 43: Both Shoes wireless

Figure 44: The final design of the Smart Shoes

Figure 45: Testing the Smart Shoes GPS

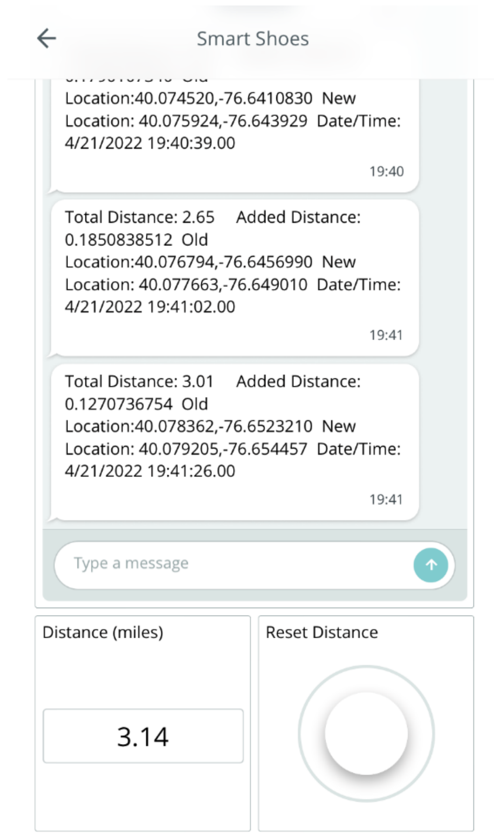


Figure 46: Results from testing the GPS

Figure 47: Weighing the Smart Shoes. One shoe weighs 1.3 pounds.

The Smart Shoes integration began with testing the weight sensors in Figure 31. After adjusting the calibration factor, the Smart Shoes produced an accurate weight measurement which can be seen in Figure 32. The next integration step was including the GPS and Power Switching Circuit which can be seen in Figure 33. The Power Switching Circuit had already been tested by itself and still worked with the other sub-systems attached. The GPS worked as expected and an accruing total distance variable was added to work with it which can be seen in Figure 34. One of the last sub-systems to integrate was the micro-SD card and module seen in Figure 35. The SD card at first only recorded serial data but then was converted to readable data. This process step is shown in Figure 36 with the data showing serially and then converted to readable data. Once all the main components were connected, all the sub-systems were tested again. Figure 37 shows all the electrical components connected with both shoes working. Figure 38 then shown how the weight sensors were integrated into a pair of shoes so that data such as steps could be measured. Example data of the fully integrated Smart Shoes can be seen in Figure 39. Additionally, testing was done during this integration process and on every sub-system.

After having the electronics work properly, the next step was to make them work completely wireless. This was achieved and can be seen in Figures 40 and 43. Figures 41 and 42 show different methods for charging the battery of the Smart Shoes and testing if the battery could be charged wirelessly. The process of imbedding the electronics inside the sole of the shoes began. While this may have been possible, the chance of components breaking over the weight of the user was considered, and alternative methods were developed. In the end, the electronics were taped to the side of the shoe for ease of access and display purposes. This final design is shown in Figure 44. From this point, system integration tests were performed on the Smart Shoes and their results can be viewed in the Appendix E

The Smart Shoes were validated against the engineering requirements in the tests that were done. One of the first tests that was performed was testing the weight sensors. The results to this test can be seen in Table 17. The largest issue with the weight sensors is that they do not settle out very well and jump in values. This made it hard to test the weight sensors, but from averaging some values the Smart Shoes were often able to measure a person's weight within +/-2% with the sensors by themselves. Other tests that were performed tested the functionality of the rechargeable battery and wireless receiver. The results for these tests are in Tables 14, 15, and 26. The results from these tests showed that the wireless receiver could receive power and charge the battery. It also proved that the battery worked as well. The only concern with wireless charging is that it could not fully charge the battery at 5 volts. In order to fully charge the battery, the wireless receiver would need to output a voltage closer to 5.5 volts or higher. This is possible but is not standard with wireless receivers.

The next tests dealt with the power switching circuit. Its ability to turn the micro-controller on and off was tested as well as the force sensors voltage. The results from these tests can be seen in Tables 16 and 19. Following those tests, the SD card module and memory storage system was tested. The results of these tests are in Tables 18 and 27 and prove that the Smart Shoes can store memory in a micro-SD card. The GPS component was tested to see how accurately it could measure distance. From the results in Table 25, the GPS was unable to measure distance within +/- 5% for the first three and a half miles. The results did show that the GPS becomes more accurate the farther the user walks. According to the trend of the test, The GPS should be able to measure distance within +/- 5% after 5 miles.

The rest of the tests done on the Smart Shoes were system tests with the Smart Shoes completely assembled. Table 24 shows how much data the smart shoes stored over 10 hours. This satisfies the engineering requirement of storing 4 GB of data because 4 GB of the micro-SD card is formatted to store measurement data. Table 23 validates that the Smart Shoes can measure weights from 0-400. This satisfies the engineering requirement but still has the issue of the weight data jumping around. The next test shows how long the smart shoes can last for. The results to this test can be seen in Table 22. The engineering requirement was that the Smart Shoes can last for 24 hours but the test shows that they can only last 10 hours. While the Smart Shoes do fall short of the 24 hours, they do last more than the average workday (8 hours) and a larger battery could be used to satisfy this engineering requirement in the future. Table 21 shows the results of testing how the Smart Shoes count steps. The test shows that the Smart Shoes may miss a step every now and then depending on how fast the user is walking. If the user is walking at a slow pace, the Smart Shoes can satisfy the engineering requirement of measuring steps within +/- 5%.

The last test was a whole system test that measured weight, distance, and stored weight. The results to this test can be viewed in Table 20. The results show that the Smart Shoes can measure weight within +/- 3% but this is by choosing an averaged value from a range of values at the output. Since the weight data jumps around, the Smart Shoes cannot measure weight within the engineering requirements of +/- 2%, but this may be possible in the future by averaging values and a more secure scale/shoe system. Currently, the Smart Shoes can let a user know what weight range they are in. The test also shows that the Smart Shoes can measure distance within 8% after 3.4 miles. With the trend, the Smart Shoes should be able to fulfil the engineering requirement of measuring distance within +/-5% after the user has walked 5 miles. This can be improved by setting the update distance to less than .06 miles or using a more precise GPS.

The Smart Shoes were unable to use Bluetooth communication due to the lack of Bluetooth apps available. Thus, the engineering requirement of using Bluetooth was not satisfied but the functionality of the device can be seen using Wi-Fi communication. The Smart Shoes cost around $182 to produce which is well below the engineering requirement of a less than $300 production cost. Another engineering requirement was that the system would fit inside a men's size 10 pair of shoes. This was slightly achieved as the Smart Shoes are one unit, but the electronics are on the outside of the shoes not the inside. The Smart Shoes can receive power wirelessly from a wireless receiver and satisfy the engineering requirement. The only issue to this is that a custom wireless receiver and charger will have to be used to produce a voltage of 5.5V and not 5V to charge the battery. Figure 47 shows that one Smart Shoe weights about .592 Kg which is equal to 1.3 pounds. Two Smart Shoes would weight around 2.6 pounds which is well below the engineering requirement of weighing less than 5 pounds. Finally, The Smart Shoes will turn off after less than 2 minutes of not being in use. Due to computation time in the microcontrollers, the Shoes will not turn off exactly after 1 minute of not being in use but will turn off in under 2 minutes and satisfy the engineering requirement.

In conclusion, the Smart Shoes completely met 6 of the 11 engineering requirements mentioned above and seen in Table 1. The other 5 engineering requirements are not far off from being met and future work can be done to meet them. The requirements were a challenge to meet and ambitious from the start. The final product is a testimony that this design works, and that Smart Shoes could potentially be the wearable technology of the future.

4. Project Management

A. Project Schedule (Gantt)

	Week 1 1/10/22	Week 2 1/17/22	Week 3 1/24/22	Week 4 1/31/22	Week 5 2/7/22	Week 6 2/14/22	Week 7 2/21/22	Week 8 2/28/22	Week 9 3/7/22	Week 10 3/14/22	Week 11 3/21/22	Week 12 3/28/22	Week 13 4/4/22	Week 14 4/11/22	Week 15 4/18/22	Week 16 4/25/22
Research																
Design Overall Product																
Design Subcircuit																
Build Prototype/ Debug																
Test Product																
Write Report																
Create presentation																

Table 11: Project Schedule

The Smart Shoes project was proposed December 15th, 2021. After the project was reviewed and approved, the project launched on January 10th, 2022. From this point, the project took 4 months to complete. The breakdown of work during each week can be seen in Table 11. All of the work contributed to this project was done by Tyler Rupp. He devoted approximately 138 hours to the project with an average of 9 hours per week. Research on different electrical components, shoes, and software were done during the first 10 weeks of the project with most of it being done in the first 3 weeks. Once most of the research was finished the design of the Smart Shoes began. The design phase was accomplished in the first 5 weeks of the project with some small additions added later. After the overall design was made, focus was set on designing the Power Switching Circuit (Subcircuit). This process took about 5 weeks and overlapped with building a prototype and debugging any issues. The Power Switching Circuit and prototype were completed by week 10 of the project. The fully functional prototype was tested over 5 weeks and the results are recorded in Appendix E. During the course of the project, everything was recorded in this report and formatted for understanding. Finally, in the last 4 weeks of the Smart Shoes Project a presentation was made for a Capstone Conference.

B. Project Budget

Item Description	Unit/Batch	Price/Unit	Price	Quantity	Cost
Arduino Nano 33 IoT	ea	\$20.00	\$20.00	2	\$40.00
50kg Human Scale Load Cell Weighing Sensor	8	\$1.50	\$12.00	1	\$12.00
Wireless Charging Adapter Qi Charger Receiver	ea	\$10.98	\$10.98	2	\$21.96
Resistors, Integrated circuits	ea	\$10.95	\$10.95	1	\$10.95
Rechargeable Battery Pack	2	\$8.50	\$17.00	1	\$17.00
GPS Module	ea	\$8.69	\$8.69	1	\$8.69
2PCS Micro SD Storage Board Memory	ea	\$6.59	\$6.59	1	\$6.59
64GB microSD Card	ea	\$8.99	\$8.99	1	\$8.99
Micro USB to DIP Adapter	10	\$0.53	\$5.29	1	\$5.29
Pair of Shoes	ea	\$40.00	\$40.00	1	\$40.00
PCB	ea	\$1.00	\$1.00	10	\$10.00
Force Sensors	ea	\$12.50	\$12.50	2	\$25.00
Total Cost					\$181.47

Table 12: Project Budget

The budget goal of the project was to create a pair of Smart Shoes under \$300. The total cost of the project was \$181.47 which is broken down in Table 12. No funding was received for this project and all expenses were paid for by the team.

5. Summary and Conclusion

A. Summary and Conclusion

The Smart Shoes were successful in measuring a person's weight, counting steps, tracking distance, and storing data. While not all the engineering requirements were met, many of them were close to being satisfied. In many tests, the Smart Shoes could measure a person's weight within +/- 3% even with the values jumping around. Another success is the Smart Shoes app that was able to display all the data from the shoes to a smart device. This made viewing the data very easy so that any user could understand what they were looking at. The electronics for the Smart Shoes ended up being taped to the side of the shoe instead of in the sole of the shoe. While this is not ideal, some people have expressed that they like being able to see the circuitry and that it could be used intentionally to have the shoe look futuristic. The Smart Shoes are light weight (less than 3 pounds) and can last up to 10 hours which is more than most people walk in a day. When it comes to counting steps, the Smart Shoes do this every time the user lifts weight off the shoe. This eliminates cheating but is only accurate up to a certain speed of walking. The Power Switching Circuit that was designed from scratch works very well and will help to save battery life in the Smart Shoes.

In conclusion, with some further work in manufacturing and overall design, these Smart Shoes could be a product on the market for people to buy. The system should be able to work in many different shoes and styles. These shoes will further wearable technology and hopefully become a reality in the upcoming years. People will soon be able to measure their weight wherever and whenever they want, be able to identify their stride and walking patterns, and track how far they've walked without their phone.

6. Future Considerations

Many ideas and upgrades can be made to the Smart Shoes in the future. The first being a more accurate weight measuring system. This can be achieved by more accurate weight sensors and making sure that the scales are an equal distance form the ground on both shoes. The next upgrade would be having a more accurate GPS system and having the Smart Shoes measure velocity as well. This could allow the Smart Shoes to know when the user is walking/running or driving in a car. Another future consideration would be to make the Smart Shoes waterproof and create a more secure and robust placement of the electronics used. This would allow users to walk in the rain or puddle and not have to worry about ruining their shoes. Further work could be done to have the Smart Shoes have their own app that users could login into and have all their data saved on the cloud. Additionally, a larger battery could be used so that the Smart Shoes would not need to be charged every 10 hours.

7. References

[1] A. Piau *et al*, "A Smart Shoe Insole to Monitor Frail Older Adults' Walking Speed: Results of Two Evaluation Phases Completed in a Living Lab and Through a 12-Week Pilot Study," *JMIR mHealth and uHealth,* vol. 9, *(7),* pp. e15641, 2021.

[2] P. H. Truong *et al*, "Wearable System for Daily Activity Recognition Using Inertial and Pressure Sensors of a Smart Band and Smart Shoes," *International Journal of Computers, Communications & Control,* vol. 14, *(6),* pp. 726-742, 2019.

[3] H. Eom *et al*, "Deep Learning-Based Optimal Smart Shoes Sensor Selection for Energy Expenditure and Heart Rate Estimation," *Sensors (Basel, Switzerland),* vol. 21, *(21),* pp. 7058, 2021.

[4] N. Shah *et al*, "Walking pattern analysis using deep learning for energy harvesting smart shoes with IoT," *Neural Computing & Applications,* vol. 33, *(18),* pp. 11617-11625, 2021.

[5] B. Eskofier *et al*, "An Overview of Smart Shoes in the Internet of Health Things: Gait and Mobility Assessment in Health Promotion and Disease Monitoring," *Applied Sciences,* vol. 7, *(10),* pp. 986, 2017.

[6] *Nike HyperAdapt 1.0*. HyperAdapt 1.0. (n.d.). Retrieved May 3, 2022, from https://www.nike.com/launch/t/hyperadapt-1-0

[7] Under Armour. (2022, March 27). *Men's UA HOVR™ Sonic 4 running shoes*. Under Armour. Retrieved May 3, 2022, from https://www.underarmour.com/en-us/p/running/mens_ua_hovr_sonic_4_running_shoes/3023543.html?dwvar_3023543_color=002&start=0&breadCrumbLast=Running

[8] *Feetme insoles*. FeetMe. (2021, September 21). Retrieved May 3, 2022, from https://feetmehealth.com/insoles/

[9] *NURVV - Run Insole Activity Tracker GPS*. Best Buy. (n.d.). Retrieved May 3, 2022, from https://healthcare.bestbuy.com/site/nurvv-run-insole-activity-tracker-gps-medium-us-womens-8-5-9-5-mens-7-5-8-5-gray/6397952.p?skuId=6397952

[10] Veness, C. (n.d.). *Movable type scripts*. Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript. Retrieved May 3, 2022, from https://www.movable-type.co.uk/scripts/latlong.html

[11] Neamen, D. A. (2010). *Microelectronics: Circuit analysis and design*. McGraw-Hill.

8. Appendix A: Parts List

Parts List	Amount
Arduino Nano 33 IoT	2
50kg Human Scale Load Cell Weighing Sensor	4
Wireless Charging Adapter Qi Charger Receiver	2
Resistors, Integrated circuits	18
Rechargeable Battery Pack	2
GPS Module	1
2PCS Micro SD Storage Board Memory	1
64GB microSD Card	1
Micro USB to DIP Adapter	2
Pair of Shoes	1
PCB	2
Force Sensors	2

Table 13: Parts List

9. Appendix B: Datasheets

INCHANGE Semiconductor — isc Product Specification

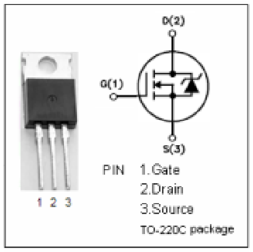
isc N-Channel MOSFET Transistor — IRFZ44N

FEATURES

- Drain Current –I_D=49A@ T_C=25℃
- Drain Source Voltage-
  : V_{DSS}= 55V(Min)
- Static Drain-Source On-Resistance
  : $R_{DS(on)}$ = 0.032Ω (Max)
- Fast Switching

DESCRIPTION

- Designed for low voltage, high speed switching applications in power supplies, converters and power motor controls, these devices are particularly well suited for bridge circuits where diode speed and commutating safe operating areas are critical and offer additional safety margin against unexpected voltage transients.

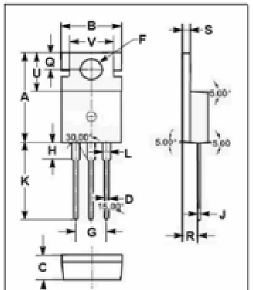


ABSOLUTE MAXIMUM RATINGS(T_a=25℃)

SYMBOL	PARAMETER	VALUE	UNIT
V_{DSS}	Drain-Source Voltage	55	V
V_{GS}	Gate-Source Voltage-Continuous	±20	V
I_D	Drain Current-Continuous	49	A
I_{DM}	Drain Current-Single Pluse (t_p≤10μs)	160	A
P_D	Total Dissipation @T_C=25℃	94	W
T_J	Max. Operating Junction Temperature	175	℃
T_{stg}	Storage Temperature	-55~175	℃

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	MAX	UNIT
$R_{th\ j\text{-}c}$	Thermal Resistance, Junction to Case	1.5	℃/W
$R_{th\ j\text{-}a}$	Thermal Resistance, Junction to Ambient	62	℃/W



DIM	mm MIN	mm MAX
A	15.70	15.90
B	9.90	10.10
C	4.20	4.40
D	0.70	0.90
F	3.40	3.60
G	4.98	5.18
H	2.70	2.90
J	0.44	0.46
K	13.20	13.40
L	1.10	1.30
Q	2.70	2.90
R	2.50	2.70
S	1.29	1.31
U	6.45	6.65
V	8.66	8.86

isc website: www.iscsemi.cn 1

Figure 48: NMOSFET datasheet page 1

Equation 6: NMOS Linear Mode IDS $\{I_{DS} = \mu_n C_{ox}\left[(V_{GS} - V_T)V_{DS} - \frac{(V_{DS})^2}{2}\right](1 + \lambda V_{DS})\}$

Equation 7: NMOS Saturation Mode IDS $\{I_{DS} = \frac{\mu_n}{2} C_{ox} \frac{W}{L}(V_{GS} - V_T)^2(1 + \lambda V_{DS})\}$

Equations 6 and 7 were used to help choose a MOSFET that would be in saturation mode with a V_{GS} of 4 V or less [11].

INCHANGE Semiconductor **isc Product Specification**

isc N-Channel MOSFET Transistor **IRFZ44N**

ELECTRICAL CHARACTERISTICS

T_C=25℃ unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
$V_{(BR)DSS}$	Drain-Source Breakdown Voltage	V_{GS}= 0; I_D= 0.25mA	55		V
$V_{GS(th)}$	Gate Threshold Voltage	V_{DS}= V_{GS}; I_D= 0.25mA	2	4	V
$R_{DS(on)}$	Drain-Source On-Resistance	V_{GS}= 10V; I_D= 25A		0.032	Ω
I_{GSS}	Gate-Body Leakage Current	V_{GS}= ±20V;V_{DS}= 0		±100	nA
I_{DSS}	Zero Gate Voltage Drain Current	V_{DS}= 55V; V_{GS}= 0 V_{DS}= 55V; V_{GS}= 0; T_j= 150℃		25 250	μA
V_{SD}	Forward On-Voltage	I_S= 25A; V_{GS}= 0		1.3	V

Figure 49: NMOSFET datasheet page 2

Product Folder Order Now Technical Documents Tools & Software Support & Community

TEXAS INSTRUMENTS

SN5400, SN54LS00, SN54S00
SN7400, SN74LS00, SN74S00

SDLS025D – DECEMBER 1983 – REVISED MAY 2017

SNx400, SNx4LS00, and SNx4S00 Quadruple 2-Input Positive-NAND Gates

1 Features

- Package Options Include:
  - Plastic Small-Outline (D, NS, PS)
  - Shrink Small-Outline (DB)
  - Ceramic Flat (W)
  - Ceramic Chip Carriers (FK)
  - Standard Plastic (N)
  - Ceramic (J)
- Also Available as Dual 2-Input Positive-NAND Gate in Small-Outline (PS) Package
- Inputs Are TTL Compliant; V_{IH} = 2 V and V_{IL} = 0.8 V
- Inputs Can Accept 3.3-V or 2.5-V Logic Inputs
- SN5400, SN54LS00, and SN54S00 are Characterized For Operation Over the Full Military Temperature Range of –55°C to 125°C

2 Applications

- AV Receivers
- Portable Audio Docks
- Blu-Ray Players
- Home Theater
- MP3 Players or Recorders
- Personal Digital Assistants (PDAs)

3 Description

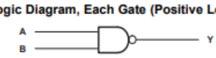
The SNx4xx00 devices contain four independent, 2-input NAND gates. The devices perform the Boolean function $Y = \overline{A \cdot B}$ or $Y = \overline{A} + \overline{B}$ in positive logic.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
SN74LS00DB	SSOP (14)	6.20 mm × 5.30 mm
SN7400D, SN74LS00D, SN74S00D	SOIC (14)	8.65 mm × 3.91 mm
SN74LS00NSR	PDIP (14)	19.30 × 6.35 mm
SNJ5400J, SNJ54LS00J, SNJ54S00J	CDIP (14)	19.56 mm × 6.67 mm
SNJ5400W, SNJ54LS00W, SNJ54S00W	CFP (14)	9.21 mm × 5.97 mm
SN54LS00FK, SN54S00FK	LCCC (20)	8.89 mm × 8.89 mm
SN7400NS, SN74LS00NS, SN74S00NS	SO (14)	10.30 mm × 5.30 mm
SN7400PS, SN74LS00PS	SO (8)	6.20 mm × 5.30 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Logic Diagram, Each Gate (Positive Logic)



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Figure 50: SN74LS00NS Data Sheet

LM339, LM239, LM139, LM2901 Quad Differential Comparators

1 Features

- Wide Supply Ranges
  - Single Supply: 2 V to 36 V (Tested to 30 V for Non-V Devices and 32 V for V-Suffix Devices)
  - Dual Supplies: ±1 V to ±18 V (Tested to ±15 V for Non-V Devices and ±16 V for V-Suffix Devices)
- Low Supply-Current Drain Independent of Supply Voltage: 0.8 mA (Typical)
- Low Input Bias Current: 25 nA (Typical)
- Low Input Offset Current: 3 nA (Typical) (LM139)
- Low Input Offset Voltage: 2 mV (Typical)
- Common-Mode Input Voltage Range Includes Ground
- Differential Input Voltage Range Equal to Maximum-Rated Supply Voltage: ±36 V
- Low Output Saturation Voltage
- Output Compatible With TTL, MOS, and CMOS
- On Products Compliant to MIL-PRF-38535, All Parameters Are Tested Unless Otherwise Noted. On All Other Products, Production Processing Does Not Necessarily Include Testing of All Parameters.

2 Applications

- Industrial
- Automotive
  - Infotainment and Clusters
  - Body Control Modules
- Power Supervision
- Oscillators
- Peak Detectors
- Logic Voltage Translation

3 Description

The LMx39x and the LM2901x devices consist of four independent voltage comparators that are designed to operate from a single power supply over a wide range of voltages. Operation from dual supplies also is possible, as long as the difference between the two supplies is 2 V to 36 V, and V_{CC} is at least 1.5 V more positive than the input common-mode voltage. Current drain is independent of the supply voltage. The outputs can be connected to other open-collector outputs to achieve wired-AND relationships.

The LM139 and LM139A devices are characterized for operation over the full military temperature range of –55°C to +125°C. The LM239 and LM239A devices are characterized for operation from –25°C to +85°C. The LM339 and LM339A devices are characterized for operation from 0°C to 70°C. The LM2901, LM2901AV, and LM2901V devices are characterized for operation from –40°C to +125°C.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM139x	CDIP (14)	21.30 mm × 7.60 mm
	LCCC (20)	8.90 mm × 8.90 mm
	CFP (14)	9.20 mm × 6.29 mm
LM139x, LM239x, LM339x, LM2901x	SOIC (14)	8.70 mm × 3.90 mm
LM239, LM339x, LM2901	PDIP (14)	19.30 mm × 6.40 mm
LM239, LM2901	TSSOP (14)	5.00 mm × 4.40 mm
LM339x, LM2901	SO (14)	10.20 mm × 5.30 mm
LM339x	SSOP (14)	6.50 mm × 5.30 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Simplified Schematic

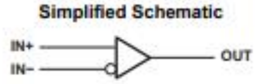


Figure 51: LM339D Data Sheet

10. Appendix E: Testing Documents

Test Author:	Tyler Rupp		
Test Name:	Wireless Power Receiver	**Test ID #:**	1
Description:	Testing if the wireless power receiver can output 5V	**Test Type:**	☐ Component X Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	March 14th, 2022
Hardware/Software Version:	NewGF Wireless Charger Tasumato Wireless Receiver model MU-A014	**Time:**	6:19 pm

Test Setup:
A wireless charger will be powered, and the wireless receiver will be placed on top of the charger at various lengths above. The test will determine how far the wireless receiver can be placed above the charger until the receiver has an output value of less than 5V.

	INPUTS	**OUTPUTS**	**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Wireless charger (mm)	Micro USB port (V) (measured)	5V	>5V		
1	0mm	5V	X			
2	1mm	5V	X			
3	2mm	5V	X			
4	3mm	5V	X			
5	4mm	5V	X			

Test Summary:
The wireless receiver can output 5V at different lengths above the wireless charger. To charge the 5V battery more than 5V may be needed to fully charge.

Table 14: Wireless Power Receiver test results

Test Author:	Tyler Rupp		
Test Name:	Rechargeable Battery	**Test ID #:**	2
Description:	Testing if the battery can charge and power circuitry	**Test Type:**	☐ Component X Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	March 14th, 2022
Hardware/Software Version:	Tenergy NiMH Receiver RX Battery	**Time:**	6:50 pm

Test Setup:
A current meter will be inserted in the circuit and used to measure if the battery is providing current. The tests will connect the battery to a load and test the battery ability to power circuitry. VDD =5V. Handheld Fluke amp meter.

	INPUTS		**OUTPUTS**		**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Battery	Load (ohms)	Current (ma) (calculated)	Current (ma) (measured)	>.001ma			
1	5V	500 Ω	10ma	9.81ma	X			
2	5V	1K Ω	5ma	4.89 ma	X			
3	5V	2K Ω	2.5ma	2.47 ma	X			
4	5V	10K Ω	.5ma	.498 ma	X			
5	5V	100K Ω	.05ma	.047 ma	X			

Test Summary:
The Battery was able to provide currents to different loads. The battery is fully charged at 5.12 V so it was drained to 5V for this test

Table 15: Rechargeable Battery test results

Test Author:	Tyler Rupp								
Test Name:	Power Switching Circuit					**Test ID #:**	3		
Description:	Testing the logic of the Power Switching Circuit					**Test Type:**	☐ Component X Subsystem ☐ System		
Name of Tester:	Tyler Rupp					**Date:**	April 5th, 2022		
Hardware/Software Version:	Tenergy NiMH Receiver RX Battery Arduino Nano 33 IoT Model ABX00027 Bolsen Tech FSR402 Pressure sensor					**Time:**	4:05pm		
Test Setup: **The power switching circuit will be connected to a force sensor and a micro-controller. Pressing the force sensor (to 3V or less) should turn the micro-controller on and after the force sensor has not been pressed for a minute the micro-controller should turn itself off. VDD = 5V. Handheld Fluke voltmeter**									
	INPUTS			**OUTPUTS**		**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Force sensor (V)	Battery (V)	Time(s)	Microcontroller State (expected)	Microcontroller State (measured)	Same as expected			
1	5V	5V	60s	on	on	X			
2	4V	5V	60s	on	on	X			
3	2.9V	5V	60s	Off after 60s	Off (after 82)	X			
4	2V	5V	60s	Off after 60s	Off (after 84)	X			
5	1V	5V	60s	Off after 60s	Off (after 79)	X			
Test Summary: The Power Switching Circuit worked as expected but took longer than expected to turn off the microcontroller. This could be due to boot up time and additional time to process all the code in the microcontroller.									

Table 16: Power Switching Circuit test results

Test Author:	Tyler Rupp		
Test Name:	Weight Sensors	**Test ID #:**	4
Description:	Testing the accuracy of the weight sensors	**Test Type:**	X Component ☐ Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	March 24th, 2022
Hardware/Software Version:	Arduino Nano 33 IoT Model ABX00027 2 ShangHJ 50kg weighing sensors + HX711 A/D	**Time:**	2:46 pm

Test Setup:
Attach the weight sensors to the plastic sole and connect them to a micro-controller. Place different known weights on the scale and see if they are within +/- 2% of the known weight.

	INPUTS	**OUTPUTS**	**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Weight (lbs)	Digital display	+/- 2%			
1	98.7 lbs	101.3 lbs		X		Values jumped around
2	122.4 lbs	124.2 lbs	X			Values jumped around
3	151.2 lbs	148.5 lbs	X			Values jumped around
4	192.6 lbs	196.4 lbs	X			Values jumped around
5	242.3 lbs	239.8 lbs	X			Values jumped around

Test Summary:
The weight shown on the digital display tends to jump around and was difficult to settle on one output. Sometimes the weight displayed would be out of the +/- 2% range but then settle back in it.

Table 17: Weight Sensors test results

Test Author:	Tyler Rupp		
Test Name:	Memory Weight Storage	**Test ID #:**	5
Description:	Testing the ability to store weight data and send weight data from storage	**Test Type:**	☐ Component X Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	April 7th, 2022
Hardware/Software Version:	Micro Center 64 GB microSDXC card Maxmoral Micro SD Storage Board module Arduino Nano 33 IoT Model ABX00027	**Time:**	3:08 pm

Test Setup:
Connect the SD storage module to the micro-controller and store weight sensor data on the micro-SD. After storing data, send the stored data to a smart device. The test will pass if the stored data is the same as displayed on the smart device

	INPUTS	OUTPUTS	Pass	Fail	N/A	COMMENTS
Test	Weight sensor data to micro-SD	Weight sensor data on Smart Device	=			
1	97.9 lbs	97.9 lbs	X			
2	123.2 lbs	123.2 lbs	X			
3	149.6 lbs	149.6 lbs	X			
4	193.7 lbs	193.7 lbs	X			
5	241.1 lbs	241.1 lbs	X			

Test Summary:
The data that was stored in the micro-SD is that same data that is sent to the smart device

Table 18: Memory Weight Storage test results

<table>
<tr><td>Test Author:</td><td colspan="6">Tyler Rupp</td></tr>
<tr><td>Test Name:</td><td>Force Sensor</td><td colspan="3">Test ID #:</td><td colspan="2">6</td></tr>
<tr><td>Description:</td><td>Testing the voltage range of the force sensor</td><td colspan="3">Test Type:</td><td colspan="2">X Component
☐ Subsystem
☐ System</td></tr>
<tr><td>Name of Tester:</td><td>Tyler Rupp</td><td colspan="3">Date:</td><td colspan="2">April 7th, 2022</td></tr>
<tr><td>Hardware/Software Version:</td><td>Arduino Nano 33 IoT Model ABX00027
Bolsen Tech FSR402 Pressure sensor</td><td colspan="3">Time:</td><td colspan="2">5:34 pm</td></tr>
<tr><td colspan="7">Test Setup:
Place the force sensor on the sole of the Smart Shoe and connect it to a micro-controller that will measure the voltage. By placing different weight on the force sensor, the different voltage ranges will be observed from 5V-0V.</td></tr>
<tr><td></td><td>INPUTS</td><td>OUTPUTS</td><td>Pass</td><td>Fail</td><td>N/A</td><td>COMMENTS</td></tr>
<tr><td>Test</td><td>Weight from pressure (lbs)</td><td>Voltage (v)</td><td>0-5V</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>0 lbs</td><td>3.78 V</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>123.2 lbs</td><td>1.38 V</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>149.6 lbs</td><td>1.36 V</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>193.7 lbs</td><td>1.32 V</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>241.1 lbs</td><td>1.29 V</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Test Summary:
The force sensor is only made for 30 pounds or less so anything above 30 pounds will give a value close to 1.35 V</td></tr>
</table>

Table 19: Force Sensor test results

Test Author:	Tyler Rupp		
Test Name:	System Test	**Test ID #:**	7
Description:	Overall test of the sub-systems connected	**Test Type:**	☐ Component ☐ Subsystem X System
Name of Tester:	Tyler Rupp	**Date:**	April 14th/21st,2022
Hardware/Software Version:	Arduino Nano 33 IoT Model ABX00027, Bolsen Tech FSR402 Pressure sensor, Micro Center 64 GB microSDXC card, Maxmoral Micro SD Storage Board module, Arduino IDE 1.8.19, HiLetgo NEO-6M GPS module version 2, 4ShangHJ 50kg weighing sensors + HX711 A/D, Tenergy NiMH Receiver RX Battery, NewGF Wireless Charger, Tasumato Wireless Receiver model MU-A014	**Time:**	3:00pm/7:37pm

Test Setup:
Have all the subsystems connected and test that the data can be sent to a smart device and the system shuts off after 1 minute. Check the weight data, GPS data, and stored data on the smart device. Use known values to compare to the data on the smart device.

	INPUTS			**OUTPUTS displayed on smart device**				**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Weight(lbs)	Distance	Time(s) after force sensor	Weight(lbs)	Distance	Stored weight data	State of Smart Shoes (on/off)	+/- 2%			
1	98.2 lbs	.3 miles	10s	100.6 lbs	.112 miles	100.6 lbs	on		X		Weight: 2.4% error Distance: 63% error
2	124.5 lbs	.8 miles	50s	126.8 lbs	.537 miles	126.8 lbs	on		X		Weight: 1.9% error Distance: 33% error
3	151.3 lbs	1.6 miles	100s	154.9 lbs	1.37 miles	154.9 lbs	on		X		Weight: 2.4% error Distance: 14% error
4	194.2 lbs	2.2 miles	150s	195.7 lbs	1.82 miles	195.7 lbs	off		X		Weight: 0.8% error Distance: 17% error
5	242.5 lbs	3.4 miles	200s	246.3 lbs	3.14 miles	246.3 lbs	off		X		Weight: 1.6% error Distance:7.6% error

Test Summary:
The weight shown tends to jump around and was difficult to settle on one output. Since the GPS calculates distance minimally at .06 miles it was more accurate the father the Smart Shoes went. Both Smart Shoes turned off after the user took off the shoes.

Table 20: Whole System integration test results

Test Author:	Tyler Rupp		
Test Name:	Steps Taken	**Test ID #:**	8
Description:	Testing how accurately the system can record steps	**Test Type:**	☐ Component ☐ Subsystem X System
Name of Tester:	Tyler Rupp	**Date:**	April 14th, 2022
Hardware/Software Version:	Complete Smart Shoes and smart device	**Time:**	7:00pm

Test Setup:
The test will have the user take steps and compare it to the steps displayed on the smart device. The test will be pass if the Smart Shoes can measure within +/- 5% of the steps taken.

	INPUTS	OUTPUTS	Pass	Fail	N/A	COMMENTS
Test	Steps	Steps displayed	+/- 5%			
1	10 steps	9 steps		X		
2	50 steps	48 steps	X			
3	100 steps	96 steps	X			
4	200 steps	193 steps	X			
5	500 steps	489 steps	X			

Test Summary:
In order for the steps to register the user will have to walk at a very slow pace. If the user walks at a slow pace the test will pass but if the user walks too fast, then the percent error could increase exponentially.

Table 21: Steps test results

Test Author:	Tyler Rupp		
Test Name:	Battery Life	**Test ID #:**	9
Description:	Testing how long the Smart Shoes can be on for	**Test Type:**	☐ Component ☐ Subsystem X System
Name of Tester:	Tyler Rupp	**Date:**	April 9-10th, 2022
Hardware/Software Version:	Complete Smart Shoes and smart device	**Time:**	10:00am-10:00am

Test Setup:
Fully charge the battery of the Smart Shoes. Keep the Smart Shoes on for 24 hours by wearing them or by placing a weight on them. The test will pass if the shoes are still on after 24 hours.

	INPUTS	OUTPUTS	Pass	Fail	N/A	COMMENTS
Test	Time (hours)	State of Shoes (on/off)	on			
1	5	on	X			
2	10	on	X			
3	15	off		X		
4	20	off		X		
5	24	off		X		

Test Summary:
The Smart Shoes only last around 10.5 hours before the battery in them dies. A solution to this would be to either get a larger battery or write more power efficient code.

Table 22: Battery Life test results

Test Author:	Tyler Rupp		
Test Name:	Different Weights	**Test ID #:**	10
Description:	Testing the accuracy of the Smart Shoes weight sensors	**Test Type:**	☐ Component ☐ Subsystem X System
Name of Tester:	Tyler Rupp	**Date:**	April 12th, 2022
Hardware/Software Version:	Complete Smart Shoes and smart device	**Time:**	4:00pm

Test Setup:
Have different weights put on the Smart Shoes and check the accuracy of the Smart Shoes. There should be a range of weights from 100-400 pounds. The test will pass if the Smart Shoes can measure the weight within +/-2%

	INPUTS		OUTPUTS		Pass	Fail	Pass	Fail	COMMENTS
Test	weight (lbs)		Weight Displayed on Smart Shoes		**+/- 2%**		**+/- 2%**		
1	100 lbs	200 lbs	102.4 lbs	206.5 lbs		X		X	
2	120 lbs	250 lbs	123.5 lbs	247.6 lbs		X	X		
3	140 lbs	300 lbs	138.1 lbs	311.4 lbs	X			X	
4	160 lbs	350 lbs	165.3 lbs	354.3 lbs		X	X		
5	180 lbs	400 lbs	183.2 lbs	413.3 lbs	X			X	

Test Summary:
The weight sensors don't always measure within +/- 2%. There could also be error in the input weights (lifting weights were used).

Table 23: Different Weights test results

Test Author:	Tyler Rupp		
Test Name:	Stored Data Capacity	**Test ID #:**	11
Description:	Testing the storage capacity of the Smart Shoes	**Test Type:**	☐ Component ☐ Subsystem X System
Name of Tester:	Tyler Rupp	**Date:**	April 8th, 2022
Hardware/Software Version:	Complete Smart Shoes and smart device	**Time:**	1:00pm-11:00pm

Test Setup:
Wear the Smart Shoes over the course of multiple days. After the time, see how much storage is left in micro-SD. The Smart Shoes should be able to store data for at least 10 hours.

	INPUTS		**OUTPUTS**		**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Using the Smart Shoes (hours)		Storage left (GB)		>0GB			
1	1 hour	6 hours	3.9 GB	3.9 GB	X			
2	2 hours	7 hours	3.9 GB	3.8 GB	X			
3	3 hours	8 hours	3.9 GB	3.8 GB	X			
4	4 hours	9 hours	3.9 GB	3.8 GB	X			
5	5 hours	10 hours	3.9 GB	3.8 GB	X			

Test Summary:
About .1 GBs were used in 10 hours of storing data. Some hours more data was stored than others.

Table 24: Stored Data Capacity test results

Test Author:	Tyler Rupp		
Test Name:	Distance Tracker	**Test ID #:**	12
Description:	Testing how accurately the system can measure distance	**Test Type:**	X Component ☐ Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	April 21st, 2022
Hardware/Software Version:	Complete Smart Shoes and smart device	**Time:**	7:37 pm

Test Setup:
The test will have the user walk different distances and compare it to the distance displayed on the smart device. The test will be pass if the Smart Shoes can measure within +/- 5% of the distance travelled.

	INPUTS		**OUTPUTS**		**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Distance Traveled		Distance displayed	Distance displayed	+/- 5%			
1	.5 miles	2.2 miles	.274 miles	1.816 miles		X		45% error at .5 miles
2	.8 miles	2.4 miles	.537 miles	2.186 miles		X		
3	1.5 mile	2.8 miles	.975 miles	2.535 miles		X		
4	1.6 miles	3.1 miles	1.37 miles	2.832 miles		X		
5	1.9 miles	3.4 miles	1.633 miles	3.14 miles		X		7.6% error at 3.4 miles

Test Summary:
The Smart Shoes tend to become more accurate the farther the user walks. This is because the GPS on the Smart Shoes can only update every .06 miles. After the user has walked more than 5 miles, it should be within +/- 5%.

Table 25: Distance tracking test results

Test Author:	Tyler Rupp		
Test Name:	Rechargeable Battery Recharging	**Test ID #:**	13
Description:	Testing if the battery can Recharge	**Test Type:**	X Component ☐ Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	March 14th, 2022
Hardware/Software Version:	Tenergy NiMH Receiver RX Battery	**Time:**	7:24 pm

Test Setup:
The test will have the battery connected to the wireless receiver while the receiver is passing current. VDD =5V. Handheld Fluke amp meter. A current meter will be inserted in the circuit and used to measure if the battery is being charged.

	INPUTS	**OUTPUTS**	**Pass**	**Fail**	**N/A**	**COMMENTS**
Test	Wireless Charger	Current (ma) (measured)	>.001ma			
1	5V	75 mA	X			
2	5V	74 mA	X			
3	5V	75mA	X			
4	5V	76 mA	X			
5	5V	75 mA	X			

Test Summary:
The wireless receiver charges the battery at about 75 mA. A possible issue is that the voltage needs to be higher than 5V to fully charge the battery. In this case a custom wireless charger and receiver would need to be ordered.

Table 26: Rechargeable Battery recharge test results

Test Author:	Tyler Rupp		
Test Name:	Memory Storage	**Test ID #:**	14
Description:	Testing the ability to store data and send data from storage	**Test Type:**	X Component ☐ Subsystem ☐ System
Name of Tester:	Tyler Rupp	**Date:**	April 5th, 2022
Hardware/Software Version:	Micro Center 64 GB microSDXC card Maxmoral Micro SD Storage Board module Arduino Nano 33 IoT Model ABX00027	**Time:**	7:49 pm

Test Setup:
Connect the SD storage module to the micro-controller and store text data on the micro-SD. After storing data, read the data from the micro-SD. The test will pass if the stored data is the same as input data.

	INPUTS	OUTPUTS	Pass	Fail	N/A	COMMENTS
Test	Text sent to Micro-controller	Stored text from micro-SD card	=			
1	“1”	“1”	X			
2	“Hello”	“Hello”	X			
3	“A string of words”	“A string of words”	X			
4	“Letters and Numbers123”	“Letters and Numbers123”	X			
5	“Special Characters?!*&@#$%”	“Special Characters?!*&@#$%”	X			

Test Summary:-
The microcontroller can store and read data from the micro-SD card.

Table 27: Memory Storage test results

11. Appendix F: Customer Survey

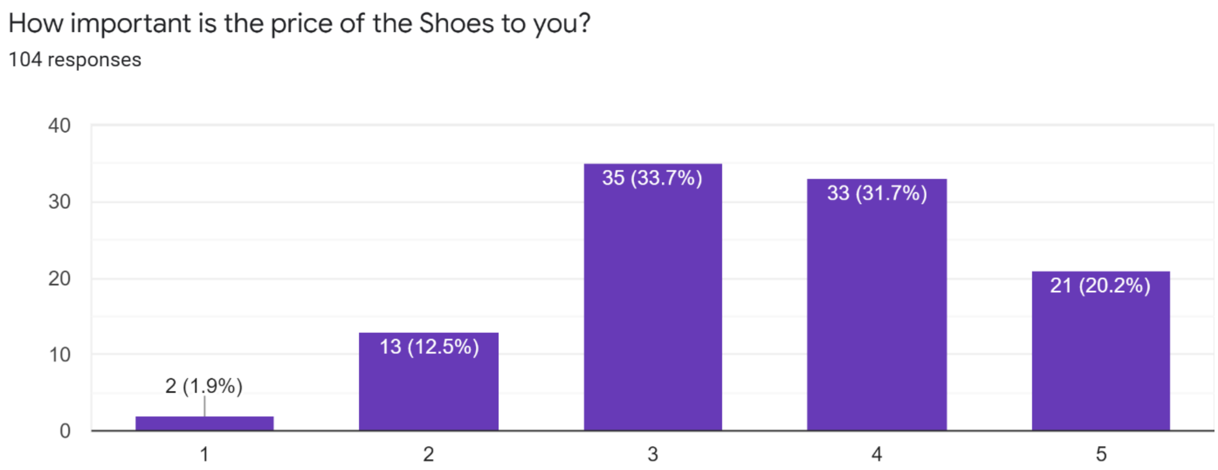


Figure 52: Customer survey results for price

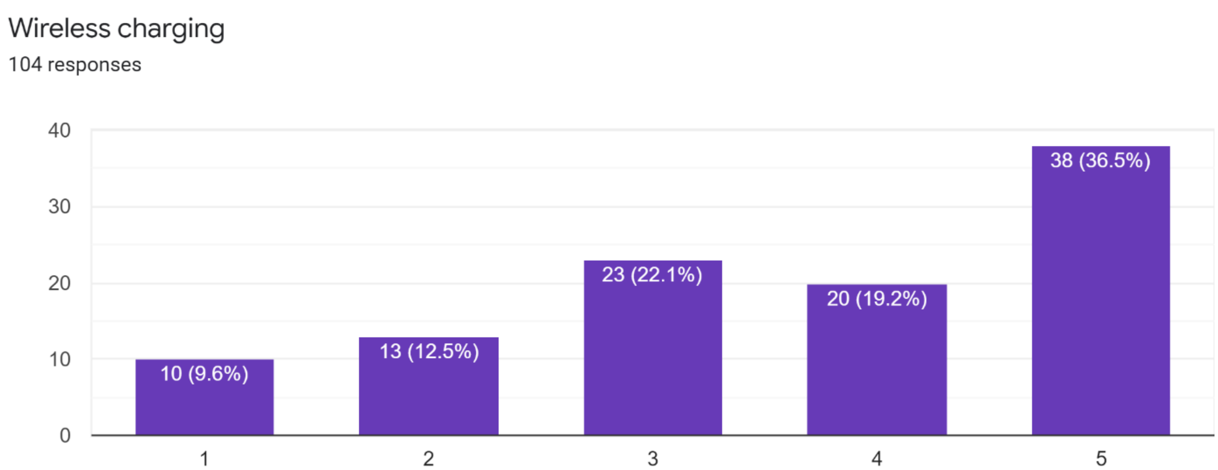


Figure 53: Customer survey results for wireless charging

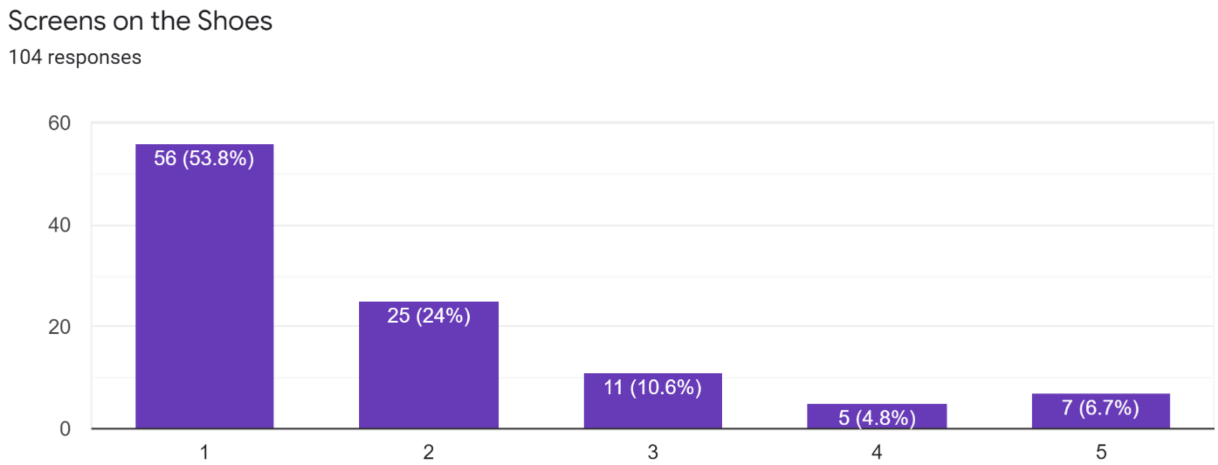


Figure 54: Customer survey results for screens on shoes

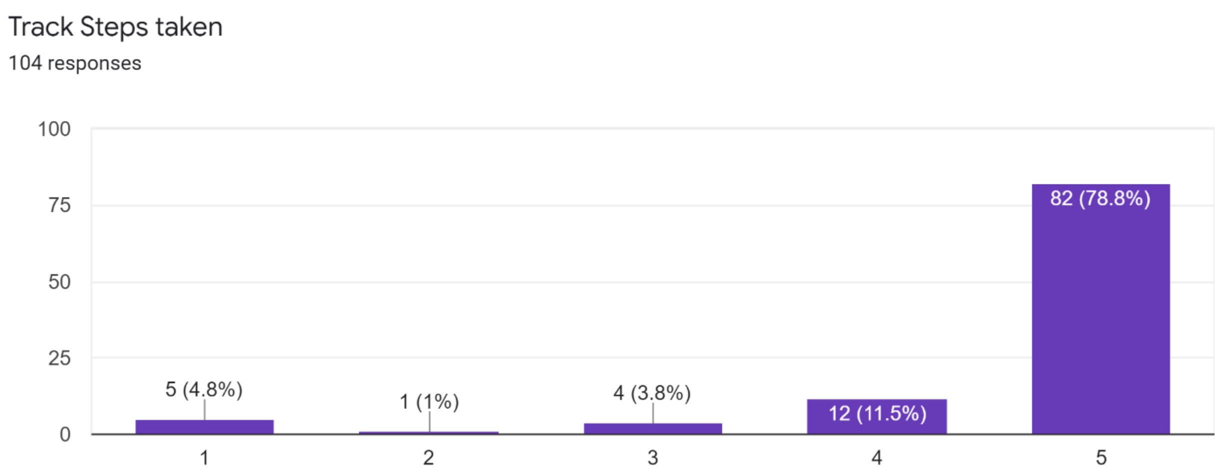


Figure 55: Customer survey results for tracking steps

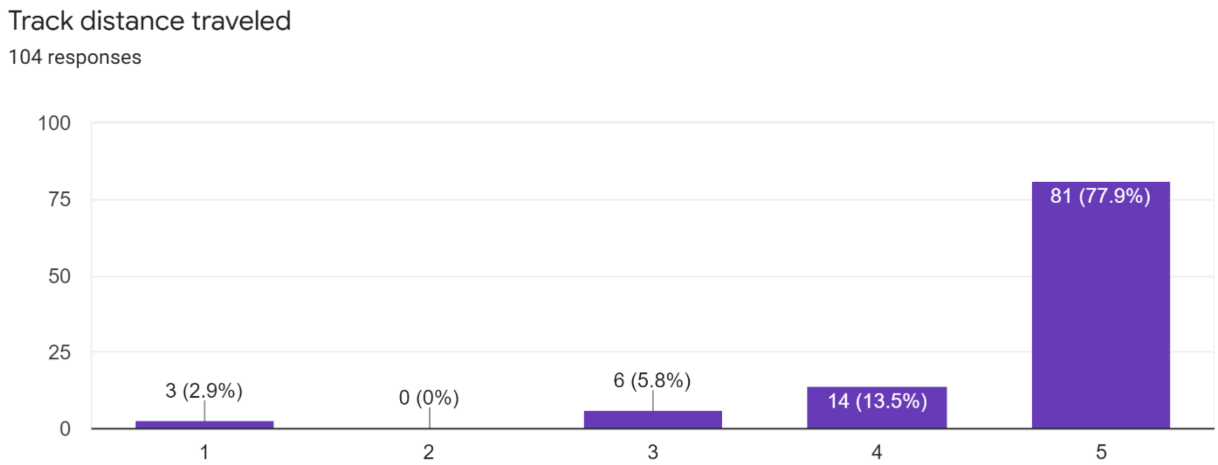


Figure 56: Customer survey results for tracking distance

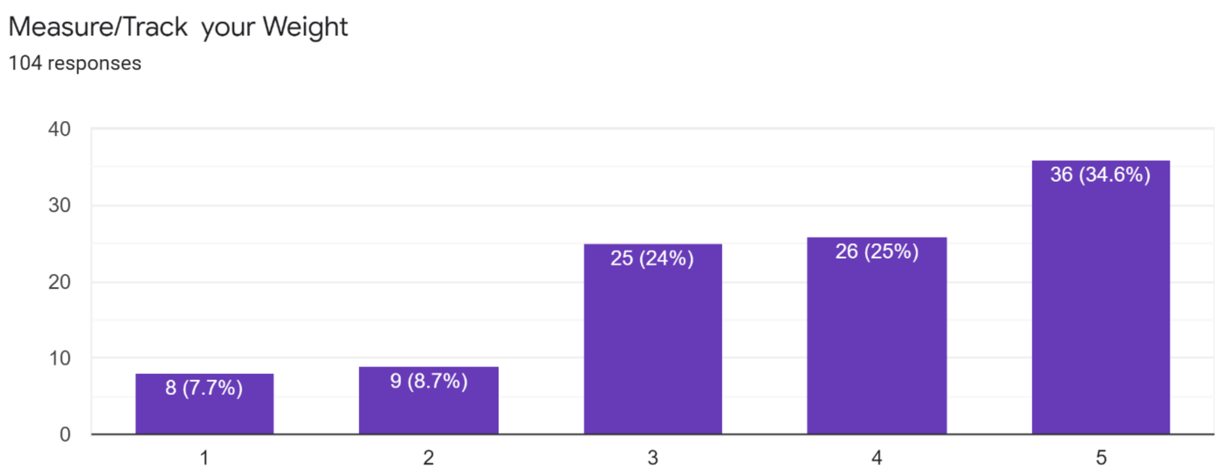


Figure 57: Customer survey results for measuring weight

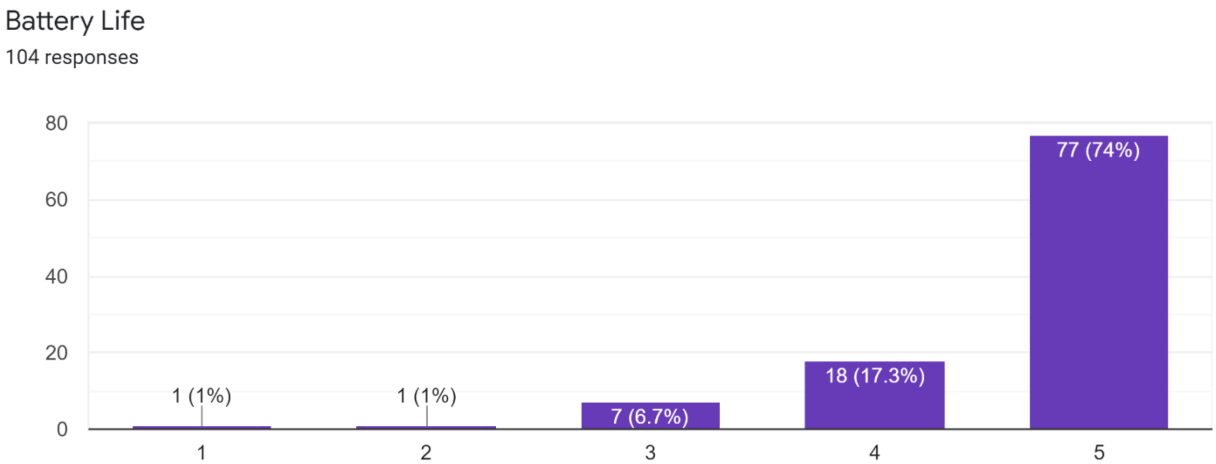


Figure 58: Customer survey results for long battery life

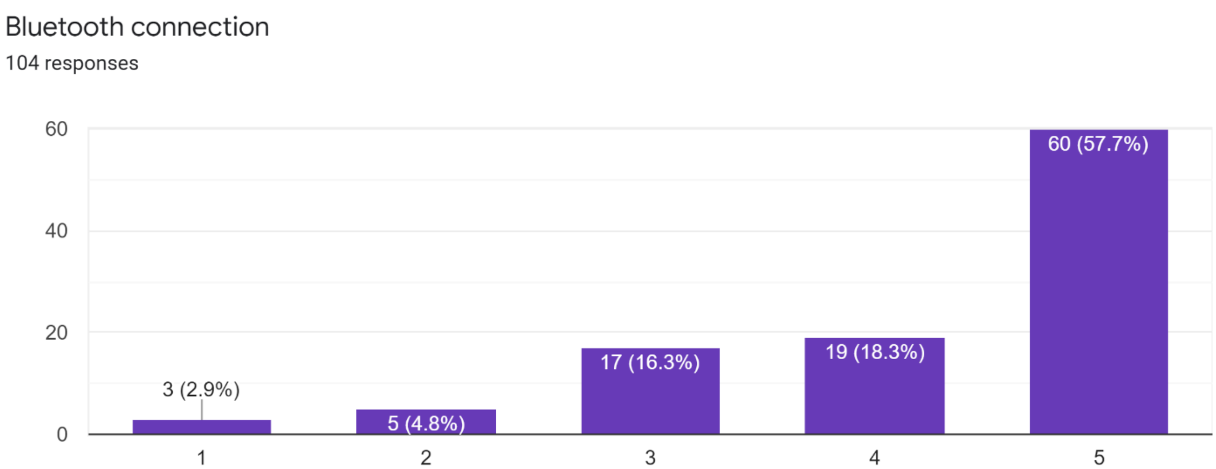


Figure 59: Customer survey results for Bluetooth connectivity

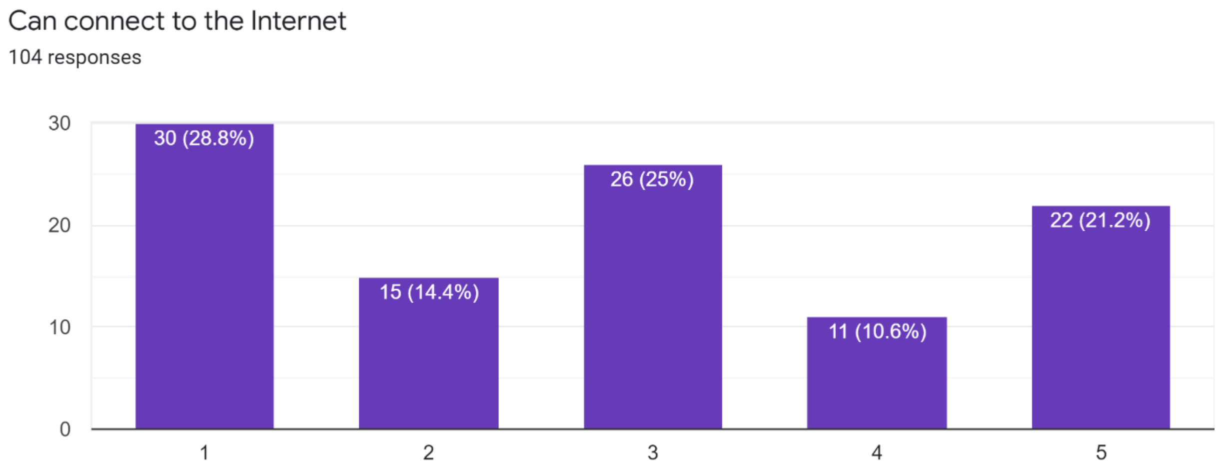


Figure 60: Customer survey results for connecting to the internet

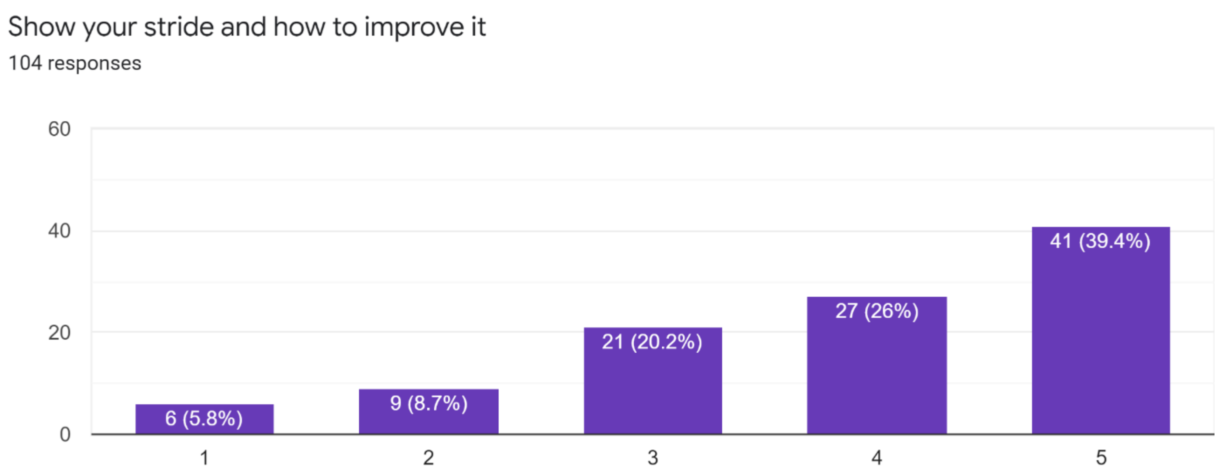


Figure 61: Customer survey results for improving stride

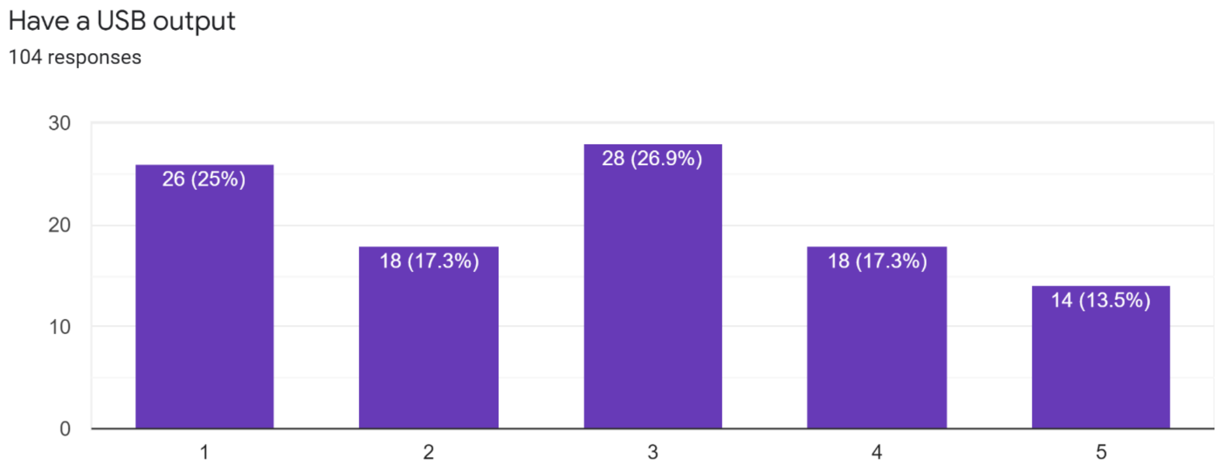


Figure 62: Customer survey results for a USB output

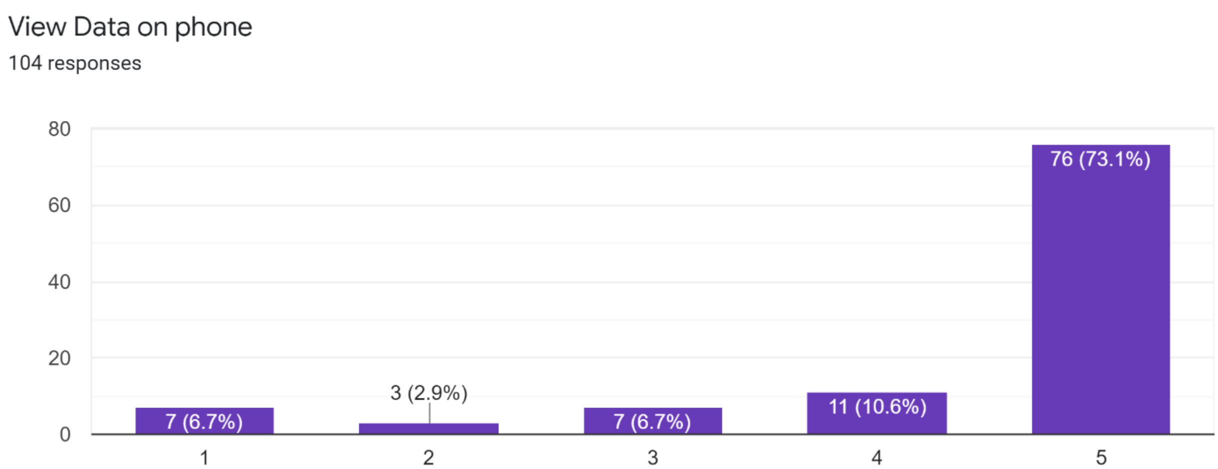


Figure 63: Customer survey results for viewing data on a mobile device

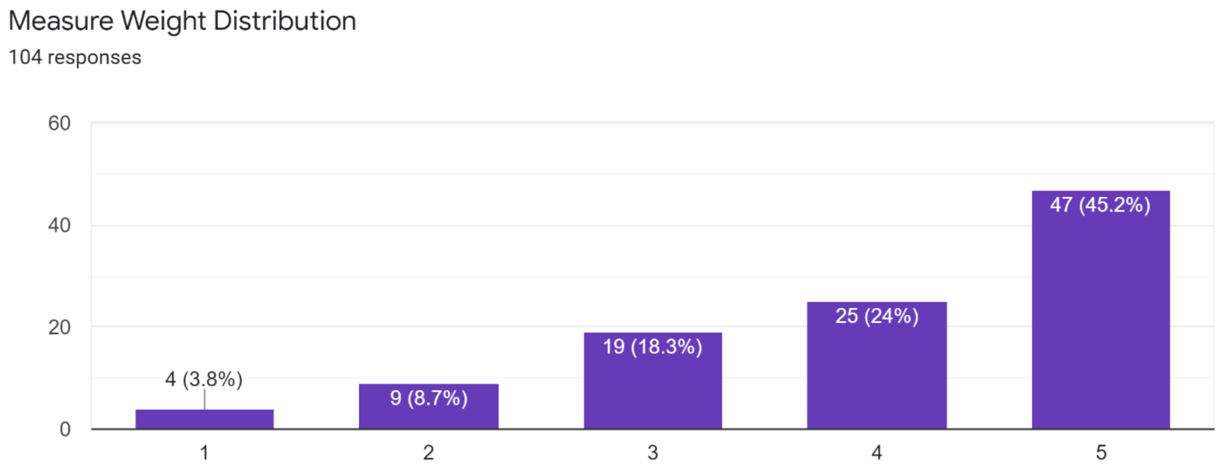


Figure 64: Customer survey results for measuring weight distribution

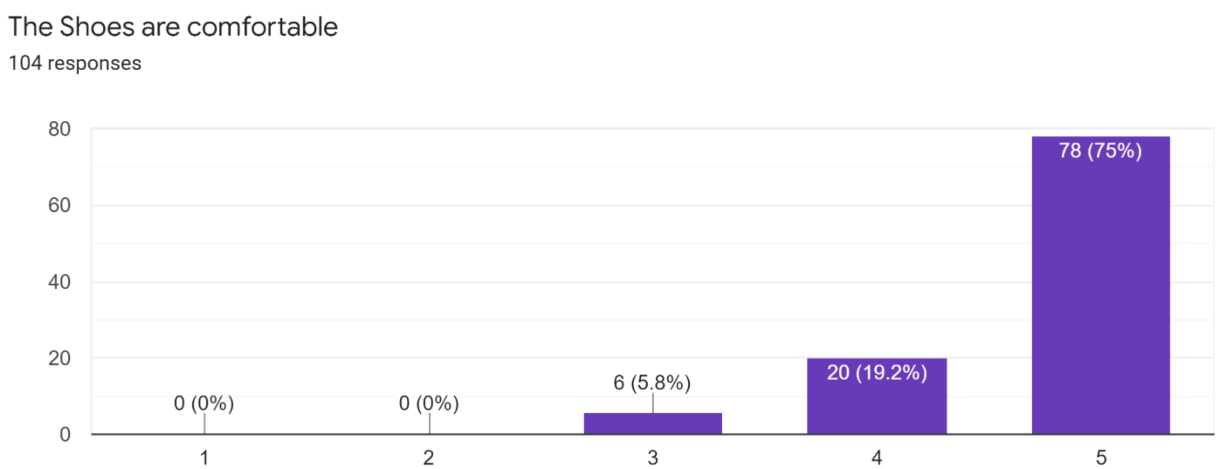


Figure 65: Customer survey results for comfortability

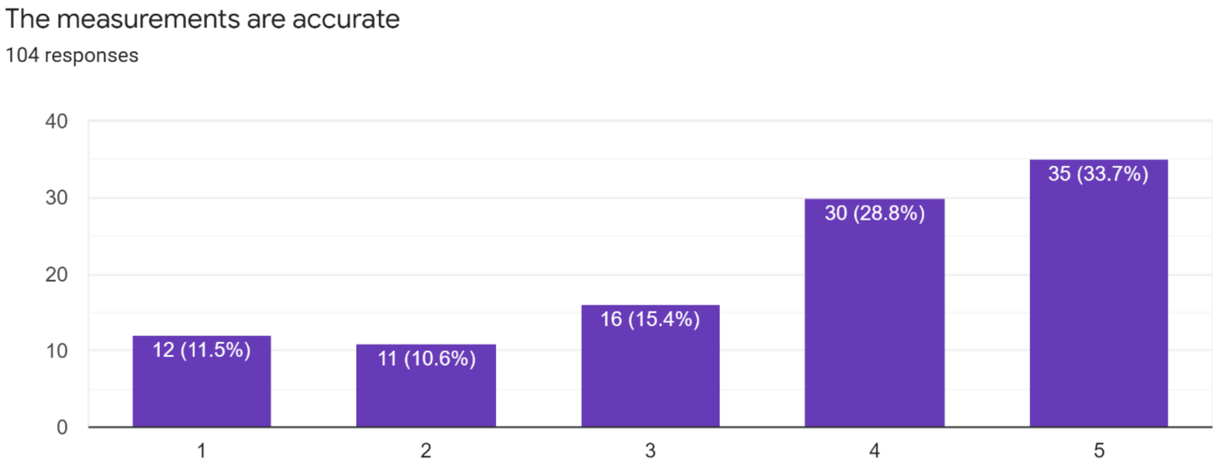


Figure 66: Customer survey results for accuracy

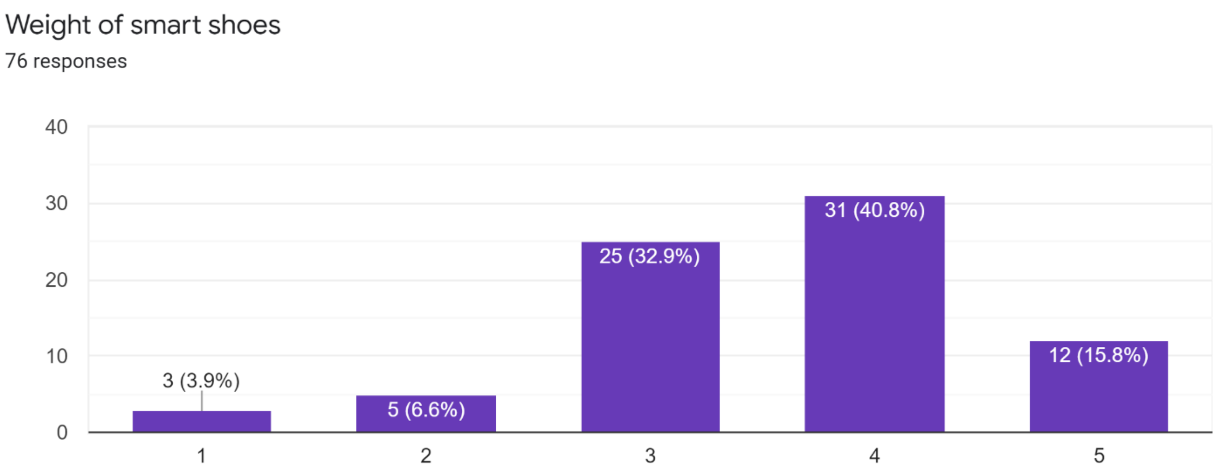


Figure 67: Customer survey results for weight of Smart Shoes