EK 210 - Final Report

Section: A6

Group Number: H8

Project Name: Biometric Device: Smart Runner 4000

Group Members: Reka Sundem, Nishtha Ladi, Michael Lwe, Ryan Wu

Executive Summary

Long-distance runners, the majority of whom are heel-strikers (landing on their heels when running), are at a higher risk of knee injuries that can hinder their performance. The Smart Runner 4000 addresses this issue by detecting excessive heel force, defined as force exceeding 30% of the total applied force, which increases the likelihood of injury (Daoud, 2012). Using precise measurements from Velostat sensors, the device alerts users via a vibration motor when excessive heel force is detected, helping them adjust their running form to prevent knee injuries. The product consists of a housing component secured to the user's leg with a Velcro strap, connected to an insole placed in the dominant leg's shoe via wires. The insole contains embedded force sensors that measure input forces and provide actionable feedback. Designed to be both portable and comfortable, the Smart Runner 4000 enhances running performance while ensuring user convenience. A Bill of Materials is provided in Appendix Section A.12.

Introduction

The Smart Runner 4000 is a compact, reliable, and comfortable device designed to improve the running form of long-distance runners. Around 80-90% of runners actually run with this improper form (Daoud, 2012). It provides alerts for excessive heel impact, helping users transition from heel-striking to forefoot-striking to reduce the risk of knee injuries. In fact, up to 50% of running related injuries are caused because of heel striking (Daoud, 2012). Our goal is to ensure that every individual can achieve their goals and complete their runs. Keeping this in mind, our four key objectives and user-based metrics, outlined in Appendix Section A.2, guided the product's development.

Problem Statement

To design a biometric device for long-distance runners that provides alerts on excessive heel impact, helping prevent knee injuries and improve running form.

Innovation

For our product, the two main components are the Velostat sensor for force sensing and the coin vibration motor for user feedback. After extensive research and testing, we selected the Velostat sensor due to its accuracy, flexibility, and ability to measure a wide range of forces. While the measured weight in our tests showed a 3 kg discrepancy from the actual weight, the sensor's accuracy error was only around 1% as shown in Appendix Section A.6, making it suitable for our needs. Compared to other force-measuring sensors like FSR, Velostat provides greater precision and range, making it ideal for our product. For user feedback, the coin vibration motor meets our needs by providing clear notifications when excessive force is detected. We found that a voltage range of 3-4V generates a comfortable vibration intensity for runners, avoiding excessive distraction. The motor is compact, requiring no additional components to operate, and fits seamlessly into the device. Both components align well with our product's goals of providing accurate feedback without disrupting the runner's experience.

Overall Description of Final Product

Introducing the Smart Runner 4000: a compact device designed to improve running form and reduce the risk of knee injuries. The housing, measuring 12.5 x 6.5 x 3 cm, attaches securely to the leg with Velcro straps, while the Velostat-embedded insole fits into the shoe as seen in Appendix Section A.7. Currently, the insole is available in one size, but expanding size options is a key focus for future development. The device is activated by a switch on top, powering the Arduino Uno inside as shown in Appendix Section A.8. To use it, the user inserts the insole into their shoe and secures the device to their leg. The insole contains two Velostat sensors positioned at the front and back to measure the forces applied by different parts of the foot over a 15-second collection period, using code outlined in Appendix Section A.10 and Appendix Section A.11. We don't expect running to be consistent, so taking an average over a short period of time helps account for variations in stride. The device calculates the average forces and alerts the user with a vibration if over 30% of the total force is from the heel (Appendix Section A.9), indicating a running form that could cause knee injuries. It also stops vibrations when the user is standing. The device is designed for one foot, typically the dominant right foot and it assumes symmetrical force application between feet but can be adjusted for the left foot if needed.

Key Objectives and Evaluation of Results

The Smart Runner 4000 was designed with four key objectives in mind: portability, comfort, precision of measurements, and readability of results. To address the primary goal of readability, we incorporated a coin vibration motor to alert users to improper running form. This compact device provides vibrations strong enough to grab attention without distracting runners. Our metric required at least 90% of users to sense the vibration, and in tests with 10 participants running and standing, all users (100%) reported sensing it, surpassing our expectations, as seen in Appendix Section A.6. For precision, we opted for Velostat sensors due to their wide measurement range and superior accuracy compared to standard FSR sensors. To accommodate natural variability in running form, we set a target error range of 10% for the moving average of force measurements. Testing revealed an error range of 2.02% to 8.85%, far exceeding our precision goals and ensuring users can trust the device's feedback, as shown in Appendix Section A.6. Portability was another crucial consideration, as long-distance runners require lightweight and non-bulky devices. We set a weight limit of 1 pound and achieved a compact design weighing just 5.6 ounces, making the device unobtrusive during use. Finally, we prioritized comfort and user-friendliness. Since the device is worn on the ankle, it needs to avoid interfering with running form. We tested this by having 10 users wear the product while standing and running; 8 out of 10 found it comfortable, meeting our 70% target. Additional user-friendly features include an easy on/off switch, velcro straps for quick attachment and removal, and a simple insole design that fits seamlessly into shoes. These key features of Smart Runner 4000 collectively make our product an effective, precise, lightweight, and comfortable tool for improving running form.

Lessons Learned

Developing the Smart Runner 4000 was a rewarding journey that taught our team invaluable lessons in teamwork, technical implementation, and working with constraints, outlined in Appendix Section A.3. Managing the project timeline while balancing extensive testing and troubleshooting proved crucial, sharpening our organizational skills along with learning about the Pairwise Comparison Chart, Glass Box, and Morph Chart to decide exactly what we believed the client wanted (Appendix Sections A.1, A.4, A.5). On the technical side, we learned to refine our design choices through trial and error. For instance, we realized that using a voltage input instead of a current input from the Velostat sensor would enhance the accuracy of force readings. Additionally, we opted for Velostat sensors due to their wider force range compared to FSR sensors, chaining them as our primary sensing devices. We also found the importance of integrating a switch—a component initially overlooked in our circuit and housing diagrams. Recognizing this, we incorporated a switch into the design for easy user control, positioning it on top of the device for accessibility. Similarly, we replaced the Arduino Nano with the Arduino Uno for the ankle housing because the device overheated due to excess power, which works well with the Uno as seen in Appendix A.13. Although we initially considered ribbon wires for their compactness, we ultimately retained jumper wires for their durability and strength, ensuring reliability during use. Our housing design process also provided valuable insights. We tested various configurations for the switch, wires, lid with screws, and ankle strap extension, finalizing a design that minimizes wire length to reduce entanglement. The final positioning for the switch was on top of the device for easy access by the user. Throughout this process, we encountered numerous challenges and "engineering breakpoints," which prompted creative problem-solving and iterative design improvements. Safety and durability were paramount, leading us to implement key protective features. We applied silicone spray to waterproof the device, safeguarding it against sweat, rain, or snow. Exposed wires were covered with electrical tape for additional protection, and wires leading to the insole were designed to detach in case of accidental tripping, preventing damage. To avoid short circuits, we incorporated 10k Ω resistors. We also programmed the device to stop vibrating when the user is standing still, conserving energy and enhancing functionality. Additional features, such as adjustable Velcro straps and a replaceable battery, make the Smart Runner 4000 versatile and user-friendly. The adjustable straps ensure compatibility for users of all sizes, while the removable battery provides convenience. These thoughtful details reflect our commitment to creating a reliable, practical product. From optimizing component placement to enhancing durability and usability, every step taught us invaluable lessons about engineering, collaboration, and adaptability. These learnings not only helped refine the Smart Runner 4000 into a reliable and user-friendly product but also allowed us to grow as individuals, equipping us with skills and insights that will benefit us in future endeavors. For the Smart Runner 4000 2.0, future improvements could include replacing the wired connections to the insole with Bluetooth technology and further compacting the housing component to enhance the runner's mobility. Additionally, the housing could be redesigned as a wristband, providing screen-based alerts for the user.

Bibliography

1. “Uno R3.” *Docs.Arduino.Cc*, docs.arduino.cc/hardware/uno-rev3/. Accessed 6 Dec. 2024.

This reference helped us figure out a very important aspect of our Arduino code - converting the voltage input into the arduino.

2. *Nfpshop*, nfpshop.com/wp-content/uploads/2023/04/SPEC-NFP-WS0625-20230329_Watermark.pdf.

This link was helpful with the specs for the vibration motor

3. “PKCELL 6f22 9V Carbon Battery Extra Heavy Duty Battery.” *Https://Www.Fuspower.Com/*, www.fuspower.com/pkcell-extra-heavy-duty-battery-6f22-9v-product/.

This link provided the battery capacity of the 9V battery needed to calculate the battery life.

4. *Foot Strike and Injury Rates in Endurance Runners*, scholar.harvard.edu/files/dlieberman/files/2012b.pdf.

This link provides us with the statistics, from a study by Adam I. Daoud, Daniel E. Lieberman, Gary J. Geissler, Frank Wang, Jason Saretsky, and Yahya A. Daoud, of heel-striking and forefoot-striking, on the basis of which our product was developed.

Appendix

A.1 Pairwise Comparison Chart (PCC)

Table A.1 Pairwise Comparison Chart

Objectives	Portable	Precision	Readability of Results	Comfortable	Cleanable	Easy to Use	Accuracy	Total score
Portable	-	1	1	1	1	1	1	6
Precision	0	-	1	0	1	1	1	4
Readability of Results	0	0	-	0	1	1	1	3
Comfortable	0	1	1	-	1	1	1	5
Cleanable	0	0	0	0	-	0	0	0
Easy to Use	0	0	0	0	1	-	1	2
Accuracy	0	0	0	0	1	0	-	1

From the total score calculated from the PCC (Table A.1), we were able to organize a final list of objectives from most to least importance:

1. Portable
2. Comfortable
3. Precision
4. Readability of Results
5. Easy to Use
6. Accuracy
7. Cleanable

A.2 Objective/Metrics

Using the top 4 objectives (underlined above) from the PCC (Table A.1), we were able to identify metrics for each of the objectives that our product should satisfy (Table A.2).

Table A.2 Objective/Metrics

1. PORTABLE	Weight of product **<= 1 pound**
2. USER-FRIENDLY AND COMFORTABLE	**>=80%** of users find the product comfortable
3. PRECISE	Moving average of measured forces is in the range of **<10%**
4. READABLE RESULTS	**>=90%** of users sense a vibration

A.3 Constraints

1. Product attached to user's ankle - The current product is attached to user's ankle and has a wired connection. If future versions incorporate Bluetooth or wireless connections, this constraint can be avoided.
2. Insole Size Compatibility - The current version of the product is designed for a specific shoe size, limiting its use for runners who do not fit that size. Expanding size options would increase the complexity of the design and manufacturing process.
3. Battery Life - Although not rechargeable, ensuring long battery life for long-distance runners is essential, and this may limit the power capabilities or compactness of the device.

A.4 Glass Box Analysis

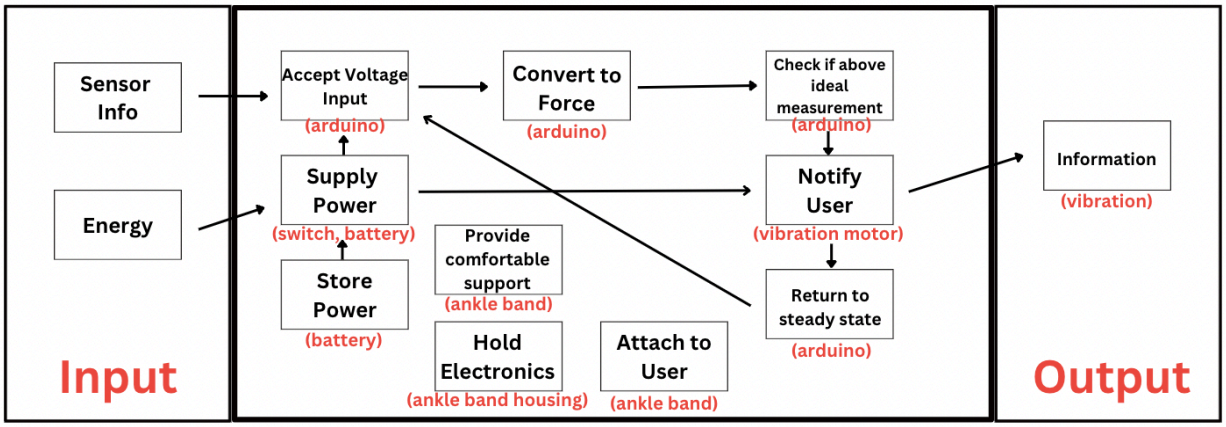


Figure A.4 Glass Box Analysis

The glass box (Figure A.4) contains all the functions in boxes for our product to support all our objectives. Most of the functions have been connected using arrows to show the flow of execution of functions in our product. The means that our group has chosen are denoted below each function box in red. We have also added the input and output parameters required to execute the functions of our product, according to the 3 categories - energy, material and information. For the ankle band, a stretchy and adjustable material is ideal. Thus, we chose velcro as our material to use for the strap. Our glass box analysis made us understand how our product functions, essentially forming a loop that accepts the voltage input every 15 seconds, makes a decision, and repeats this process until the product is turned off. It also gave us a clear sense of the logistics of how the user would use the product.

A.5 Morph Chart

Table A.5 Morph Chart

Functionality	1	2	3	4	5
Accept Current/Voltage/ Force Input	FSR Sensors	Force Plate	Motion Capture Technology	Pressure Mapping Insoles	Velostat Sensors
Notifies User	Vibration Motor	LCD Screen	Sound effects	Blinking LED	—
Attach to User	Velcro Strap	Elastic strap	Held in hand	Waist Belt	Harness
Provide Comfortable Support	Wrist housing	Ankle housing	Lightweight and portable	___	___

Using our main functionalities of the product from the Glass Box Analysis (A.4), we were able to ideate various solutions to fulfil each functionality to create our Morph Chart. The highlighted (in red) solutions in the table (Table A.3) are the solutions that we chose to implement in our product to satisfy the functional requirements.

A.6 Testing Main Components - Vibration motor, Velostat Sensor

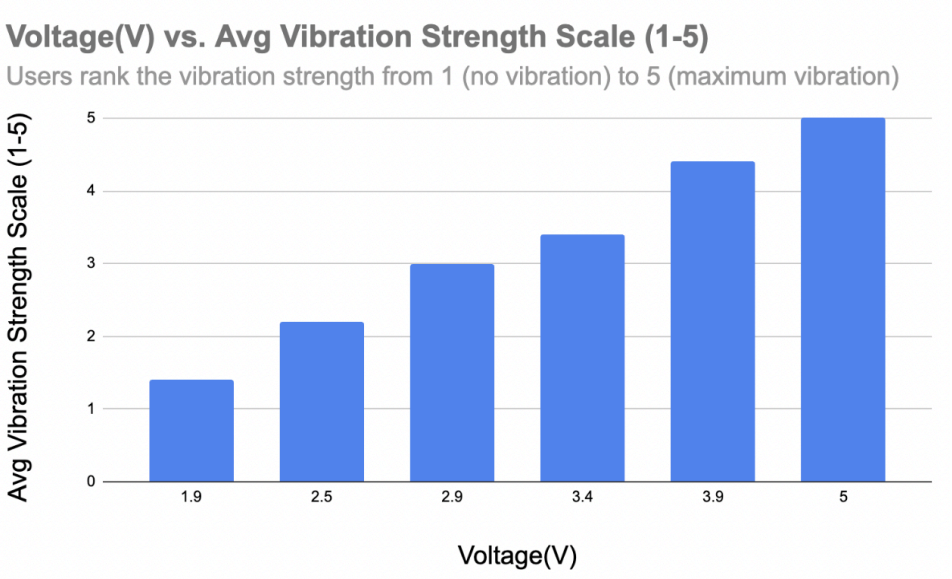


Figure A.6a Vibration Motor - Vibration Strength Data

The strength of the vibration in the range of voltages provided by the vibration motor is ideal for our users (runners). It gives a mechanical output of a vibration when the force exerted by their feet on the ground exceeds a certain limit. After testing a range of strengths for vibrations through voltage measurements (Figure A.6a), we have a fair idea on which vibrations can be felt on the hand and which can not. Moreover, we also got a perspective into users' opinions on which strength/voltage can be used for our product which is comfortable and efficient for a runner. We set the vibration strength as 200 or 4V.

Table A.6a Velostat Current Data Table

People/Object	Red Water Bottle (Control)	TA1	TA2	Ryan Wu	Michael Lwe	Student1
Measured Weights (kg)	0.818	65.78	65.77	77	88	108.86
Measured Current 1 (Amps)	0.00089	0.07	0.074	0.084	0.099	0.115
Measured Current 2 (Amps)	0.0009	0.076	0.076	0.086	0.094	0.114
Measured Current 3 (Amps)	0.00088	0.075	0.078	0.088	0.097	0.118
Measured Current 4 (Amps)	0.00087	0.077	0.076	0.082	0.096	0.121
Measured Current 5 (Amps)	0.00088	0.079	0.079	0.081	0.095	0.117
Measured Current 6 (Amps)	0.00089	0.08	0.08	0.084	0.094	0.118
Measured Current 7 (Amps)	0.00089	0.077	0.08	0.085	0.095	0.12
Measured Current 8 (Amps)	0.0009	0.078	0.079	0.082	0.096	0.123
Measured Current 9 (Amps)	0.00089	0.079	0.082	0.084	0.103	0.123
Measured Current 10 (Amps)	0.00089	0.077	0.077	0.085	0.093	0.121
Average Current (Amps)	0.000888	0.0768	0.0781	0.0841	0.0962	0.119

*This data shows the 10 current measurements we took for each object/person along with the average of the currents, the conversion factor, and the calculated weight.

Table A.6b Velostat Current Precision Table

Precisions	Control Precision	TA1 Precision	TA2 Precision	Ryan Precision	Michael Precision	Student1 Precision
(Average-Min)	0.000018	0.0068	0.0041	0.0031	0.0032	0.005
(Max-Average)	0.000012	0.0032	0.0039	0.0039	0.0068	0.005
Largest Precision Percent	2.027027027	8.854166667	5.249679898	4.637336504	7.068607069	4.201680672

Table A.6c Velostat Accuracy Table

Calculated Weight	N/A	70.74594595	71.94346847	77.4704955	88.61666667	109.6193694
Individual Error Percent (%)	N/A	7.549324941	9.386450461	0.611033111	0.7007575758	0.6975651014

*This table shows the error from true value for each person.

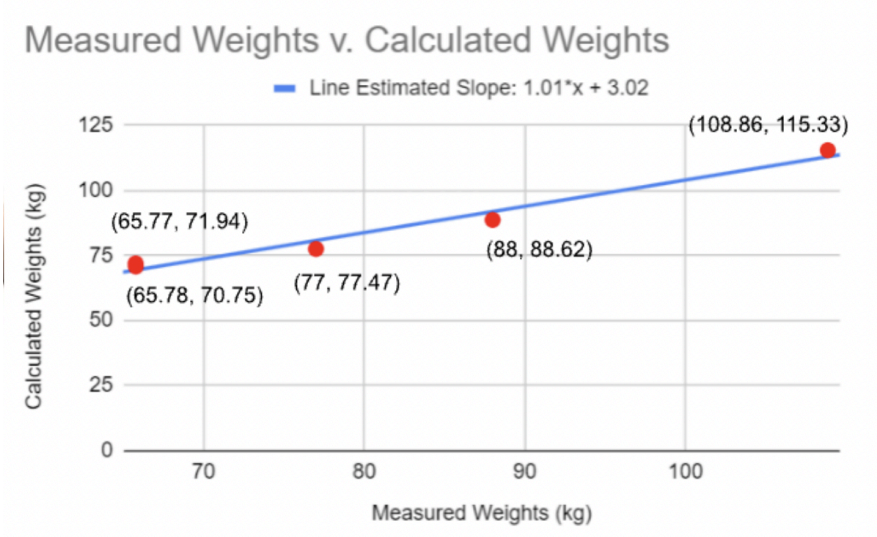
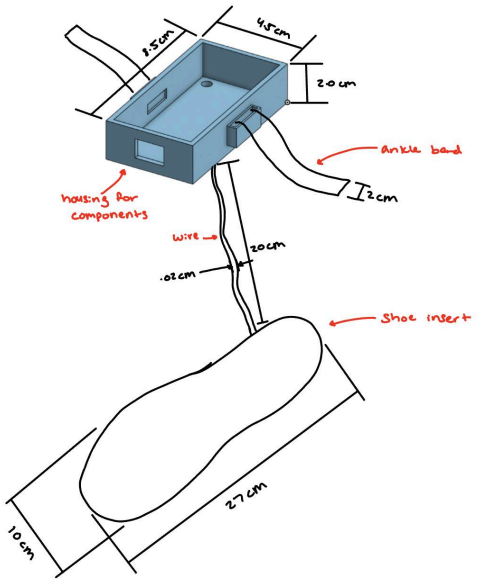


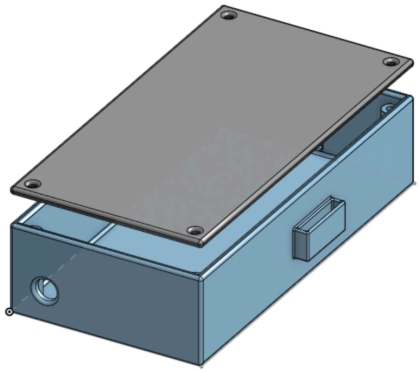
Figure A.6b Velostat Accuracy Graph

Based on the graph (Figure A.6b), while the accuracy error is only around 1%, the measured weight, the line of best fit shows that the calculated weight is off by around 3 kg from the actual measured weight of the person. Based on this analysis, we decided to use the Velostat sensor as our sensing device. Compared to other force-measuring sensors like FSR, Velostat provides greater precision and range, making it ideal for our product.

A.7 CAD Drawings and Final Product Showcase



Visual A.7a Preliminary CAD Representation of Our Device with Dimensions



Visual A.7b Final CAD Representation of Our Device with a Lid

Figure A.7 Final Product (Worn)

Figure A.7 Final Product (Not Worn)

A.8 Circuit Diagram

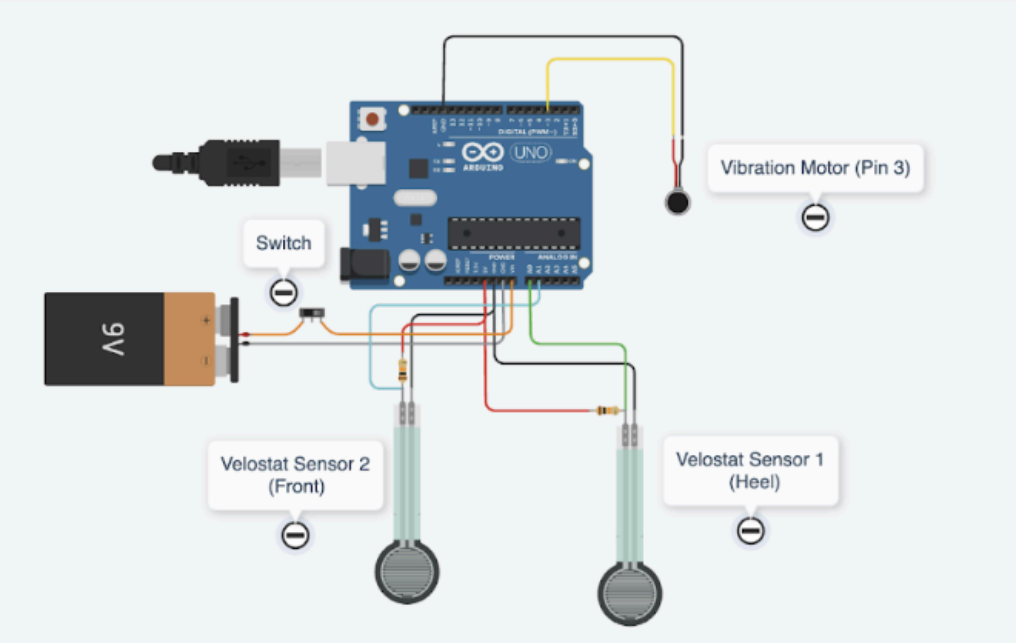


Figure A.8 Circuit Diagram

The above circuit diagram (Figure A.8) includes all the parts that we will be using for our final product. The parts include:

1. 9V Battery
2. Two Velostat Sensors
3. Arduino Uno
4. Vibration Motor
5. Switch
6. Connecting Wires
7. Two 10,000 Ω Resistors

The two velostat sensors, which accept voltage input, are connected to pin A0 and A1, along with ground. The 9V battery, that stores and supplies power for the product, is connected to ground and V(in). The vibration motor is connected to Digital Pin 3. The switch controls the circuit's funneling of energy into the motor. It then completes the motor circuit into the ground of the arduino. The arduino board converts the analog values to digital values and performs all the logical calculations and tasks (converting voltage input to force input and check conditions). All the components are connected with connecting wires and soldered to keep them intact. The two sensors that are used to measure the force impact are connected in a parallel connection. Lastly, the resistors are added to the circuit to prevent any possible short circuit from occurring.

A.9 Logic

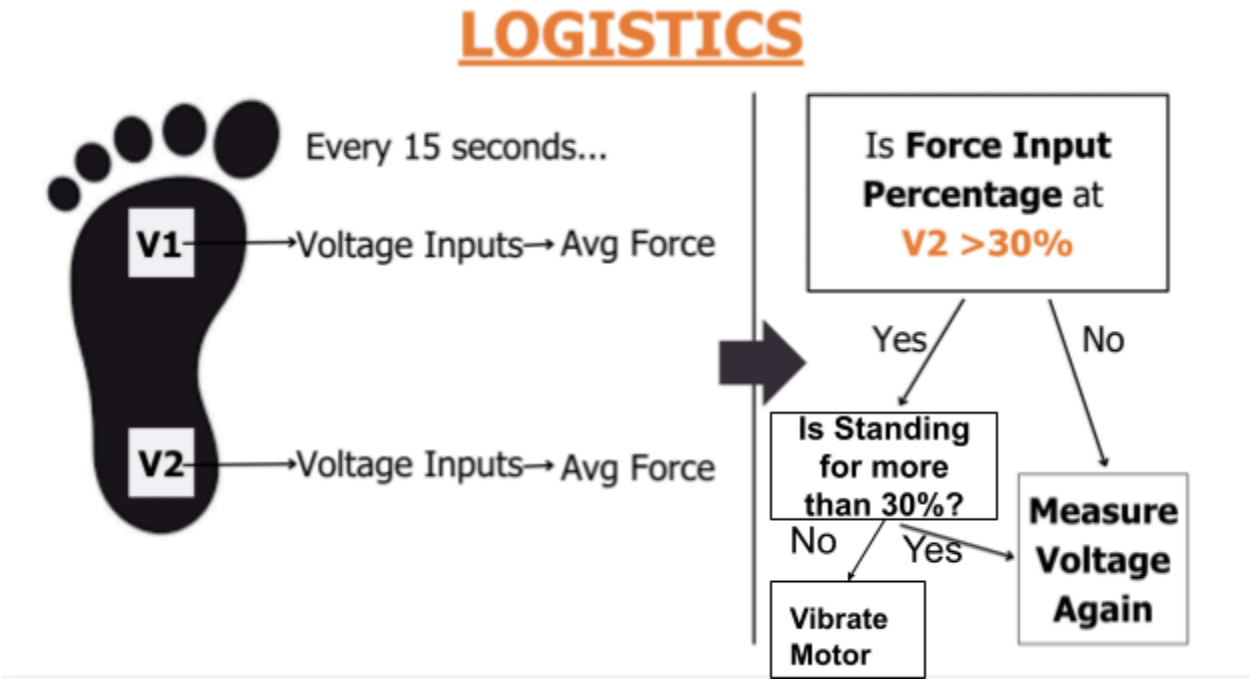


Figure A.9 Logistics Diagram

Figure A.9 explains how the product works for a duration of every 15 seconds. We incorporated an additional condition to check if the user is standing too. The logic for checking the force input percentage value is based on our research of heel-strikers and forefoot-strikers:

Key Ratios:

- Heel Strikers:
  - Heel-to-Forefoot Force Ratio: ~3:1 (heel receives ~75% of the force, forefoot ~25%).
- Forefoot Strikers:
  - Forefoot-to-Heel Force Ratio: ~2:1 (forefoot receives ~67% of the force, heel ~33%).

To make runners switch from heel-striking to forefoot-striking, any average force input percentage at the heel above 30% would cause a vibration in the product.

If the measured force value of the heel is in the range of 10% of the average force value, the measurement is considered to be a standing measurement. By counting the number of measurements that are considered standing, we can find out if the user is standing for 30% or. 4.5 seconds of the 15 second collection period. This value was considered as a default setting for our product, but can easily be changed.

A.10 Code Flowchart

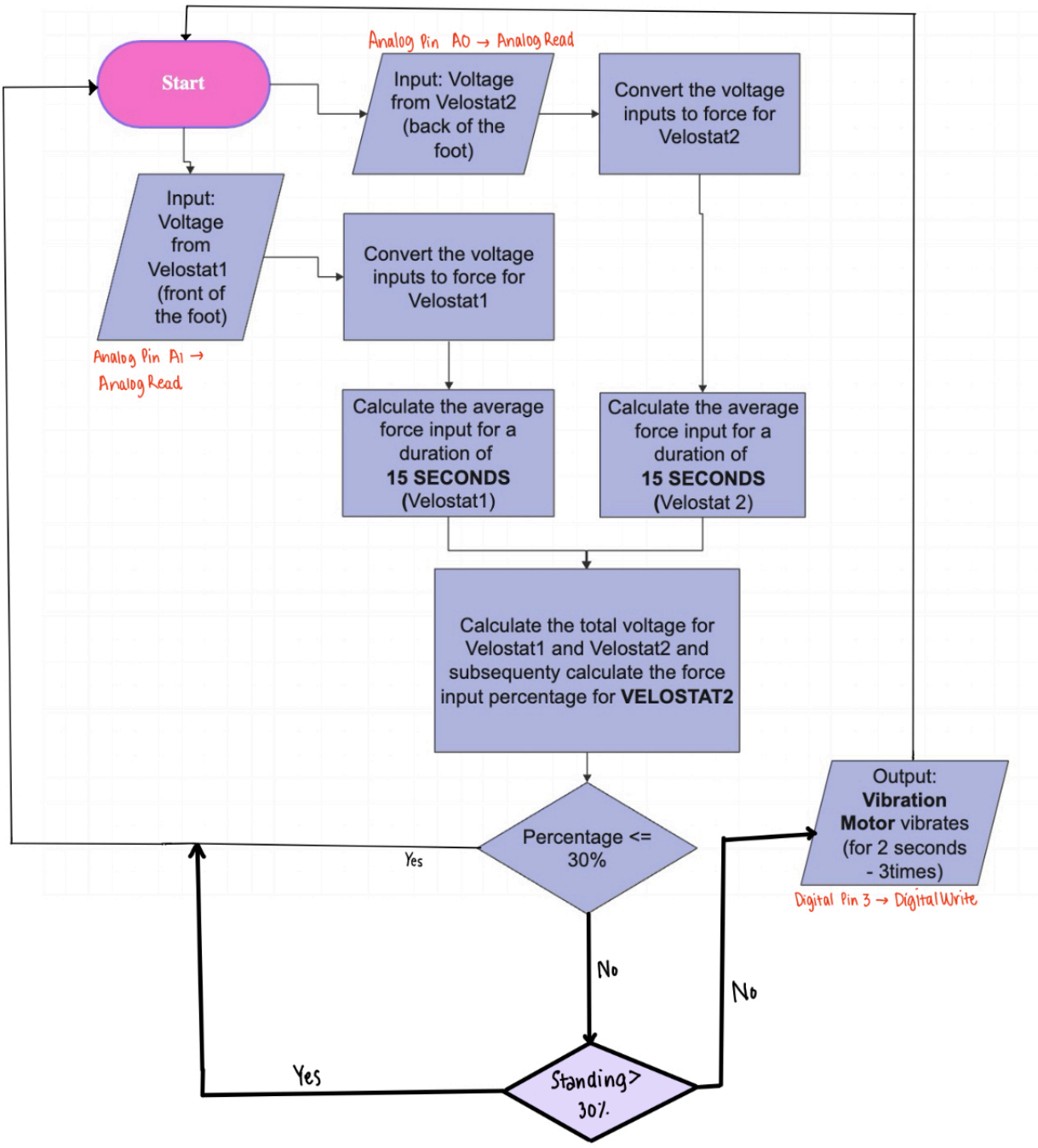


Figure A.10 Code Flow Chart

A.11 Final Arduino Code

```
const byte velostatBack = A0;    // Velostat (back foot) is connected
to analog pin A0
const byte velostatFront = A1;    // Velostat (front foot) is
connected to analog pin A1
const int motorPin = 3;  // Vibration motor is connected to digital
pin 3
const float inputVoltage = 5.0;  // Supply voltage
const float R_ref = 10000.0;  // Reference resistor (10kΩ)
const float calibrateForce = 10656.5; // Calibration of Resistance to
accurate Force measurement
bool triggerVibrationMotor = false;
double countStandingCheck = 0;
const int vibrationStrength = 200; // Measured strength of the
vibration

unsigned long previousMillis = 0;  // Stores the last time the
averaging was done
const long interval = 15000;  // 15 seconds delay

float accumulatedA0 = 0;  // Accumulator for Velostat 1 (front foot)
float accumulatedA1 = 0;  // Accumulator for Velostat 2 (back foot)
double sampleCount = 0;      // Number of samples taken during the
interval

void setup()
{
  Serial.begin(9600);
  pinMode(motorPin, OUTPUT);  // Set the motor pin as an output
}

void loop()
{
  //Function that returns the time the Arduino has been running the
code
  unsigned long currentMillis = millis();

  // Read data from velostatBack (front foot, connected to A0)
  int analogBack = analogRead(velostatBack);
  float vBack = (analogBack * inputVoltage) / 1023.0;  // Convert
analog reading to voltage

  // Read data from FSR2 (back foot, connected to A1)
  int analogFront = analogRead(velostatFront);
```

```
  float vFront = (analogFront * inputVoltage) / 1023.0;  // Convert
analog reading to voltage

  // Calculate the resistance of FSR1 (front foot) and FSR2 (back
foot)
  float resistanceBack = (vBack * R_ref) / (inputVoltage - vBack);
// Resistance of FSR1
  float resistanceFront = (vFront * R_ref) / (inputVoltage - vFront);
// Resistance of FSR2

  // Approximate the force based on the resistance (force is
inversely proportional to resistance)
  // Calculated using voltage divider and calibrated using the data
from our modeling assignment)
  float forceBack = calibrateForce / resistanceBack;  // Force from
velostatBack (back foot)
  float forceFront = calibrateForce / resistanceFront;  // Force from
velostatFront (front foot)

  // Accumulate the forces for averaging
  if(forceBack/(accumulatedA0/sampleCount) <= 1.10 &&
forceBack/(accumulatedA0/sampleCount) >= 0.90){
    countStandingCheck++;
  }
  accumulatedA0 += forceBack;
  accumulatedA1 += forceFront;
  sampleCount++;

  // Vibrates the Vibration Motor in 3 seconds on/2 seconds off time
intervals
  if(triggerVibrationMotor){
    if((currentMillis - previousMillis)/1000 >= 0 && (currentMillis -
previousMillis)/1000 < 3){
      Serial.println("Buzz Once");
      analogWrite(motorPin, vibrationStrength);    // Turn on the
vibration motor
    }
    else if((currentMillis - previousMillis)/1000 >= 3 &&
(currentMillis - previousMillis)/1000 < 5){
      Serial.println("Buzz Halt1");
      analogWrite(motorPin, 0);    // Turn off the vibration motor
    }
    else if((currentMillis - previousMillis)/1000 >= 5 &&
(currentMillis - previousMillis)/1000 < 8){
      Serial.println("Buzz Twice");
```

```
      analogWrite(motorPin, vibrationStrength);    // Turn on the
vibration motor
    }
    else if((currentMillis - previousMillis)/1000 >= 8 &&
(currentMillis - previousMillis)/1000 < 10){
      Serial.println("Buzz Halt2");
      analogWrite(motorPin, 0);    // Turn off the vibration motor
    }
    else if((currentMillis - previousMillis)/1000 >= 10 &&
(currentMillis - previousMillis)/1000 < 12){
      Serial.println("Buzz Thrice");
      analogWrite(motorPin, vibrationStrength);    // Turn on the
vibration motor
    }
    else if((currentMillis - previousMillis)/1000 >= 12 &&
(currentMillis - previousMillis)/1000 < 15){
      Serial.println("Buzz Halt3");
      analogWrite(motorPin, 0);    // Turn off the vibration motor
    }
  }

  // Check if 15 seconds have passed
  if (currentMillis - previousMillis >= interval) {
    // Reset the timer
    previousMillis = currentMillis;

    // Calculate the average force for FSR1 (front foot) and FSR2
(back foot)
    float avgForceBack = accumulatedA0 / sampleCount;
    float avgForceFront = accumulatedA1 / sampleCount;
    float totalForce = avgForceBack + avgForceFront;

    // Reset accumulation variables for the next 15-second period
    accumulatedA0 = 0;
    accumulatedA1 = 0;

    // Check if the average force on the back foot exceeds 30% of the
total force
    if (((avgForceBack / totalForce) > 0.30) &&
(countStandingCheck/sampleCount >= 0.3)) {
      Serial.println("Back foot is bearing more than 30% of the total
force. Buzzing motor...");
      triggerVibrationMotor = true;
    }
    else {
```

```
      triggerVibrationMotor = false;
    }
    countStandingCheck = 0;
    sampleCount = 0;
  }

  // Delay for stability in reading (can be reduced for smoother
readings)
  delay(100);
}
```

A.12 Bill of Materials (BOM)

Table A.12 Cost Sheet (BOM)

Item	Quantity	Price (dollars)
Velostat	1	20
Vibration Motor	1	3
Arduino Nano	1	2.55
Bluetooth sensors	1	10
9V Battery	1	4
Battery Connector	1	0
Wago Connector	1	0
Wires	1	0
Switch	1	2.75
Velcro Watch Band	1	12
Elastic Band	1	15
Ribbon Wires	1	10
Insole	1	11
	Total Amount:	90.3

Our cost sheet, as shown in Table A.12, provides all the components required for our product, as well as the individual prices of each component. Our total expenditure for this product was $90.3, which was well under the $200 budget limit provided for this project. This proves that our product is not too expensive to manufacture, keeping it feasible for customers and potential for more sales.

A.13 Power Budget Calculations

Table A.13 Power Budget Data and Calculations

Components	Max Voltage (V)	Max Current (A)	Max Power (W)	Duty Cycle (%)	Accumulated Power (W)
Battery	9	N/A	N/A	100	N/A
Arduino Uno	5	0.2	1	100	1
Velostat Sensor (Front)	5	0.2	1	100	1
Velostat Sensor (Back)	5	0.2	1	100	1
Vibration Motor	4	0.2	0.8	60	0.48

$$Battery\ Life\ = \frac{Battery\ Capacity * Battery\ Efficiency}{Total\ System\ Current} = \frac{1200\ mAh * 0.90\%}{80\ mA} = 13.5\ hours$$

Equation A.13 Theoretical Battery Life Calculations

From our calculations, we calculated the battery life to be 13.5 hours, using all of our components. Assuming the runtime of a whole marathon to be an average of 4.5 hours, we can conclude that our product can sufficiently run for 3 whole marathons, and 6 half marathons, which is a pretty good amount of time for the product to run in its first version. Furthermore, the battery is easily replaceable and convenient for the user, so changing the battery should not be a problem for the user.