

IMPACT INSOLES

INSOLE PRESSURE SYSTEM FOR RUNNING SHOES

By
Joseph Casino
Aarush Sivanesan
Matthew Weng

Design Document for ECE 445, Senior Design, Spring 2026

Professor: Craig Shultz

TA: Xiaodong Ye

24 February 2026

Project No. 68

Contents

Contents.....	2
1. Introduction.....	3
a. Problem and Solution.....	3
b. Visual Aid.....	4
c. High-Level Requirements List.....	4
2. Design.....	5
a. Physical Design.....	5
b. Block Diagram.....	6
c. Subsystem 1: Thin-Film Pressure Sensor Insole Array.....	6
d. Subsystem 2: Analog Front-End + ADC Data Acquisition.....	7
e. Subsystem 3: Microcontroller + BLE Wireless Telemetry.....	9
f. Subsystem 4: Power Management + Charging.....	10
g. Subsystem 5: Phone Interface / Data Visualization.....	11
h. Subsystem 6: Optional Motion Sensing (IMU).....	13
i. Tolerance Analysis.....	14
3. Cost and Schedule.....	14
a. Cost Analysis.....	14
b. Schedule.....	16
4. Ethics and Safety.....	17
5. Citations.....	18

1. Introduction

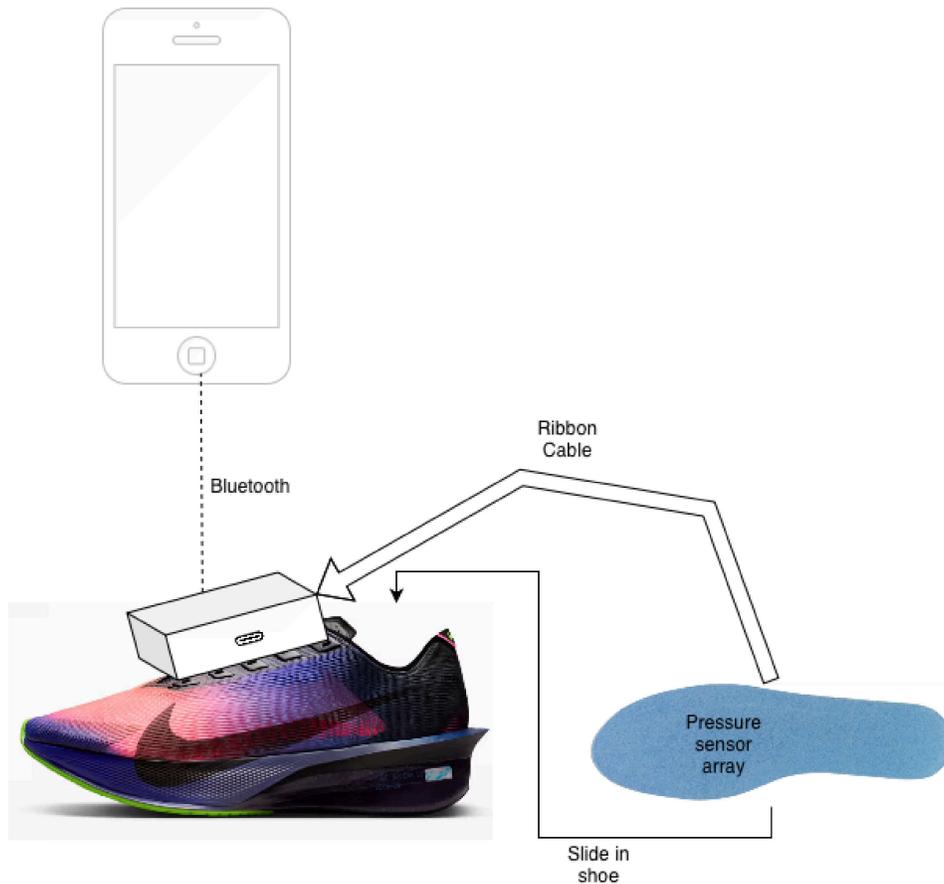
a. Problem and Solution

Running is one of the most popular and accessible forms of physical activity worldwide, offering significant cardiovascular and mental health benefits. However, running is also associated with a high incidence of overuse injuries, with studies estimating that 50% of runners experience a debilitating injury each year. Common causes include excessive impact forces, poor foot-strike mechanics (e.g., heel striking vs. midfoot striking), asymmetrical loading between feet, and inconsistent cadence. Many of these issues stem from improper gait mechanics that go unnoticed by runners until pain or injury develops.

Currently, accurate measurement of foot-ground interaction requires gait laboratories equipped with force plates, pressure mats, or instrumented treadmills. These systems are expensive, stationary, and typically limited to clinical, research, or elite athletic settings. While some consumer wearables can estimate cadence or pace using wrist or hip mounted inertial sensors, they do not directly measure foot pressure or impact forces, which are critical indicators of injury risk and running efficiency. There is a societal need for an affordable and wearable system that provides runners with direct insight into how their feet interact with the ground during real outdoor runs. Such a system has implications for public health by helping reduce injury rates, economic welfare by lowering healthcare and physical therapy costs, and athletic accessibility by democratizing biomechanical feedback previously limited to professionals. Additionally, foot mechanics can significantly affect running economy. Having force data and statistics at hand can give runners valuable insights for improvement at comparatively low cost versus a lab analysis. By enabling runners to better understand and adjust their foot force, this project addresses broader concerns related to health, safety, and long-term physical well-being.

We propose a thin-film pressure sensor insole system that integrates directly into the running shoes that measures the force applied by the foot to the ground throughout each step. A flexible sensor array of thin film force sensors embedded on top of the shoe foam (or placed under the insole) will capture pressure through the foot's main contact points (forefoot, heel, and midfoot) and track the pressure distribution of the foot throughout the run. A compact electronics module will attach to the shoe heel or tongue and contain the microcontroller, battery, and Bluetooth modules. The MCU will sample the pressure sensor data, detect foot-strike events, and compute basic metrics such as step count and contact time. Optionally, an inertial measurement unit (IMU) can supplement pressure data to improve robustness and stride estimation. Data is transmitted wirelessly to a smartphone, where it can be visualized, logged, and analyzed. This approach provides runners with direct, actionable feedback on their running mechanics without requiring expensive lab equipment or professional supervision.

b. Visual Aid



This figure represents the pressure sensor array that will be arranged into an insole type shape. This array of sensors will be able to slide into the shoe either on top or on the bottom of the insole. A small ribbon cable will connect the pressure sensor array to a small 3D printed box module that will house the PCB. This PCB will communicate via bluetooth to record all the pressure sensor data.

c. High-Level Requirements List

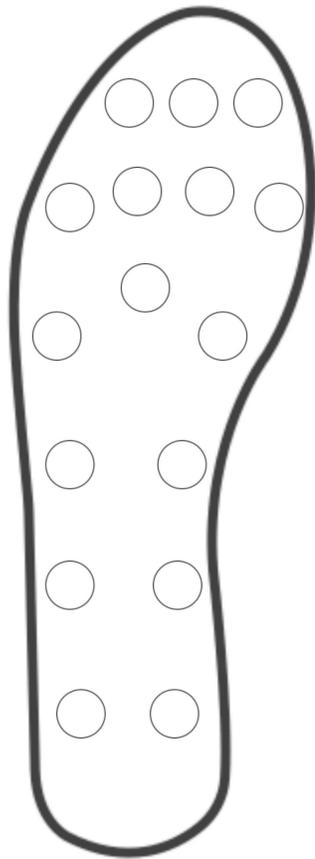
For our project to be successful in solving our problem, it must meet these standards:

- **Efficiency:** The system shall sample plantar pressure sensor data at a minimum rate of 100 Hz and transmit the data over Bluetooth Low Energy with no more than 5% packet loss during continuous operation.
- **Accuracy:** The system shall detect foot-strike events and report running cadence with an accuracy of ± 3 BPM compared to a stopwatch or smartwatch reference over a controlled running trial.
- **Continuity/Longevity:** The device shall operate continuously for at least 1 hour on battery power while performing active sensing and BLE data streaming.

2. Design

a. Physical Design

The Impact Insoles system consists of a flexible sensor insole, a compact electronics module, and a wireless smartphone interface. The insole contains thin-film force sensing resistors (FSRs) positioned at key plantar regions: heel, midfoot (arch), and forefoot. There will be 16 sensors arranged around the foot as shown in the figure below.

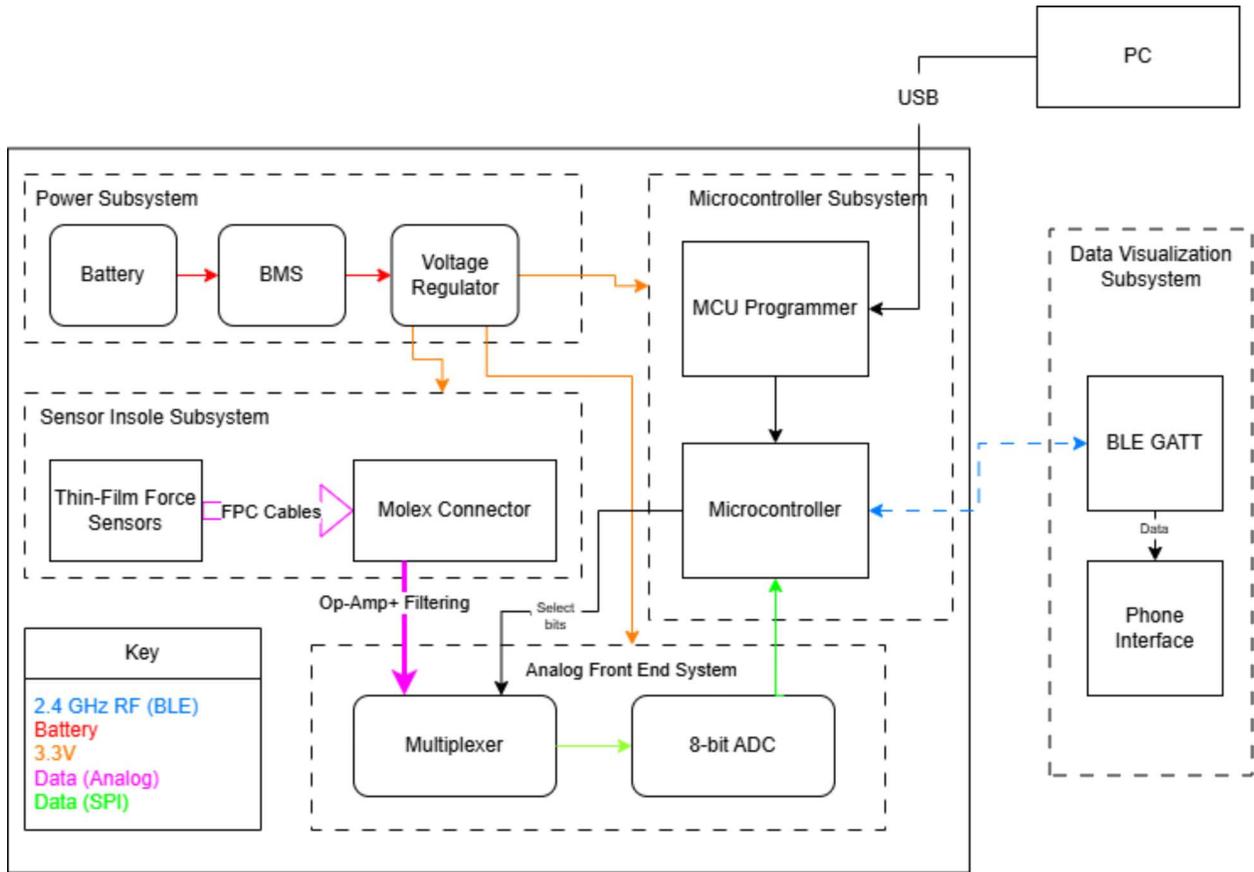


The sensors will be embedded in or laminated onto a polyimide flex PCB that matches the geometry of a standard running shoe insole. The total thickness will be around 0.3 mm to avoid altering the shoe fit or comfort of the shoe. Flexible FFC/FPC ribbon cables route signals from the insole to a small PCB module housed in a low-profile 3D-printed enclosure mounted near the shoe heel or tongue. This enclosure dimensions will be approximately:

- Target size: 50 mm x 50 mm
- Target weight: 50 grams

The enclosure will contain the ESP32 MCU and BLE Module incorporated on a PCB with analog front end. The enclosure will also house a battery compartment to store the Li-Po battery. The enclosure will be designed in such a way that exposes the USB-C port as well as connector ports for debugging and charging. Because the enclosure is 3D-printed, the design will be lightweight in construction. This ensures minimal interference with natural running movements while enabling reliable sensor data acquisition.

b. Block Diagram



c. Subsystem 1: Thin-Film Pressure Sensor Insole Array

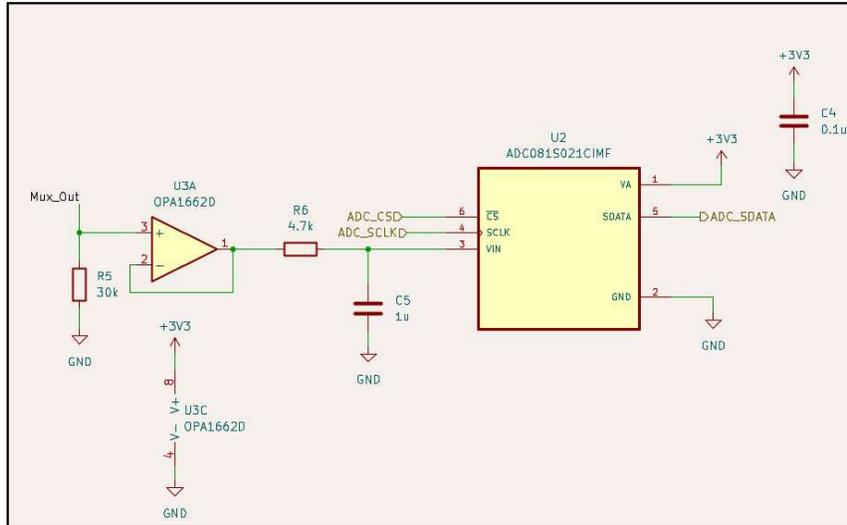
This subsystem is responsible for converting plantar loading or the foot-to insole force during running movement into electrical signals that can be sampled and processed. This subsystem consists of multiple force-sensitive resistors (FSRs) arranged into an array the shape of a shoe insole such that each sensor corresponds to a specific region of the foot such as the heel, midfoot, and forefoot. As pressure is applied during gait events such as the heel strike, mid-stance, and toe-off, the FSR changes resistance in a repeatable manner. This produces a measurable change in sensor output voltage when biased by subsystem 2.

The pressure sensor system will meet the project's goals by ensuring there are enough sensing points to observe pressure distribution changes across the surface of the foot through a running stride. If this subsystem meets all the requirements, the rest of the ImpactInsole system will have valid physical pressure data to sample (subsystem 2), transmit (subsystem 3), and visualize (subsystem 6).

Requirements	Verification
<ul style="list-style-type: none"> The subsystem will incorporate 8-16 pressure sensors, distributed in the heel, midfoot and forefoot regions. 	<ul style="list-style-type: none"> Visually inspect the fabricated insole and count total sensors. Confirm region placement by comparing sensor locations to a foot outline template.
<ul style="list-style-type: none"> Each sensor should have a resistance of ~500 ohms when subjected to a static normal force of $10\text{ N} \pm 0.5\text{ N}$ under nominal bias conditions. 	<ul style="list-style-type: none"> Place the insole on a rigid flat surface and measure unloaded sensor output voltage V_0 using a DMM. Apply a calibrated 10 N load centered on the sensor and measure loaded voltage. Compute $\Delta V = V_{10N} - V_0$ and calculate $R = \frac{V}{I}$ Repeat this process for all of the utilized sensors.
<ul style="list-style-type: none"> Each sensor's tongue-to-insole electrical joint should withstand hundreds of bends of up to 45 degrees from normal, and hold up in real-world conditions 	<ul style="list-style-type: none"> Place the insole on a rigid flat surface and measure unloaded sensor output voltage V_0 using a DMM. Flex the insole at the targeted joint by 45 degrees 100 times Repeat this process for all of the utilized sensors. Inspect joint quality. Use insoles immediately afterward on a real-world 2 hour run test (2x battery spec). Inspect joint quality.

d. Subsystem 2: Analog Front-End + ADC Data Acquisition

This subsystem connects to the sensor board and the purpose is to convert the data appropriately so it can be used by the MCU. The 16 max different sensor data goes into a 16 bit multiplexer so the MCU can read one data at a time. Then the output of the multiplexer is amplified through an op-amp and goes through a low pass filter. The signal finally goes through an ADC to be read by the MCU through SPI. The schematic for this subsystem is shown below.



Requirements	Verification
<ul style="list-style-type: none"> This subsystem will be able to sample all 16 sensors at 100Hz 	<ul style="list-style-type: none"> Run test sampling program on MCU at 100 Hz for 10 seconds. Repeatedly stress all sensors concurrently at high frequency. Can have two people do this with fingers. Compute deltas between each sample and plot for each sensor. Ensure there is no significant weight at the end of the distribution compared to the rest.
<ul style="list-style-type: none"> The subsystem will have enough resolution to track gradual pressure shifts across sensors 	<ul style="list-style-type: none"> Run test sampling program on MCU at 100 Hz for 5 seconds with sensor-insole visualization. Start standing on balls of foot. Over 5 seconds, smoothly transition all weight to the heel. Visually check that pressure transition is tracked by the sensors. Compute sum of squared differences and ensure values are close to zero.

e. Subsystem 3: Microcontroller + BLE Wireless Telemetry

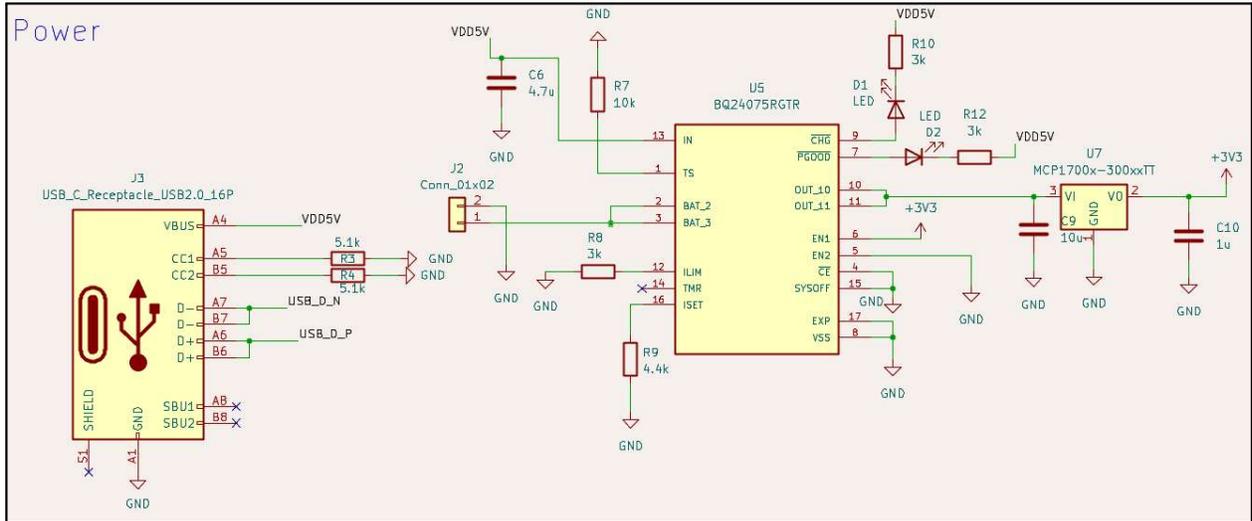
This subsystem is responsible for performing real-time data acquisition and transmitting sensor data through bluetooth to an external device such as an app on a phone. The ESP32-C3 microcontroller controls the analog acquisition chain by selecting sensor channels (via the MUX control lines), triggering the ADC conversions, time-stamping samples, and formatting the resulting data into BLE packets. The device exposes a BLE GATT service that allows a smartphone (or BLE-capable PC) to subscribe to notifications containing the pressure sensor samples. This subsystem directly supports the project’s high-level performance requirements by (1) ensuring the sensor array is sampled at an effective per-channel rate sufficient to resolve foot-strike timing (target ≥ 100 Hz) and (2) sustaining reliable BLE transmission with low packet loss during continuous operation. The MCU operates from a regulated 3.3 V rail and will maintain stable timing, buffer management, and wireless throughput.

Requirements	Verification
<ul style="list-style-type: none"> The subsystem will acquire sensor data at an effective per-channel sampling rate of ≥ 100 Hz for the configured number of active channels. 	<ul style="list-style-type: none"> Enable timestamp logging in firmware for each sample event. Collect timestamped data for at least 30 seconds during continuous operation. Compute the time difference between consecutive samples of the same channel. Calculate effective per-channel sampling rate. Verify minimum observed rate ≥ 100 Hz across all channels.
<ul style="list-style-type: none"> BLE Packet Delivery Requirement. During continuous streaming at full sampling rate, the system shall achieve $\geq 95\%$ packet delivery at 5 m line-of-sight. 	<ul style="list-style-type: none"> Embed an incrementing sequence number in each BLE packet. Stream data continuously for 5 minutes at full sampling rate. Record the received sequence numbers on the phone (e.g., via nRF Connect log). Compute delivery ratio = (received packets / transmitted packets) and verify delivery $\geq 95\%$.
<ul style="list-style-type: none"> The MCU will be able to sustain simultaneous sampling and BLE transmission for ≥ 1 hour without reset, 	<ul style="list-style-type: none"> Fully charge battery and configure system at maximum sampling rate, then begin continuous BLE streaming

buffer overflow, or data stall.	<p>with live reception.</p> <ul style="list-style-type: none"> ● Allow the system to run for 1 hour uninterrupted and monitor for resets, dropped streaming, or firmware crashes ● Verify uninterrupted operation and continued packet reception at end of test
---------------------------------	---

f. Subsystem 4: Power Management + Charging

This subsystem provides the board with power through the LiPo battery as well as being able to use the LiPo battery. The main component of the system is BQ24075RGTR which is a battery management as well as charging regulation. There is a USB-C on the board as well as LiPo battery. The purpose of the USB-C is to charge the battery as well as program the board. BQ24075RGTR is able to control charging by using resistors as reference and has pins to confirm charging is operating properly. When the USB-C is plugged in the board will be powered off the USB-C and use the LiPo otherwise. The LiPo is a 3.7V 400 maH battery that should last the board operating for at least 1 hour. The schematic for this subsystem is shown below.



Requirements	Verification
<ul style="list-style-type: none"> The power subsystem will have the ability to charge battery at 200mA 	<ul style="list-style-type: none"> Connect USB-C power cable to USB-C multimeter, and multimeter to charging port. Commence charging and ensure 200 ± 5 mA charging rate throughout duration
<ul style="list-style-type: none"> The subsystem should be able to supply ICs and MCU with stable 3V3 and 5V 	<ul style="list-style-type: none"> Run the main program for 5 minutes. Measure voltage at 3V3 and 5 V testpoints. Ensure $3*SD$ is less than 1% of each.
<ul style="list-style-type: none"> The Battery life meets desired specification (1 hour minimum) 	<ul style="list-style-type: none"> Time the main program until shutoff. Ensure shutoff does not occur before 1 hour.

g. Subsystem 5: Phone Interface / Data Visualization

This subsystem provides the wireless user-facing interface for receiving and displaying the pressure data streamed from subsystem 3. We will use a smartphone as the BLE central device. We will create a phone app that will scan for the device’s BLE advertisements, program a connection and subscribe to the pressure data notifications. The phone interface is responsible for converting the raw streamed packets into meaningful, interpretable feedback for runners while preserving the integrity of the underlying measurements.

On the data path, the phone receives BLE notification packets containing (at minimum) sensor sample values and a sequence number and/or timestamp. The interface detects gaps in the sequence count to estimate packet loss and can flag communication quality issues during use. Once data is received, the phone application (or a BLE viewer such as nRF Connect during prototyping) parses the payload into a per-sensor time series and applies lightweight processing to extract high-level running metrics. At minimum, the phone displays: (1) live per-sensor pressure traces (or an aggregated “total pressure” signal), and (2) step events and cadence derived from pressure peaks and threshold crossings. The visualization is designed to operate in real time, updating at a rate fast enough to appear continuous to the user while still logging the full-rate sensor data in the background if enabled. The interface must remain robust under real-world usage conditions. Some example environments include having the phone in a pocket, varying BLE link quality, and the runner starting/stopping or changing pace. Therefore, the phone app will have clear session controls such as a start/stop stream button and optional save/export data button.

Requirements	Verification
<ul style="list-style-type: none"> The phone interface will correctly decode the incoming BLE packets into individual sensor channels without channel corruption. 	<ul style="list-style-type: none"> Apply pressure to only one of the sensors while keeping all others untouched. Observe live data display and confirm only the corresponding channel changes significantly. Repeat tests for all the different sensor locations. Verify no unintended channels respond during isolated loading.
<ul style="list-style-type: none"> The phone interface shall update displayed sensor values or plots at a rate sufficient to appear continuous during active streaming. 	<ul style="list-style-type: none"> Begin live streaming session. Rapidly tap and release pressure on a sensor at ~2–3 Hz. Observe visualization and confirm pressure peaks appear in real time without noticeable lag or freezing.
<ul style="list-style-type: none"> The phone interface will compute cadence from the pressure-derived foot-strike events within ± 3 BPM compared to a reference measurement. 	<ul style="list-style-type: none"> Conduct a controlled short 3–5 minute running test that measures the cadence independently using a stopwatch or smartwatch. Record feet with 120 fps video camera. Process foot-strike events and compute cadence over similar time intervals manually based on camera data Compare sensor-computed cadence to the referenced control test and verify absolute error ≤ 3 BPM in each test window.

h. Subsystem 6: Optional Motion Sensing (IMU)

This extra optional subsystem will provide inertial measurements while the user is running. We chose the LSM6DSOXTR as our IMU of choice. It will take accelerometer/gyro data to gather speed data, estimate and improve stride data and length, and improve cadence robustness when the pressure signals are noisy. The subsystem will incorporate a 3-axis accelerometer and gather measurements that we can later use to complement our pressure data measurements for a better overall user experience.

The accelerometer communicates with the microcontroller over I²C. The subsystem must support running accelerations without saturating, provide a stable stationary baseline, and maintain consistent timing relative to the pressure samples. If implemented properly, this subsystem can be used to validate foot-strike timing as well.

Requirements	Verification
<ul style="list-style-type: none"> The accelerometer shall sample 3-axis acceleration at a rate of 100 Hz \pm 5 Hz during active operation. 	<ul style="list-style-type: none"> Configure the accelerometer output data rate to 100 Hz. Enable firmware timestamp logging at each accelerometer read event then record timestamps for 30 seconds during steady operation. Compute the sampling frequency from the average time difference between consecutive samples.
<ul style="list-style-type: none"> The accelerometer operates with a dynamic range of at least $\pm 4g$ and shall not saturate during steady running at cadences between 140–200 BPM 	<ul style="list-style-type: none"> Configure the accelerometer full-scale range to $\pm 4g$. Perform a 2-minute running trial within the 140–200 BPM cadence range while logging acceleration data. Inspect logged acceleration values for clipping at $\pm 4g$
<ul style="list-style-type: none"> When stationary on a horizontal surface, the measured acceleration magnitude should equal 1.00 g \pm 0.10 g. 	<ul style="list-style-type: none"> Place the device flat on a rigid horizontal surface and record acceleration for 10 seconds. Compute average acceleration magnitude over the interval and check to see if the average magnitude is between 0.90 g and 1.10 g.

i. Tolerance Analysis

A significant consideration on the data transmission side of this project is the balance between battery life, transmission rate, and memory. When lowering the transmission rate between the phone and the insole system to conserve battery, more data needs to be stored on device before it can be cleared. This means that the flash storage size and speed will set the limits for our battery management approach. Additionally, these storage parameters will determine if the device can be used headless, an important feature for runners running light. On top of this, to decrease complexity and size of the final product, it is highly preferable to only use the inbuilt flash memory for these roles rather than external solutions.

If each sensor generates 1 byte of data per poll, one foot's insole array generates 16 bytes of data per poll. At our max poll rate of 100 Hz, the array will generate data at a rate of 1.6 kB/s max. This is within spec of the ESP32 flash memory, meaning that the flash memory can handle the writing and storage of this data. Additionally, this means that if we set aside 14 MB of the 16MB flash memory in the ESP32, we can store 8750 seconds (~145 minutes) of raw data max before clearing. For data transmission considerations, this is more than enough space to conduct transmission events at any feasible rate we desire. For headless mode, this duration is long enough to accommodate a standard two-hour run with headroom. Therefore, it is feasible to use the inbuilt flash memory of the ESP32 to ensure long battery life and headless functionality in our device.

3. Cost and Schedule

a. Cost Analysis

The total cost to purchase the parts to create one device for one shoe before shipping is \$114.52 (see table below). Shipping costs are an estimated \$6.99 since all parts that are not sourced from the Illinois Eshop can be found on Digikey. We can also assume Illinois state tax of 6.25%.

This brings the total cost to create one ImpactInsole to be \$128.67. We can expect a reasonable salary of \$40/hour * 2.5 * 80 hours to complete = \$8000 per team member. With our 3 members, this comes to \$8000 * 3 = \$24,000 in labor costs. Therefore the total costs of the project is \$24,128.67.

Note: The cost of the initial creation of ImpactInsoles research and development, as well as the option to create multiple ImpactInsoles (perhaps you would want to use them on both feet) will increase this price.

Part Description	Manufacturer	Price per part	Quantity	Total Price
FSR 402 Round Force Sensing Resistor	Interlink Electronics	6.39	16	102.24
IC BATT CHG LI-ION 1CELL 8SOIC (BQ24075RGTR)	Texas Instruments	2.36	1	2.36
ESP32-S3-WROOM-1 WiFi + BLE Module	Espressif Systems	Free From Eshop	1	0
CD74HC4067 16-Channel MUX	Texas Instruments	0.63	1	0.63
ADC081S021 8-bit ADC	Texas Instruments	1.56	1	1.56
OPA1662 Dual Op-Amp	Texas Instruments	0.35	1	0.35
LSM6DSOX IMU	STMicroelectronics	4.07	1	4.07
MCP1700-300 LDO Regulator	Microchip Technology	0.51	1	0.51
USB-C Receptacle (GT-USB-7010ASV)	G-Switch	0.49	1	0.49
JST S2B-PH Battery Connector	JST	0.11	1	0.11
Molex 503480-1800 18-Pin Connector	Molex	1.5	1	1.5
Capacitors 0603 (10uF, 1uF, 0.1uF, 4.7uF)	Various	Free From Eshop	13	0
Resistors 0603 (4.7k (3), 10k (2))	Various	Free From Eshop	5	0
Resistors 0603 30k	YAGEO	0.1	1	0.1
Resistors 0603 5.1k	YAGEO	0.1	2	0.2
Resistors 0603 4.4k	YAGEO	0.1	1	0.1
Resistors 0603 3k	YAGEO	0.1	3	0.3
LEDs (0805, 1206)	Various	Free From Eshop	2	0
Push Button Switch	Generic	Free From Eshop	1	0
			Total:	114.52

b. Schedule

This table serves as a guideline for the tasks we plan to accomplish each week leading up to the final demo to get the project completed.

Week	Task	Person
March 1 - March 7 <i>March 5: 2nd PCB Order</i>	Order parts for 1st PCB order Revisions to PCB Design Begin writing driver functions for sensors Start board assembly Prototype sensor with breakoutboard for breadboard demo	Everyone Matthew Joseph Aarush Everyone
March 8 - March 14 <i>March 9-March 11: Breadboard Demo March 12: 3rd PCB Order</i>	Revisions to PCB Design Establish BLE connection from ESP32 to phone Order more parts if needed	Matthew Joseph Everyone
March 15 - March 21	<u>*Spring Break*</u> Look over drivers and design implementation	Everyone
March 22 - March 28 <i>March 26: Final PCB Order</i>	Finalize final PCB design Finalize final sensor array layout Get sensor data transmission from MCU to phone app	Matthew Aarush Joseph
March 29 - April 4	Have phone app fully built by 4/5 for progress demo PCB Debugging	Joseph Aarush
April 5 - April 11 <i>April 6-April 8: Progress Demo</i>	3D print enclosure Assemble the running shoe First Trial Run	Matthew Aarush Joseph
April 12 - April 18	Extensive Testing, debugging	Everyone
April 19 - April 25 <i>April 20-April 24: Mock Demo April 23/April 24: Mock Presentation</i>	Final Debugging, fine-tuning design <i>April 25: Illinois Half Marathon (use project?)</i>	Everyone Joseph
April 26 - May 2	<i>April 27-April 29: Final Demo April 30/May 1: Final Presentation</i>	Everyone

4. Ethics and Safety

The main source of ethical consideration in this project is the potential misunderstanding of this system as a medical device. Since the system tracks biometrics and may offer interpretation of that data in the form of statistics or algorithmically-calculated suggestions, users could assume it can assess clinical injury or provide medical recommendations. In accordance with the IEEE Code of Ethics, which emphasizes holding paramount the safety, health and welfare of the public, we will need to clearly present the device as a training and performance aid only. This means we may also have to include disclaimers describing the product's limitations, as well as advising users to make their own personal medical judgment with the data.

The IEEE Code of Ethics also requires engineers to be honest and realistic in their claims based on the available data. Due to this, the reported metrics from our device will have to be validated against reference measurements. Safety is also a top priority, as an electronic device will be attached to the foot, a crucial part of the human body. The system must remain lightweight, securely mounted, and low-profile to not interfere with normal movement or create a tripping or discomfort hazard. Since the device will operate with a battery, a proper enclosure must be created to prevent overheating and electrical risks during use.

The societal impact of this system is not necessarily widespread, but can still be significant. This product appeals to a niche market of those who care about improving their running stride and maintaining good form. This system can provide a more accessible option for users outside of expensive laboratory environments. The positive effects compound with more miles run, so avid runners in particular can benefit greatly from this.

5. Citations

- [1] National Center for Biotechnology Information (NCBI), “Article available at PMC8500811,” PubMed Central (PMC). Available at: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8500811/>. Accessed Feb. 27, 2026.
- [2] ScienceDirect, “Article available at S2950273X24000729,” ScienceDirect. Available at: <https://www.sciencedirect.com/science/article/pii/S2950273X24000729>. Accessed Feb. 27, 2026.
- [3] National Center for Biotechnology Information (NCBI), “Article available at PMC7734358,” PubMed Central (PMC). Available at: <https://pmc.ncbi.nlm.nih.gov/articles/PMC7734358/>. Accessed Feb. 27, 2026.
- [4] Interlink Electronics, “FSR 402 Round Force Sensing Resistor,” product highlight page. Available at: <https://www.digikey.com/en/product-highlight/i/interlink/fsr-402-round-force-sensing-resistor>. Accessed Feb. 27, 2026.
- [5] Interlink Electronics, “FSR Model 402 Force Sensor,” web store page. Available at: <https://buyinterlinkelectronics.com/products/fsr-model-402>. Accessed Feb. 27, 2026.
- [6] Interlink Electronics, “FSR 402 Force Sensing Resistor Datasheet,” datasheet, 2019. Available at: https://files.seeedstudio.com/wiki/Grove-Round_Force_Sensor_FSR402/res/FSR402.pdf. Accessed Feb. 27, 2026.
- [7] Molex LLC, “Premo-Flex FFC/FPC Jumper Cables,” product category page. Available at: <https://www.mouser.com/c/wire-cable/ffc-fpc-jumper-cables/?m=Molex&tradename=Premo-Flex>. Accessed Feb. 27, 2026.
- [8] Molex LLC, “5034800400 FFC/FPC Connector,” product page. Available at: <https://www.molex.com/en-us/products/part-detail/5034800400>. Accessed Feb. 27, 2026.
- [9] Molex LLC, “5034800440 Connector, 0.50 mm Pitch,” Digi-Key product page. Available at: <https://www.digikey.com/en/products/detail/molex/5034800440/5021353>. Accessed Feb. 27, 2026.

- [10] Texas Instruments, “ADC081S021CIMF/NOPB 8-Bit ADC,” Mouser product page. Available at: <https://www.mouser.com/ProductDetail/Texas-Instruments/ADC081S021CIMF-NOPB>. Accessed Feb. 27, 2026.
- [11] Texas Instruments, “ADC081S021 8-Bit Analog-to-Digital Converter,” datasheet, 2015. Available at: <https://www.ti.com/lit/ds/symlink/adc081s021.pdf>. Accessed Feb. 27, 2026.
- [12] Texas Instruments, “CD74HC4067SM96 Analog Multiplexer,” Digi-Key product page. Available at: <https://www.digikey.com/en/products/detail/texas-instruments/CD74HC4067SM96/390310>. Accessed Feb. 27, 2026.
- [13] Texas Instruments, “CD74HC4067 High-Speed CMOS Analog Multiplexer/Demultiplexer,” datasheet, 2017. Available at: <https://www.ti.com/lit/ds/symlink/cd74hc4067.pdf>. Accessed Feb. 27, 2026.
- [14] Texas Instruments, “TLV9062IDR Operational Amplifier,” Digi-Key product page. Available at: <https://www.digikey.com/en/products/detail/texas-instruments/TLV9062IDR/7364940>. Accessed Feb. 27, 2026.
- [15] Texas Instruments, “TLV9062 Low-Power, Rail-to-Rail Operational Amplifier,” datasheet, 2018. Available at: <https://www.ti.com/lit/ds/symlink/tlv9062.pdf>. Accessed Feb. 27, 2026.
- [16] Espressif Systems, “ESP32-C3-WROOM-02-H4 Wi-Fi/BLE Module,” Digi-Key product page. Available at: <https://www.digikey.com/en/products/detail/espressif-systems/ESP32-C3-WROOM-02-H4/14553028>. Accessed Feb. 27, 2026.
- [17] Espressif Systems, “ESP32-C3-WROOM-02 Datasheet,” datasheet, 2022. Available at: https://documentation.espressif.com/esp32-c3-wroom-02_datasheet_en.pdf. Accessed Feb. 27, 2026.
- [18] Tag-Connect LLC, “TC2030-IDC Programming/Debug Cable,” Digi-Key product page. Available at: <https://www.digikey.com/en/products/detail/tag-connect-llc/TC2030-IDC/4571121>

. Accessed Feb. 27, 2026.

[19] STMicroelectronics, “LSM6DSOXTR 6-Axis IMU,” Digi-Key product page. Available at: <https://www.digikey.com/en/products/detail/stmicroelectronics/LSM6DSOXTR/9841882>

. Accessed Feb. 27, 2026.

[20] STMicroelectronics, “LSM6DSOX iNEMO Inertial Measurement Unit,” datasheet, 2021. Available at: <https://www.st.com/resource/en/datasheet/lsm6dsox.pdf>

. Accessed Feb. 27, 2026.

[21] Institute of Electrical and Electronics Engineers (IEEE), “IEEE Code of Ethics,” IEEE Board of Directors, June 2020. Available at:

<https://www.ieee.org/about/corporate/governance/p7-8.html>

. Accessed Feb. 27, 2026.