STRIDE: Wearable Running Gait Analysis

Luke Andresen
Duke Pratt School of Engineering
Durham, NC, USA
luke.andresen@duke.edu

George Fang
Duke Pratt School of Engineering
Durham, NC, USA
george.fang@duke.edu

Gavin Kitch
Duke Pratt School of Engineering
Durham, NC, USA
gavin.kitch@duke.edu

Evan Lankford
Duke Pratt School of Engineering
Durham, NC, USA
evan.lankford@duke.edu

Figure 1: Final STRIDE Wearable Device.

Abstract

This paper outlines the development of a wearable device that analyzes a user's running stride. The prototype is easily integrated into most adult running shoes and interfaces with a Garmin watch for data collection, visualization, and storage. It uses force-sensitive resistive insoles to detect pressure during a step, an inertial measurement unit (IMU) to track the path of the foot, and runs on an ESP32-S3.

Duke ECE 469, Durham, NC

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing design and evaluation methods**; **Mobile devices**.

Keywords

Wearable, Stride, Gait Analysis, Foot Path, Garmin

ACM Reference Format:
Luke Andresen, George Fang, Gavin Kitch, and Evan Lankford. 2025. STRIDE: Wearable Running Gait Analysis. In *Proceedings of Wearable and Ubiquitous Computing Systems Design (Duke ECE 469).* ACM, New York, NY, USA, 8 pages.

1 Introduction

1.1 Motivation

When running with existing wearable technologies, a user unlocks the ability to track their pace, gait symmetry, running path, and many other useful data-points. However, these existing systems are unable to capture the full picture of the user's run. Specifically, they are unable to track detailed biomechanical insights like foot path, and foot impact zones over the course of a run. Runners and coaches often rely on subjective observations or expensive biomechanical lab testing to assess running form. A cost-effective and portable solution would provide runners with personalized, continuous, data-driven insights to improve form and efficiency and reduce injury risk. Thus, we aimed to create a system that fills these gaps, sensing and analyzing a runner's strike and stride to improve their running experience.

1.2 Contribution

Through our work, we developed a unique sensor suite that can be inserted into running shoes without modification. We also created a data visualization/analysis application through Garmin's ConnectIQ development platform. Data is displayed in real time, and continuously aggregated for post-run analysis. Users can view and track their foot pressure distribution and foot path while running, giving them key metrics to review their form. Ultimately, the STRIDE platform allows for the integration of advanced running metrics into a user's regular running routine.

1.3 Product Requirements

In order for this product to be considered successful and advance the fitness wearable space in a meaningful way, we defined the following requirements:

- Accurate collection of data from force sensitive resistor pads to determine the force distribution for each step.
- Analysis and processing of IMU data to calculate the foot path.
- A ConnectIQ app for the Garmin Forerunner 265, including data visualizations for each running session.
- Data transmission via Bluetooth Low Energy (BLE) between the microcontroller unit and Garmin watch.
- A rechargeable power system to enable real-world use cases.

1.4 Complex Characteristics

In developing STRIDE, we encountered a variety of complexities. First, due to the project's development in the Garmin ecosystem, the product needed to interface with their existing devices. The application was built in Monkey C on the ConnectIQ platform. We had to use BLE technology to interface with a Garmin watch, which introduced a multitude of challenges. In this vein, intensive data compression was required to create a real-time product. On the embedded side, complex signal processing, data analysis, and filtering were used to visualize the foot path from IMU data. A PCB was also developed to simplify the circuitry.

2 Background

Gait analysis looks at biomechanical metrics that are both popular and important for runners to improve performance and reduce the risk of injury. Beyond the basic performance metrics provided by current wearable devices on the market, two key running statistics are foot pressure and foot path. Pressure distribution refers to how force is spread across the bottom of the foot while running. It provides insight into where the foot strikes the ground, how weight transfers across the foot, and how evenly the load is managed during a step. Foot path refers to the motion pattern of the foot throughout a stride. Tracking foot path can help runners detect asymmetries, too much or little vertical lift, and excessive lateral motion.

3 Our Solution: STRIDE

Our final prototype design integrates several components to generate novel methodologies for tracking these key aspects of gait analysis. This solution is the product of many iterations of hardware configurations, signal processing, physical attachments, and data visualizations. The result is a modular, compact device that is compatible with existing shoes and Garmin watches, offering users in-depth insights into valuable running statistics for form and efficiency improvement.

3.1 Engineering Standards Used

We used Jira for task management and to facilitate our weekly sprints. We added tickets each week for our tasks and had two weekly scrums about our progress on them. We used Git for version control of our embedded and application software code bases. We used a Gantt Chart to track our progress toward project goals and requirements each week.

3.2 Project Components

Hardware

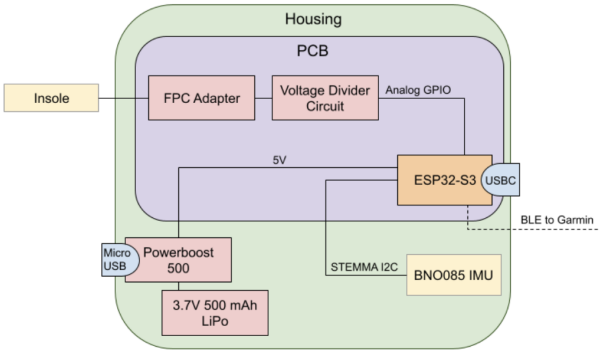

Figure 2: Hardware Block Diagram.

Microcontroller: STRIDE is built upon a BLE-enabled ESP32-S3 microcontroller. We selected Adafruit's QT Py S3 development

board due to its small form factor, 8+ ADC channels, and STEMMA QT I2C port.

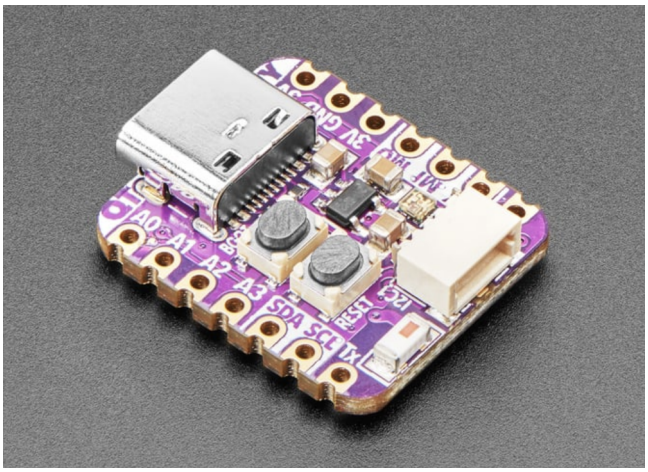

Figure 3: Adafruit QT Py ESP32-S3.

Force Sensitive Resistor Insole: To detect the pressure that the foot applies, we used Alpha's MF18-N-0A8-A01 pre-made Force Sensitive Resistor (FSR) Insole. This sensor has 8 FSR pads in the shape of a foot that vary in resistance based on the pressure applied to each.

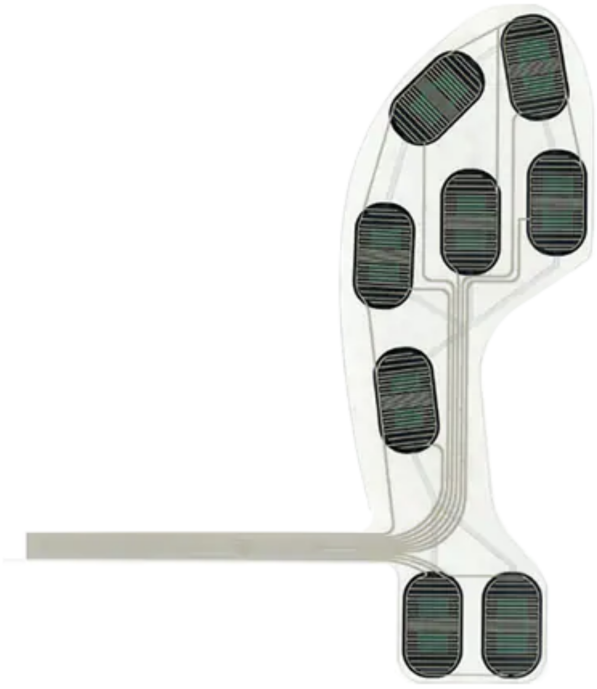

Figure 4: Pre-made FSR Insole.

Inertial Measurement Unit: To measure the path of the foot, we used Adafruit's BNO085 breakout board due to its high-performance, rich supporting library, and STEMMA QT I2C port.

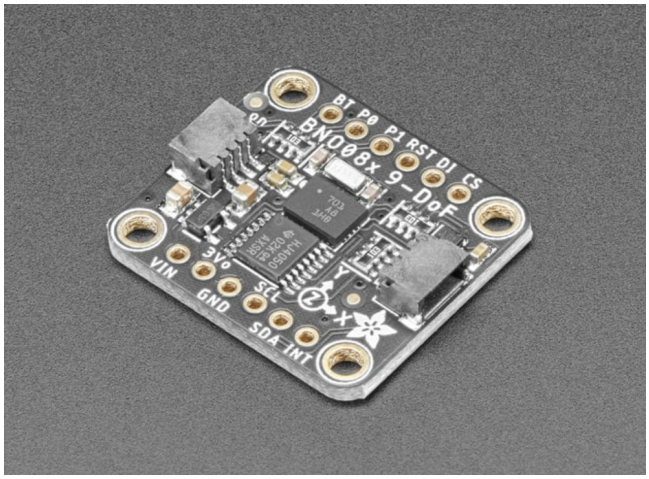

Figure 5: Adafruit BNO085 IMU Breakout Board.

Power Circuitry: To power the system, we combined a 3.7V 500 mAh lithium-ion battery with Adafruit's PowerBoost 500, which steps the voltage of the battery up to 5V to power the MCU while also enabling easy recharging of the battery through a micro-USB port.

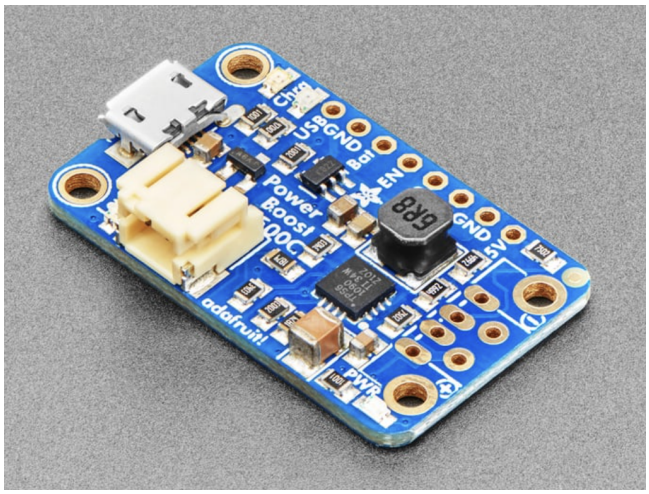

Figure 6: Adafruit PowerBoost 500.

Printed Circuit Board: To read the changing resistance values from the FSR insole, we built 8 voltage divider circuits on a breadboard using an FPC adapter. We experimented with resistor values for the circuit and found $10k\Omega$ to offer the best data resolution. However, we quickly found that this to be extremely restrictive in testing due to the delicate nature of the breadboard based circuit. Therefore, we used EAGLE to develop a PCB that combines the MCU, the FPC adapter, and the voltage dividers into one streamlined package.

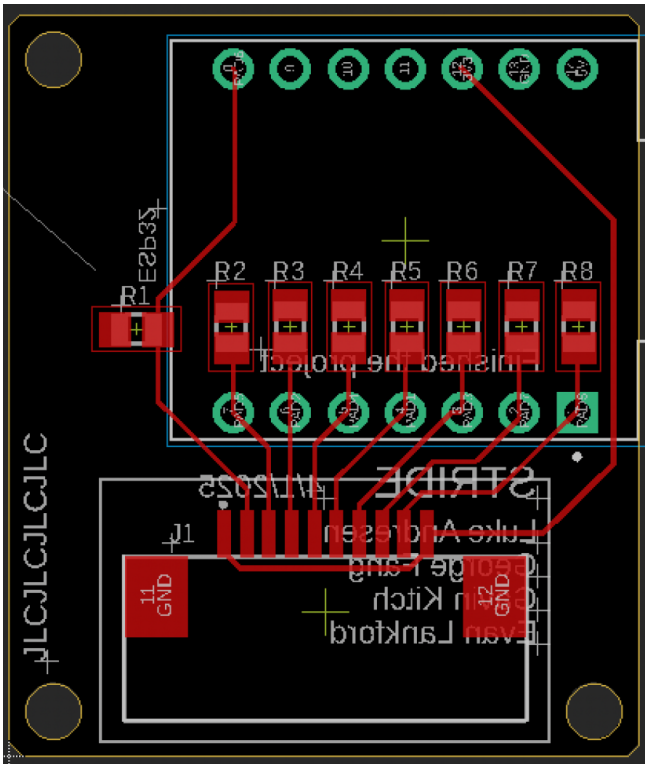

Figure 7: PCB Board Layout.

Housing: We created a 3D printed box to house the components on the shoe. We added holes for the FSR insole connector, charging, and reprogramming. There is a clip to attach the box to the side of the user's shoe. To hold the components, we screwed each into threaded inserts.

Embedded Development

Microcontroller Script Overview: At its core, the script that the ESP32 is running constantly checks the FSR for sufficient pressure, indicating a step. If no step is detected, the IMU data is sampled and stored. If a step is detected, the MCU rapidly collects and stores FSR data until there is no longer sufficient pressure on any pad. At the end of the step, the FSR data is compressed and packaged with the processed IMU data and transmitted over BLE.

Bluetooth Connectivity: We simply used the ESP32-S3 BLE library to create a characteristic. The MCU will only collect and send data if it has an active connection listening for updates.

IMU Processing: The IMU collects gyroscopic and linear acceleration data. With the quaternion about the global x-axis (from the exterior, into the ankle), we calculate the upward and forward acceleration with trigonometry. Raw data is fed through a low-pass filter and thresholding to remove insignificant values, then integrated to calculate relative velocity. Then, a high-pass filter is used to remove drift, before relative position is calculated. Finally, we select 4 significant points: at the minimum and maximum horizontally, at the maximum vertically, and the midstep.

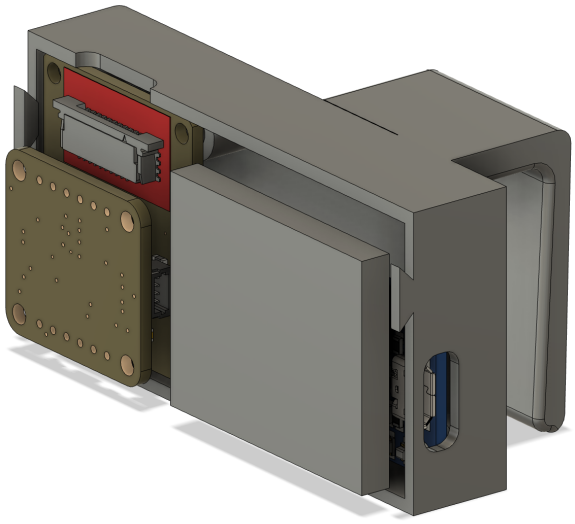

Figure 8: Housing CAD.

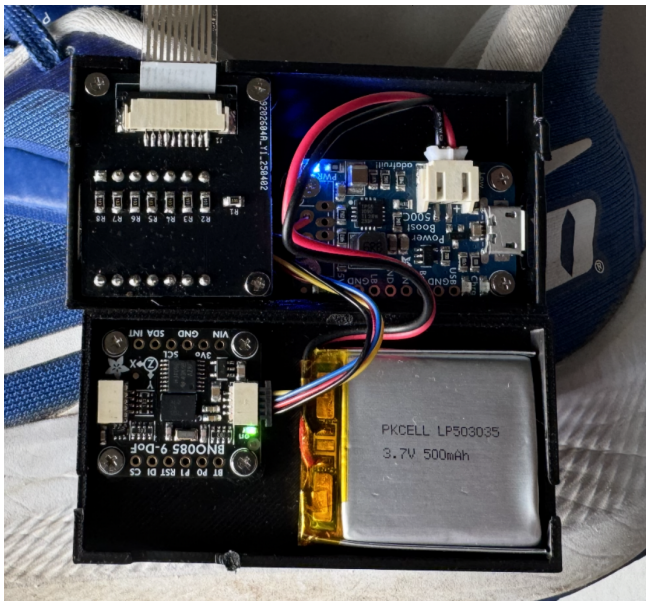

Figure 9: Assembled Device Interior.

Data Compression: Due to restrictions with connect IQ, we needed to compress thousands of bytes of FSR and IMU data into 20 bytes for transmission in one packet to avoid latency. To do this, we first tested and identified the best relative indices that illustrate the land, load, and launch of the step. Next, we tracked the maximum pressure value recorded during each step and used this to scale all values from the selected indices to a value between 0 and 15, making them 4 bits. Next, these 3 sets of 8 values could now be combined, with one value in the 4 most significant bits of a byte and one occupying the 4 least significant bits, taking 12 total bytes. We also transmit the maximum value, scaled from 0 to 255 using

the maximum possible ADC value of 4095. This value could be used to un-scale the data on the ConnectIQ side. Lastly, we packed the 4 IMU points (pairs of x and y values) into the final 7 bytes by omitting the y value from the final point, as the maximum y must occur when the foot returns to the ground.

ConnectIQ Development

Application Overview: The Garmin ConnectIQ STRIDE application allows users to scan and connect to the hardware module via BLE, view live data visualizations of foot pressure and foot path, start and stop running sessions, and view post session analysis of a run. It uses a View-Controller-Model flow and ConnectIQ Persisting Data for on-watch storage. The user interacts with the view screens of the application, which each have controllers that render content and process requests (taps, swipes, etc). The controllers in turn interact with models that store and send device-specific data. There are four main view stacks in the application, as seen in the figure below.

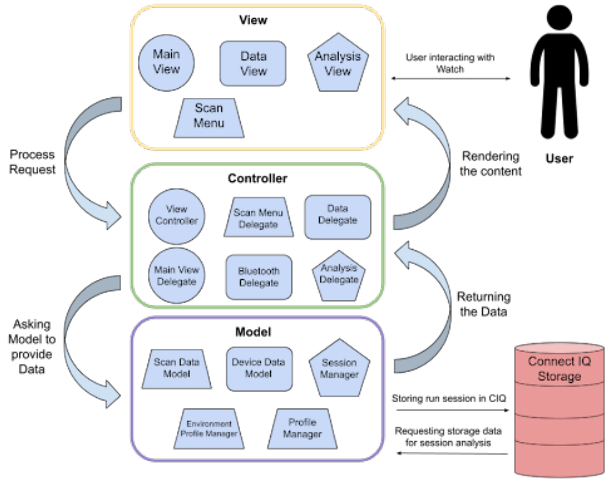

Figure 10: ConnectIQ Block Diagram

Home Screen: A simple landing page for users opening the application prompts users to hold the "Up" button to enter the scanning menu.

Scan Menu: After pressing the "Up" button, the user is prompted to select which foot's hardware they want to Bluetooth scan for. They also have the option to stop scanning. This view is controlled by a delegate that handles item selection and is linked to a few models related to scanning and storing Bluetooth results. It looks for a specific Bluetooth service being advertised by the STRIDE hardware and returns an option to "Tap to Connect" when the service is found. Once tapped, the application pairs to the Bluetooth service and awaits data.

Figure 11: Home View

Figure 12: Scan Menu View

Data View: As soon as the watch starts receiving Bluetooth packets from the hardware module, it first decompresses and decodes the data. This occurs in the models beneath the data view, which pass the decoded values to the data view controller. Then, the foot pressure values are converted to a yellow red color scale and displayed spatially on the view screen in the shape of a foot. Each dot

corresponds to a pressure pad in the insole. When a user swipes on the screen, they move to the foot path screen that displays IMU data. This screen displays the x and y movement over the course of each step in a graphical format. For both the foot pressure and foot screen pages in the data view, the screens and models are refreshed each time a new Bluetooth packet is received. The user has the option to select the start/stop button while in the data view to begin recording a running session. Once the user finishes the session by pressing stop, all of the data collected during the run is aggregated and sent to persistent data storage on the watch.

Analysis View: After finishing a session on the data view, the controller requests the most recent session from persistent storage and pushes the analysis view. Persistent storage uses key-value pairs, and each run is assigned a unique key based on the time when it was started. Given Garmin's 8KB size limitation for key-value pairs, we store step count and cumulative values instead of every single data point collected. Similar to the data view, the analysis has two pages that users can swipe between. The first page displays aggregated foot pressure distribution from the three snapshots within each step: land, load, and launch. On the back end, basic averaging is done with the data pulled from persistent storage. The second page displays average foot path IMU data. It uses the same method of averaging on the back end as the first page. Once done reviewing the analysis page, users can press the back button to return to the data view and begin a new session. The two figures below show the two distinct pages within the analysis view that users can swipe between.

Figure 13: Foot Pressure Analysis View

Figure 14: Foot Path Analysis View

3.3 Risks

Several potential risks were identified during the development and testing of STRIDE. On the hardware side, the flexible cable connecting the FSR insole to the housing is relatively short, and improper routing around the sock or shoe could create tension that may pull the connector loose during use. Additionally, the 3D-printed housing is not fully waterproof, so the device is vulnerable to damage if exposed to rain or wet running conditions. From a software standpoint, while running session data is successfully stored on the Garmin watch, there is currently no functionality to delete stored sessions, which could eventually lead to storage constraints over long-term use. Although these risks were not critical during prototype testing, they represent important considerations for future design improvements to enhance device robustness and usability.

3.4 Trade-offs

A key part of settling on the final prototype was considering various design choices and the impacts of their tradeoffs. In particular, we evaluated the following components throughout the prototyping process.

Development Boards

At the core of our hardware system, the development board we selected was crucial to the capability of our device. Following the guidance of the Garmin team, Professor Younes, and previous groups, we considered three microcontroller units. In particular, we evaluated processing power, ease of integration with the Garmin system, BLE capabilities, power consumption, and compatibility with our sensor suite.

Nordik nRF52840-DK: This was the suggestion of the Garmin team,

and checks the boxes for minimal power consumption, integration with Garmin, and BLE. However, it lacks in analog pins, and has extremely minimal on-board processing power.

Unexpected Maker TinyS3: A previous group whose project required more processing power suggested the TinyS3, and provided a framework for BLE and compatibility with Garmin. However, the boilerplate repositories did not include support for this option. It has substantial analog-to-digital converter (ADC) pins for the FSR, but these overlap with SDA/SCL pins for the IMU, and does not offer a STEMMA connector as an alternative.

Adafruit QT Py ESP32-S3: A small form factor, substantial processing power, a STEMMA cable, and enough ADCs to support our peripheral sensors made this an inviting choice. It has BLE support, but while it is built around the same microcontroller as the TinyS3, there was no existing evidence of compatibility with Garmin.

A few crucial barriers impacted our choice. Primarily, due to the limitations within ConnectIQ for BLE data transmission (20 bytes per packet), we needed a board that permitted substantial processing power to analyze our raw data immediately, making the nRF52 a less-viable option. Also of importance was the compatibility with our selected sensor suite, highlighting the QT Py ESP32-S3 as a clear frontrunner. However, the Garmin compatibility was up in the air, so we required testing before finalizing this as our choice. We cross-referenced the existing tutorials and previous team's BLE transmission template to develop code for the QT Py, and proved successful data transfer via LED control between it and the Garmin watch. With this proof of concept, we were able to select this as our final choice.

Device Placement

The placement of our module housing also required testing for functionality, comfort, and level of intrusion in exercise. We were constrained by a minimally viable weight and form factor for our prototype, as well as the restrictive cable of the FSR. We considered three possibilities for attachment, and evaluated them based on these standards in our testing.

Shoelace Attachment: The shoelace attachment felt natural while running, and was aesthetically appealing to users. In addition, it is well positioned to capture footpath and is unobtrusive. However, connecting to the FSR attachment, which originates on the exterior side of the foot, without stressing the cable or adding a second module, causes some difficulty.

Ankle Strap: In testing, this felt the least-intrusive in terms of additional weight on the foot, and while still an easy attachment to the FSR. However, a few drawbacks made this a poor solution. First, any pronation or vertical rotation of the ankle——a frequent occurrence in running——causes potential strain on the FSR cable. Second, an ankle strap could lead to reduced comfort, and reduce the compatibility across users. Finally, the overall form factor and aesthetic as a marketable product is reduced.

Shoe Clip-on: The clip-on offers all the positives of the shoelace attachment pertaining to comfort, ease of use, and aesthetics. In fact, our concerns of discomfort on the ankle were mitigated in testing, as the natural position of the foot results in no contact with the clip. The added benefit of looping over the FSR attachment for easy connection with minimal strain made this the best solution.

3.5 Evaluating Against Requirements

We evaluated STRIDE's performance against the product requirements outlined at the beginning:

Accurate Collection of Force Data: To verify the functionality of the FSR insole, we performed controlled stepping tests where users deliberately struck the heel or forefoot repeatedly. We analyzed the resulting pressure heatmaps and console logs to ensure that pressure was accurately detected in the expected zones. Post-run heatmaps consistently matched known strike patterns, confirming that force data collection was accurate and reliable.

Footpath Estimation: Testing footpath estimation was more nuanced. We performed runs with deliberately exaggerated step patterns — including high knees, butt kickers, and forward shuffles — to verify that the IMU-based foot trajectories reflected distinct gait styles. High knees resulted in a higher vertical arc in the path graphs, butt kickers showed a pronounced backward trajectory, and shuffling showed minimal vertical and backward movement. These tests confirmed that STRIDE could differentiate between varied running forms based on IMU data.

ConnectIQ Application Development: To validate the recording and playback functionality of our application, we used the watch's analysis view to check that stored session data matched real-world activity. For example, we completed sessions composed entirely of heel strikes and verified that the session averages displayed predominantly heel pressure. Similarly, toe-tap sessions showed forefoot-dominated pressure. Real-world testing across team members during runs further confirmed that the app could capture differences between natural heel strikers, midfoot runners, and forefoot runners.

Bluetooth Data Transmission: Throughout development, we incrementally validated Bluetooth communication, beginning with basic LED control from the watch to the device. As real sensor data was integrated, we cross-checked live Bluetooth transmissions by comparing values displayed on the watch against console logs from the ESP32. Once confident in the numerical accuracy, we transitioned to graphical displays (heatmaps and footpath plots) and verified that the compressed 20-byte packets accurately represented step data in real time.

Rechargeable Power System: To test power longevity, we performed multiple extended runs while monitoring the battery voltage before and after use. Results confirmed that the battery maintained sufficient charge for several hours of continuous operation, meeting the expected requirements for typical running sessions.

Overall, STRIDE satisfied all primary technical and functional requirements outlined at the beginning of the project.

4 Other Considered Solutions

Beyond foot-mounted pressure and motion sensing, we considered several alternative methods for tracking running form. One option was placing inertial measurement units on the waist or chest to

capture overall body posture and stride dynamics without requiring any modifications to the shoe. We also explored the idea of using smartwatch-based sensors alone to infer running symmetry and cadence from wrist motion, removing the need for any additional hardware. Another concept was creating a shoe-mounted clip that only houses an IMU, focusing solely on capturing foot trajectory and angle without the added complexity of foot pressure sensors. While these alternatives offer interesting insights, they either lacked the fine-grained foot pressure detail necessary for precise stride analysis, required additional user setup, or sacrificed the level of biomechanical insight we aimed to deliver. Ultimately, we chose a shoe-mounted system with combined pressure and footpath sensing to maximize data accuracy while maintaining convenience and robustness during real-world runs.

5 Suggested Future Directions

While STRIDE meets its initial design goals as a functional prototype, there are several paths to further develop it into a polished, commercial-ready product. One major future direction would be to design and manufacture a custom insole in-house. This would allow STRIDE to fit any shoe size, significantly increase the number of foot pads beyond the current eight for finer granularity in pressure readings, and improve the durability of the system by designing a more flexible and strain-relieved cable. Another improvement would be moving to a lower-power microcontroller combined with custom sleep-wake strategies, which could extend the device's battery life to a full week between charges, making it even more practical for everyday runners. On the software side, integrating STRIDE with Garmin Connect Mobile or a separate cloud database would allow users to track their performance over time, compare different runs, and receive personalized coaching recommendations based on trends. Finally, a deeper integration with the existing Garmin running ecosystem could allow users to overlay detailed foot pressure and stride data from STRIDE directly on a GPS map of their run, providing context-specific biomechanical feedback at different points on their route.

6 Design Considerations and Impacts

The main factors considered while designing our prototype related to the time frame and budget imposed by the course. We also had to consider our limited access to manufacturing resources and commercial-grade sensors.

- Public Health Factors: Our design will hopefully improve the overall fitness and health of the running community and all users who want to live an active lifestyle.
- Safety and Security Factors: When designing our ConnectIQ application, we initially wanted to integrate some of Garmin's running statistics into our application, but we were unable to access them due to Garmin's personal data security protocols.
- Welfare Factors: We wanted to design a product that would improve running efficiency and reduce injury risk, so we considered different designs and UI experiences to maximize that.
- Global Factors: Minimal to no impact on our design.
- Cultural Factors: Minimal to no impact on our design.
- Social Factors: We wanted to design something that would be sleek and fun to wear. This influenced the shape and size of the hardware module and the format of the UI.
- Environmental Factors: We wanted to create a product that was reusable, so we developed a design that used a rechargeable battery. However, the battery has a finite lifespan and will eventually be disposed of.
- Economic Factors: We had a budget imposed for the project that limited the materials and grade of sensors we could purchase. However, for the purpose of our prototype, it wasn't limiting in the components we wanted to buy.
- Ethical Factors: Minimal to no impact on our design.

7 Research and Self-Learning Activities

This project prompted lots of growth for our team, both technically and professionally. We gained exposure with industry standards for project management, tracking our progress in Jira and holding scrums to aid collaboration and coordination. By evaluating design constraints, we improved our understanding of important considerations and trends in the wearables space. Working with the Garmin team and technology, we researched and learned more about BLE communication, as well as development considerations when building for a specific ecosystem. In our work minimizing our form factor, we learned about custom PCB design. Lastly, we furthered our knowledge about embedded systems and electronics in researching all our components, evaluating our needs for processing capabilities and power consumption, and hands-on testing.

8 Conclusion

Over the course of the semester, we worked closely with our clients at Garmin to develop a functional prototype that satisfies our defined MVP and more. STRIDE is a novel platform that enhances the wearable fitness ecosystem through the addition of foot-striking and footpath data. The device is compact, lightweight, and wireless, allowing for easy installation into a running shoe and a comfortable running experience. The platform allows for data visualization in real-time through Garmin ConnectIQ. The user can also view the aggregated data from their runs for analysis of larger trends. Ultimately, STRIDE was an extremely educational experience that resulted in a succesful prototype.

Acknowledgments

We would like to recognize the Garmin team (Daniel Elliot, Mary Stuart Elder, Paul Nichols, and Travis Jones), Duke ECE's Chris Bingham, and our professor Rabih Younes, whose technical expertise and support helped realize this final deliverable.

Received 29 April 2025