ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

StepWise: A Smart Insole System for Gait Analysis and Muscle Rehabilitation

---

Team #24

KERUI XIE (keruix2)
NUO PANG (nuopang2)
ZHICHAO CHEN (zc64)
XIAORUI ZHANG (xz104)

March, 2026

Contents

1 Introduction **1**
- 1.1 Problem Statement . . . 1
- 1.2 Solution Overview and Visual Aid . . . 1
- 1.3 High-Level Requirements List . . . 2

2 Design **4**
- 2.1 Block Diagram . . . 4
- 2.2 Hardware Design . . . 4
  - 2.2.1 Circuits . . . 4
    - 2.2.1.1 Power System . . . 4
    - 2.2.1.2 Pressure Sensor System . . . 5
    - 2.2.1.3 Tilt Sensor System . . . 7
    - 2.2.1.4 Control System . . . 8
    - 2.2.1.5 Wireless Communication System . . . 9
  - 2.2.2 3D-Printed Insole Structure . . . 12
- 2.3 PC Software and Analysis System . . . 13
  - 2.3.1 PC Interface . . . 13
  - 2.3.2 Offline Analysis System . . . 14
- 2.4 Schematics . . . 18
- 2.5 Tolerance Analysis . . . 18

3 Cost **24**

4 Schedule **25**

5 Ethics and Safety **27**
- 5.1 Ethics . . . 27
- 5.2 Safety . . . 27

References **28**

1 Introduction

1.1 Problem Statement

Gait problems are common, but they are often hard to identify early. Many people show signs such as flat-foot-like loading, valgus or varus tendency, or unusual foot strike patterns, yet these issues are not always noticed until they begin to affect comfort, balance, or joint health. In practice, detailed gait assessment usually depends on lab-based systems, which are expensive, bulky, and not easy to access in daily settings. As a result, many users do not have a simple way to collect objective gait data outside a controlled environment.

At the same time, many low-cost wearable devices only provide limited motion or activity information. They may count steps or track general movement, but they usually do not capture enough plantar pressure and foot posture data to support basic gait screening. They also often stop at data collection and do not connect the results to simple rehabilitation guidance.

Our project addresses this gap by focusing on a low-cost, portable smart insole system for short, standardized walking tests. Instead of trying to replace a full clinical gait laboratory, StepWise is designed as a screening and rehabilitation-support tool. The goal is to collect useful pressure and tilt data from a wearable right-foot insole, send the data to a host PC, and generate clear post-session feedback that helps users better understand their gait patterns.

1.2 Solution Overview and Visual Aid

StepWise is a smart insole system designed for gait analysis and basic muscle rehabilitation support. The current prototype uses a right-foot insole with five pressure sensors and one tilt sensor. These sensors measure plantar loading and foot orientation during a short walking session. An ESP32-S3-based control board samples the sensor data, organizes the readings, and sends them to a host PC through WiFi. After the test session ends, the PC software stores the data, displays the recorded curves, and performs offline analysis.

The system is intended to support a simple and practical workflow. A user wears the insole and completes a short walking test. During the test, the insole collects synchronized pressure and tilt data. The data are transmitted to the PC in real time for recording, but the main analysis is done after the session. This design keeps the system easier to validate and avoids distracting the user during walking. Based on the recorded data, the software can show pressure trends, summarize basic gait features, screen for a few simple abnormal patterns, and provide exercise suggestions linked to the detected results.

Figure 1 shows the physical layout of the StepWise system. It illustrates the smart insole, the sensor placement, the control electronics, and the connection to the external computing device used for offline analysis.

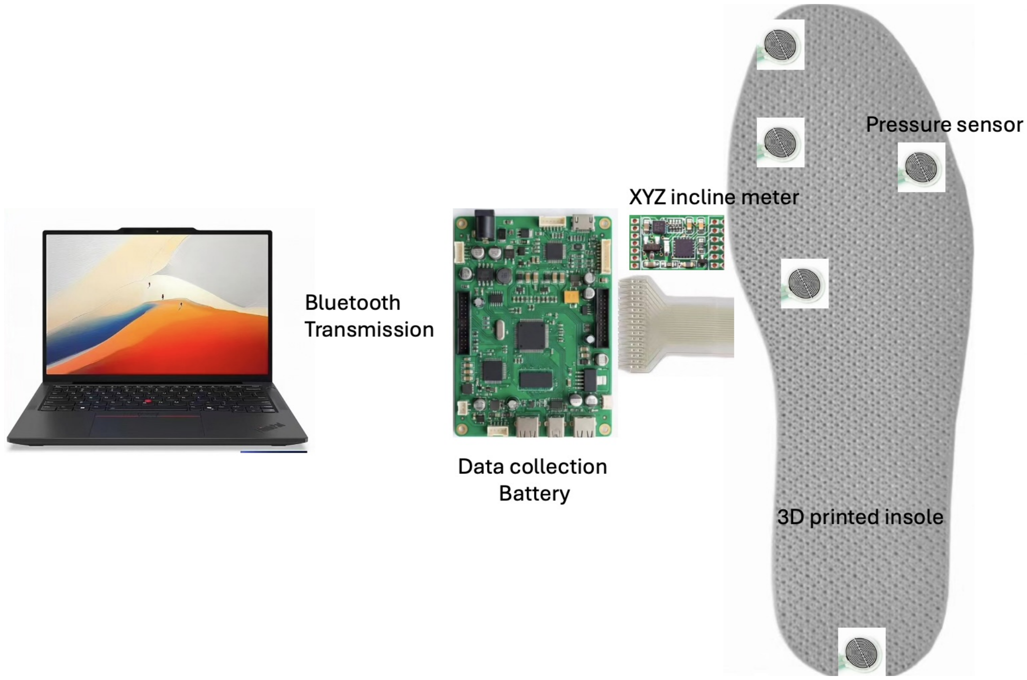

Figure 1: Physical overview of the StepWise system, showing the smart insole structure, embedded sensors, control electronics, and host-side analysis workflow.

Compared with large gait lab systems, StepWise aims to provide a simpler and more affordable way to collect useful gait data. Compared with basic consumer wearables, it focuses more directly on foot-ground interaction and post-session interpretation. The system is not intended to serve as a clinical diagnosis device. Instead, it is designed as a portable screening and rehabilitation-support platform that helps users and developers observe gait-related patterns in a more accessible way.

1.3 High-Level Requirements List

The following high-level requirements define the main goals of the StepWise prototype:

1. **End-to-end data integrity:** During a standardized walking session, the system shall capture and transmit synchronized data from all five pressure channels and the tilt sensor to the host PC, with packet loss no greater than 5% per session and no channel-level dropout longer than 1 second.
2. **Sensor accuracy and repeatability:** The pressure sensing subsystem shall be validated using static and repeated loading tests, and the tilt sensing subsystem shall be validated using fixed-angle and slow-tilt tests. The target performance is pressure error $\leq 5\%$ under static loading, pressure error $\leq 8\%$ under repeated loading, angular error $\leq 2^\circ$ under static angle testing, and angular error $\leq 3^\circ$ under slow-tilt testing.
3. **Reliable host-side reception and offline analysis:** The PC software shall success-

fully receive, parse, store, and analyze session data from the insole system. Using labeled simulated datasets, the software shall achieve at least 95% successful packet or session parsing and at least 90% classification accuracy for the selected gait-pattern screening tasks.

2 Design

2.1 Block Diagram

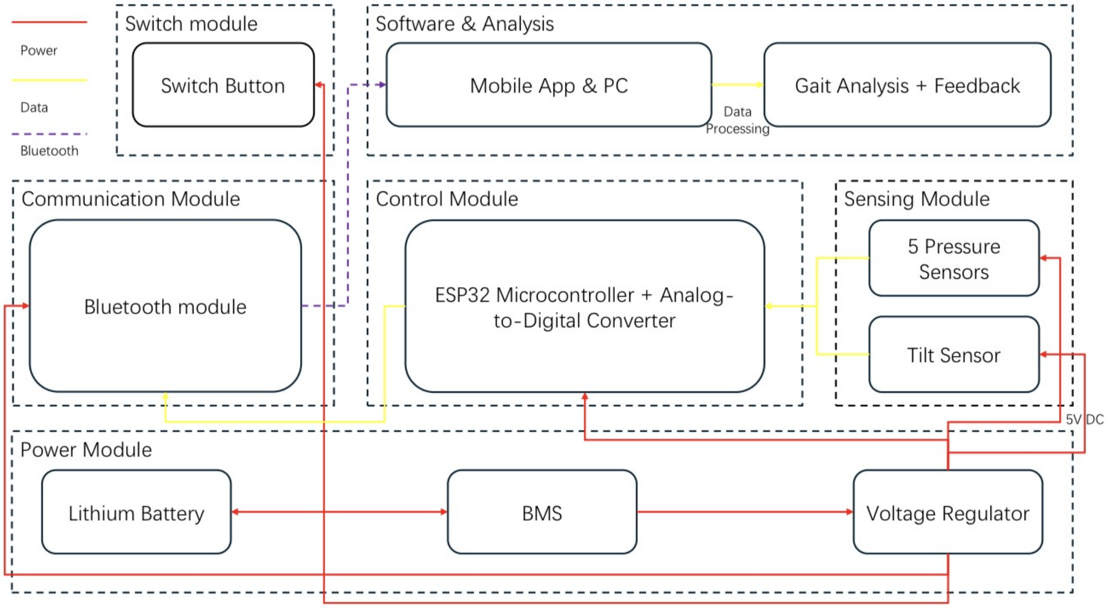

Figure 2: System block diagram illustration.

2.2 Hardware Design

2.2.1 Circuits

2.2.1.1 Power System Description.

The Power Subsystem is responsible for energy management, storage, and voltage regulation for the StepWise smart insole. To support a fully untethered wearable, the power flow operates in two stages. First, a 5V USB Type-C input is routed through a TP4056 linear charging IC to safely charge a 3.7V, 200mAh Lithium-Polymer (Li-Po) battery. Second, the battery voltage (which fluctuates between 3.0V and 4.2V) is routed into an SY8088AAC Synchronous Step-Down (Buck) Converter.

Interaction with other subsystems: The SY8088AAC provides a highly stable 3.3V DC power rail to the Control Subsystem (ESP32) and the Sensing Subsystem (MPU6050 and the FSR array). Because the FSRs act as voltage dividers, the absolute stability of this 3.3V rail is critical; any voltage fluctuation will directly result in inaccurate analog-to-digital (ADC) conversions and corrupted diagnostic data.

Requirements.

To ensure reliable operation of the microcontroller and sensors, the power subsystem must adhere to the following quantitative requirements:

Req 1.1 (Voltage Output): The buck converter must supply a regulated output voltage of 3.30V±5% (an acceptable range of 3.135V to 3.465V) across all battery charge states.

Req 1.2 (Current Delivery): The subsystem must be capable of delivering transient current spikes of at least 500mA without experiencing a voltage droop below the minimum threshold. This is required to support the instantaneous power draw of the ESP32's WiFi radio during data transmission to the host PC.

Req 1.3 (Voltage Ripple): The output voltage ripple must not exceed 50mV_{pp} under a nominal 100mA load to prevent high-frequency switching noise from interfering with the I^2C bus and the analog sensor readings.

Verification.

Verification for Req 1.1 & 1.2 (Voltage & Load Test):

Procedure: Connect a programmable DC electronic load to the 3.3V output rail. Sweep the current draw from 10mA up to 500mA. Monitor the output voltage using a calibrated Digital Multimeter (DMM).

Success Criterion: The measured voltage never drops below 3.135V or exceeds 3.465V at any current level within the sweep.

Verification for Req 1.3 (Ripple Test):

Procedure: Connect an oscilloscope to the 3.3V output rail using an AC-coupled probe. Limit the oscilloscope bandwidth to 20MHz to reject ambient RF noise. Configure the ESP32 to actively transmit WiFi packets to simulate a realistic load.

Success Criterion: The measured peak-to-peak voltage (V_{pp}) on the oscilloscope remains $\leq 50\text{mV}_{pp}$.

Supporting Material.

2.2.1.2 Pressure Sensor System **Description.**

The Pressure Sensor Subsystem is responsible for capturing the plantar pressure distribution of the user. It consists of five Force-Sensitive Resistors (FSRs) interfaced with the ESP32 via headers U5 through U9. Each FSR is configured as a high-side pull-up voltage divider, paired with a $10\text{k}\Omega$ fixed resistor connected to the 3.3V rail. The analog voltage at the junction is read by the ESP32's internal 12-bit Analog-to-Digital Converter (ADC).

Requirements.

Req 2.1 (Dynamic Range): The subsystem must output an analog voltage between 0.3V (maximum pressure) and 3.3V (no pressure).

Req 2.2 (ADC Accuracy): The derived voltage reading must have an absolute electrical error of no more than ±0.15V at a nominal 1.65V midpoint, accounting for resistor tolerances and system power ripple.

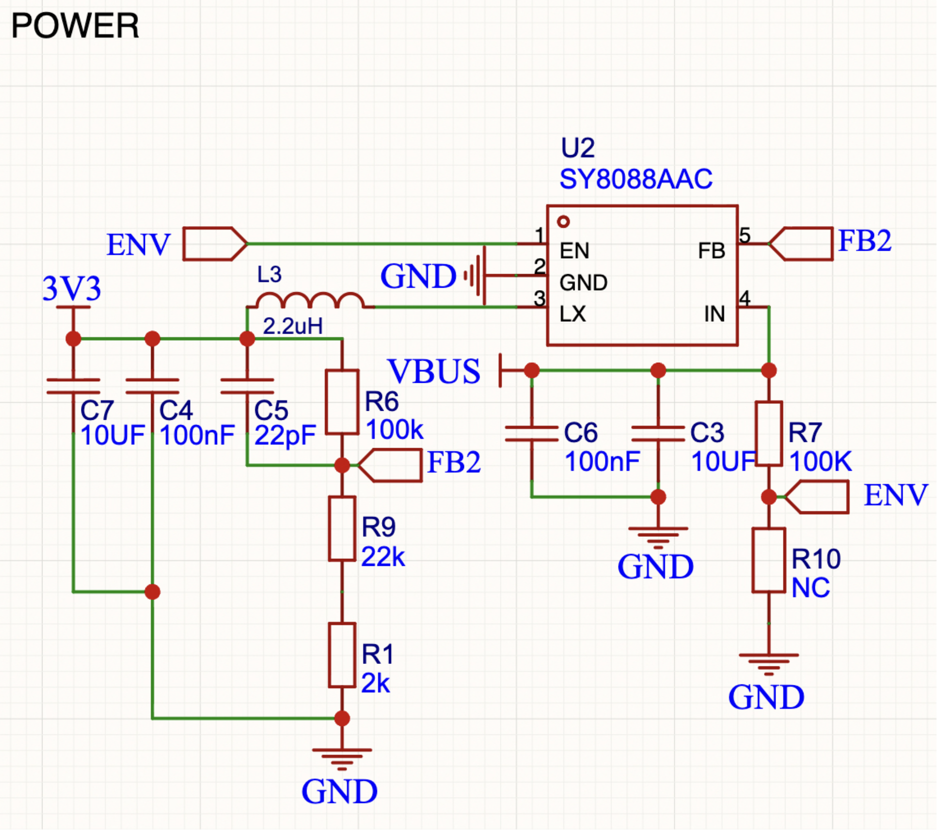

Figure 3: Power subsystem schematic.

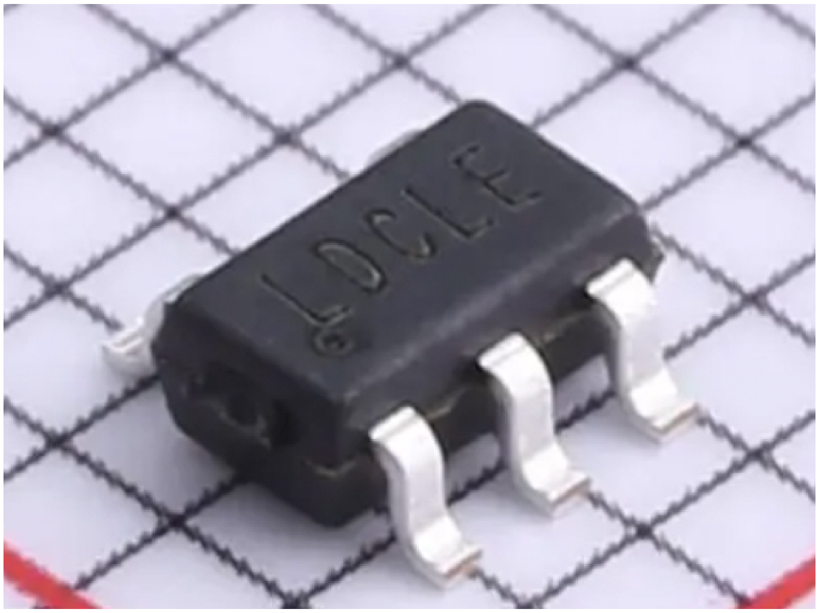

Figure 4: SY8088AAC buck converter implementation details.

Verification.

Verification for Req 2.1 (Dynamic Range):
Procedure: Apply controlled loads to each FSR channel from no-load to maximum expected foot pressure while recording ADC-derived voltages.
Success Criterion: Each channel reports voltages spanning approximately 0.3V (high load) to 3.3V (no load), with monotonic response to increasing force.

Verification for Req 2.2 (ADC Accuracy):
Procedure: At a nominal midpoint condition (target 1.65V), compare ESP32 ADC-derived voltage values against a calibrated DMM at the same node across all sensor channels, while observing ripple/noise effects.
Success Criterion: Absolute electrical error remains within ±0.15V for all channels at the midpoint operating point.

Supporting Material.

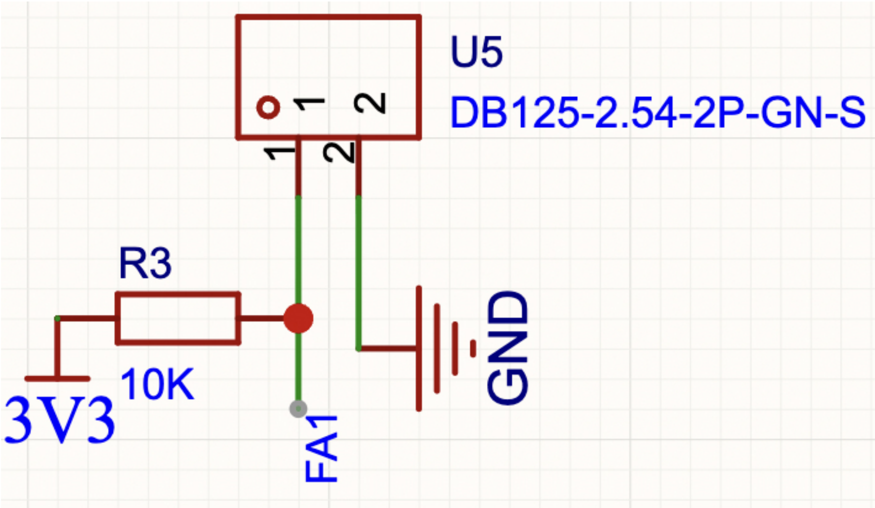

Figure 5: Pressure sensor subsystem schematic and channel mapping.

2.2.1.3 Tilt Sensor System **Description.**

This subsystem measures spatial orientation to detect gait abnormalities such as foot valgus or varus. It utilizes the MPU6050, a 6-axis IMU (Inertial Measurement Unit), communicating with the Control Subsystem via the I^2C bus at 400 kHz.

Requirements.

Req 3.1 (Angular Accuracy): The subsystem must calculate the static inversion/eversion angle of the foot with an uncalibrated accuracy of ±2°.

Req 3.2 (Sampling Rate): The IMU must successfully output accelerometer and gyroscope data at a minimum frequency of 50Hz without I^2C bus collisions.

Verification.

Procedure: Mount the PCB to a machinist's digital tilt-table. Incline the table to exactly 15° on the X-axis.

Success Criterion: The raw accelerometer data processed through the MCU's arctangent function must output an angle between 13° and 17°.

Supporting Material.

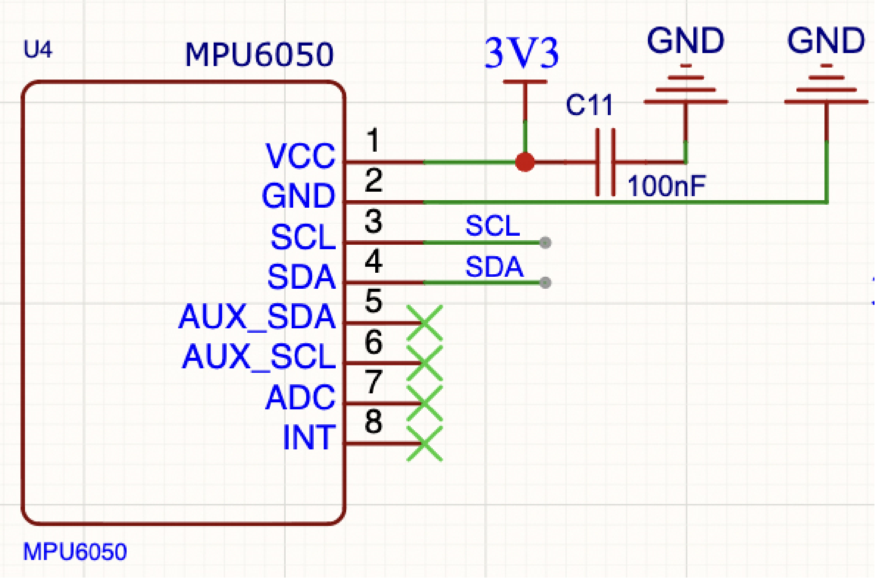

Figure 6: Tilt sensor subsystem integration.

2.2.1.4 Control System Description.

The Control Subsystem utilizes the ESP32 MCU to orchestrate the entire insole. It sequences ADC reads for the five FSRs, polls the MPU6050 over I^2C, buffers the data, and prepares it for wireless transmission. It also includes a CH340C bridge for USB debugging and auto-programming.

Requirements.

Req 4.1 (Timing/Throughput): The MCU must successfully poll all six sensors (five analog, one I^2C) within a 20ms window to guarantee a minimum 50Hz system-wide sampling rate.

Req 4.2 (Logic Levels): All communication pins must operate strictly within 3.3V CMOS logic thresholds.

Verification.

Procedure: Write firmware to toggle an unused GPIO pin HIGH at the start of the sensor-reading loop, and LOW when the loop finishes. Attach an oscilloscope to the GPIO pin.

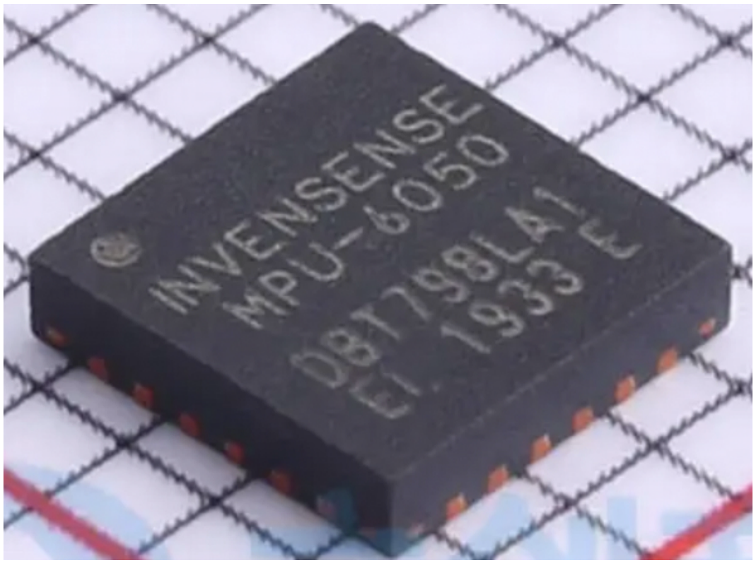

Figure 7: MPU6050 module and interface details.

Success Criterion: The pulse width measured on the oscilloscope must be consistently less than 20ms.

Supporting Material.

2.2.1.5 Wireless Communication System Description.

Integrated directly into the ESP32-S3 SoC, this subsystem uses WiFi to transmit buffered gait data from the insole to a nearby host PC for session recording and offline analysis.

Requirements.

Req 5.1 (Data Throughput): The subsystem must sustain a minimum data transmission rate of 15 kbps to prevent data-buffering bottlenecks during a continuous walking test.

Req 5.2 (Range): Maintain a stable connection (packet loss $< 1\%$) over a 3-meter line-of-sight (LoS) distance between the insole and the host PC or its WiFi access point.

Verification.

Procedure: Initiate a WiFi diagnostic stream to the host PC. Place the receiving PC or access point effectively 3 meters away from the insole and run a 60-second test.

Success Criterion: The PC-side log must show that more than 99% of the 3,000 expected packets ($50\text{Hz} \times 60\text{s}$) were successfully received and validated via checksum.

Supporting Material.

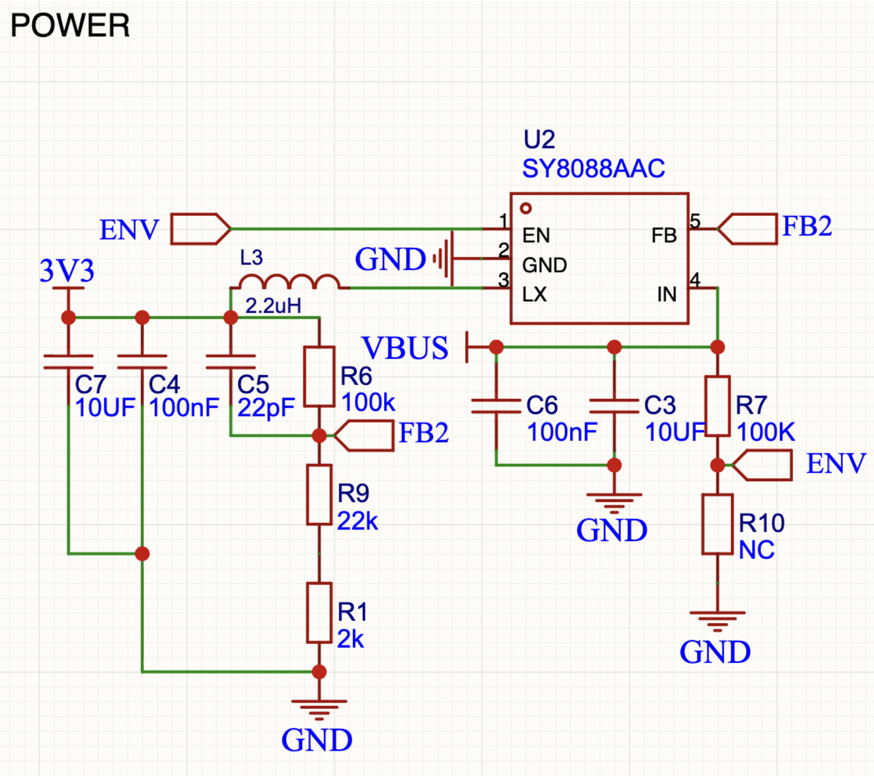

Figure 8: Control subsystem architecture and signal flow.

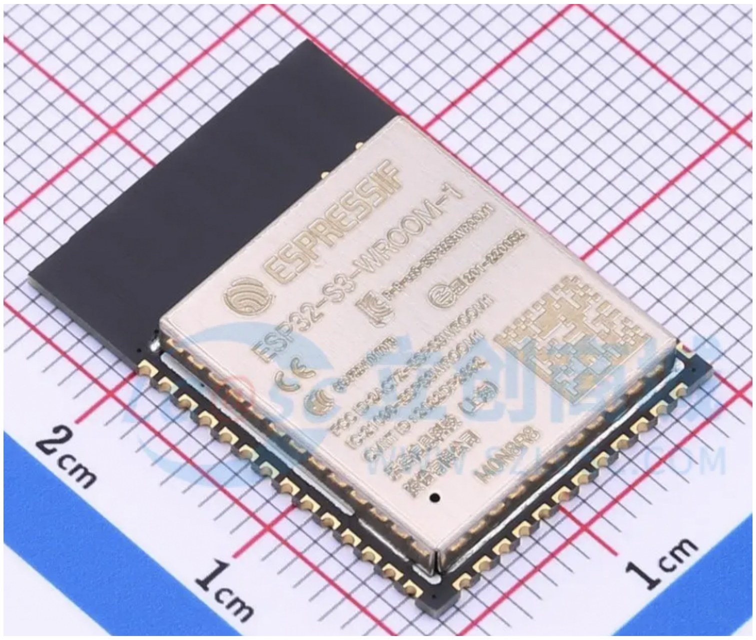

Figure 9: ESP32 module and key interface connections.

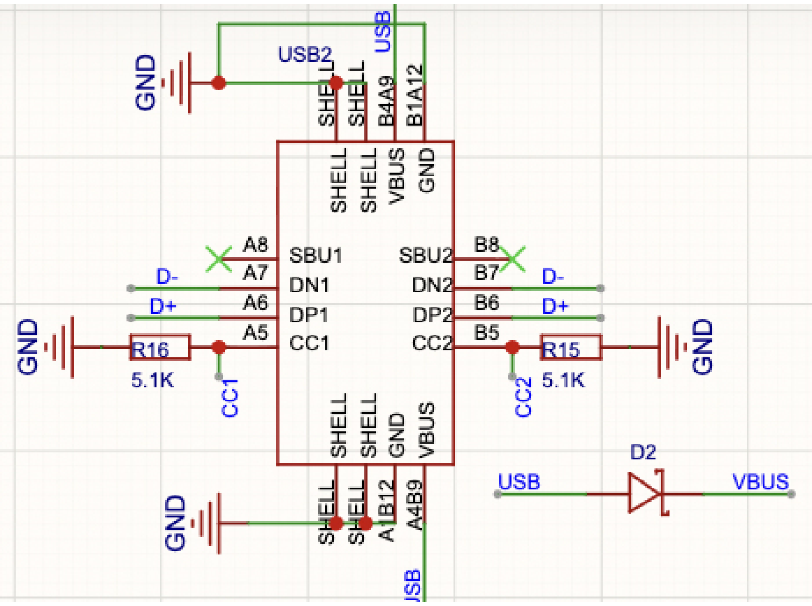

Figure 10: Wireless communication and interface support hardware.

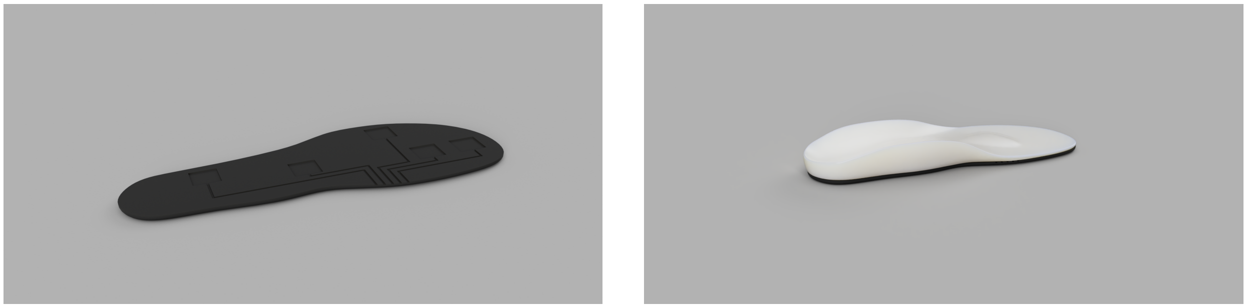

Figure 11: Two-layer insole structure and layer layout.

2.2.2 3D-Printed Insole Structure

The 3D-printed insole structure serves as the mechanical host platform of the StepWise system. Rather than acting as a passive shoe insert, it provides a controlled structural foundation for sensor placement, load transmission, electronics integration, and user-safe wearable operation. In the current prototype, the insole supports a five-point plantar pressure sensing array, an onboard orientation sensor, distributed interconnects, and a custom control board architecture built around the ESP32-S3 platform. This arrangement is consistent with the physical diagram and hardware integration plan, in which the insole body carries the sensing elements while the control, power, and communication hardware are integrated as a compact wearable subsystem.

Material Considerations Among PLA, ABS, and PETG, PETG is selected as the primary material for the StepWise 3D-printed insole. PLA offers excellent printability and dimensional accuracy, but its brittleness and poor fatigue resistance make it unsuitable for repeated walking-related loading. In addition, its rough surface is not comfortable for long-term walking use. ABS provides higher toughness than PLA, but its tendency to warp during printing reduces geometric consistency and increases fabrication risk for a thin wearable structure. Also, in some typical usage environments (e.g., poorly ventilated environments), ABS may release volatile organic compounds that can be harmful to human skin. PETG provides the best overall balance: it is tougher than PLA, easier to print reliably than ABS, and sufficiently compliant to reduce brittle failure risk while maintaining structural support for sensor placement and electronics integration. Its resistance to chemical corrosion and smoother surface provide the foot with a safer and more comfortable environment.

Structure and Layout Structurally, the insole is designed as a functionally partitioned system rather than a uniform printed part. The sensing region provides stable plantar load transfer to the pressure sensors located at key foot-contact areas. The two-layer structure holds the sensors tightly in grooves in the bottom layer to reduce measurement error caused by sensor movement during testing. The electronics region houses the custom PCB, power-related components, and local interconnects in a mechanically protected zone. Between these regions, the routing region provides wiring or flexible interconnection paths that accommodate deformation without overstressing solder joints or sensor terminals. This partitioned architecture improves integration quality by reducing mechanical interference between high-strain walking regions and relatively rigid electronic hardware. In addition, to improve wearable comfort and measurement repeatability, the insole is shaped to generally follow the arch geometry of the human foot rather than using a uniform flat structure. The midfoot region includes a mild arch-shaped elevation to better support the natural foot arch, while the rearfoot region incorporates a shallow heel-containment profile to improve foot positioning and reduce lateral motion during walking tests. This ergonomic geometry helps distribute contact more naturally across the foot, improves structural fit inside the shoe, and reduces the likelihood that the prototype introduces discomfort or measurement artifacts during use.

Requirements	Verification
1. The 3D-printed insole shall maintain a low-profile contact surface with no exposed hard edges or local protrusions that cause discomfort during walking trials. 2. The insole structure shall maintain sensor position and mechanical support such that no pressure sensor or routed connection shifts by more than 2 mm after repeated loading tests. 3. The insole shall preserve structural integrity under repeated compression and bending, with no visible crack, exposed conductor, or hardware failure.	1. Conduct a 5-minute walking trial with at least three volunteers. The structure passes if no user reports sharp local discomfort and no exposed component is felt through the contact surface. 2. Measure initial sensor/reference positions, then repeat 300-400 loading or walking-equivalent cycles. Re-measure the positions and inspect fixture integrity. The structure passes if displacement is within 2 mm. 3. Perform repeated compression and bending tests followed by structural inspection and functional readout. The structure passes if no crack is observed, no conductor is exposed, and all sensing channels remain operational.

Table 1: Requirements and verification for the 3D-printed insole structure.

2.3 PC Software and Analysis System

2.3.1 PC Interface

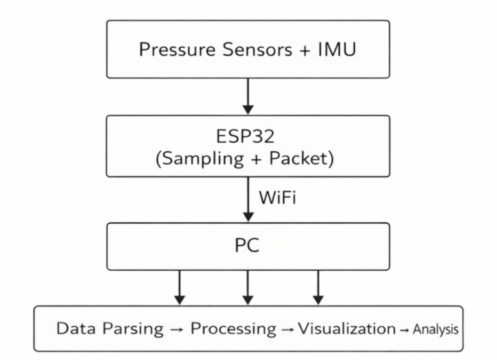

Figure 12: Overview of the PC interface and host-side data flow.

The PC software serves as the host-side interface for the StepWise system. Its main role is to receive sensor data transmitted from the ESP32-S3 over WiFi, organize the incoming packets into a session-based data structure, and provide real-time visibility into the status of the walking test.

The interface is designed around short, standardized walking sessions rather than continuous long-term monitoring. A user or operator starts a session from the PC side, the insole system streams synchronized pressure and IMU data during the test, and the PC software stores the received frames until the session ends. After the session is complete, the data are saved locally for offline processing and result generation.

Each received data frame will contain a timestamp, five pressure-sensor values, and IMU-related data used for tilt estimation. The PC interface is responsible for parsing these fields, checking frame completeness, and handling communication errors such as missing packets or connection interruptions. If the received data stream becomes unstable, the software will mark the session as invalid instead of forwarding incomplete data to the analysis stage.

In addition to data reception, the PC interface provides basic visualization and control functions. These include displaying connection status, sample count, and real-time signal curves for the pressure channels and orientation signals. The software also supports simple session-level commands such as start, stop, save, and load. This design keeps the host-side workflow easy to validate and suitable for debugging during subsystem integration.

2.3.2 Offline Analysis System

The analysis system performs offline processing of the session data after the walking test has ended. This design choice reduces the complexity of real-time in-shoe feedback and makes the software easier to validate using recorded and simulated datasets.

The analysis workflow contains four stages: preprocessing, feature extraction, rule-based screening, and result generation. In the preprocessing stage, the software checks data completeness, orders frames by timestamp, and removes clearly invalid or incomplete records. If needed, simple filtering or smoothing may be applied to reduce noise in the recorded sensor signals.

After preprocessing, the system extracts a small set of interpretable gait-related features from the pressure and IMU data. These features include per-channel peak and mean pressure, relative loading in the heel, midfoot, and forefoot regions, and summary statistics of the orientation signals. These features are selected because they are simple to compute, easy to visualize, and suitable for validating the software pipeline in a senior design setting.

The current analysis strategy is rule-based rather than model-heavy. Instead of attempting full clinical diagnosis, the software screens for a limited number of simple patterns such as normal-like loading, flat-foot-like loading, and loading asymmetry suggestive of valgus-like or varus-like behavior. The rules are derived from threshold comparisons and relative loading relationships between sensor regions. This keeps the output interpretable and aligned with the project scope.

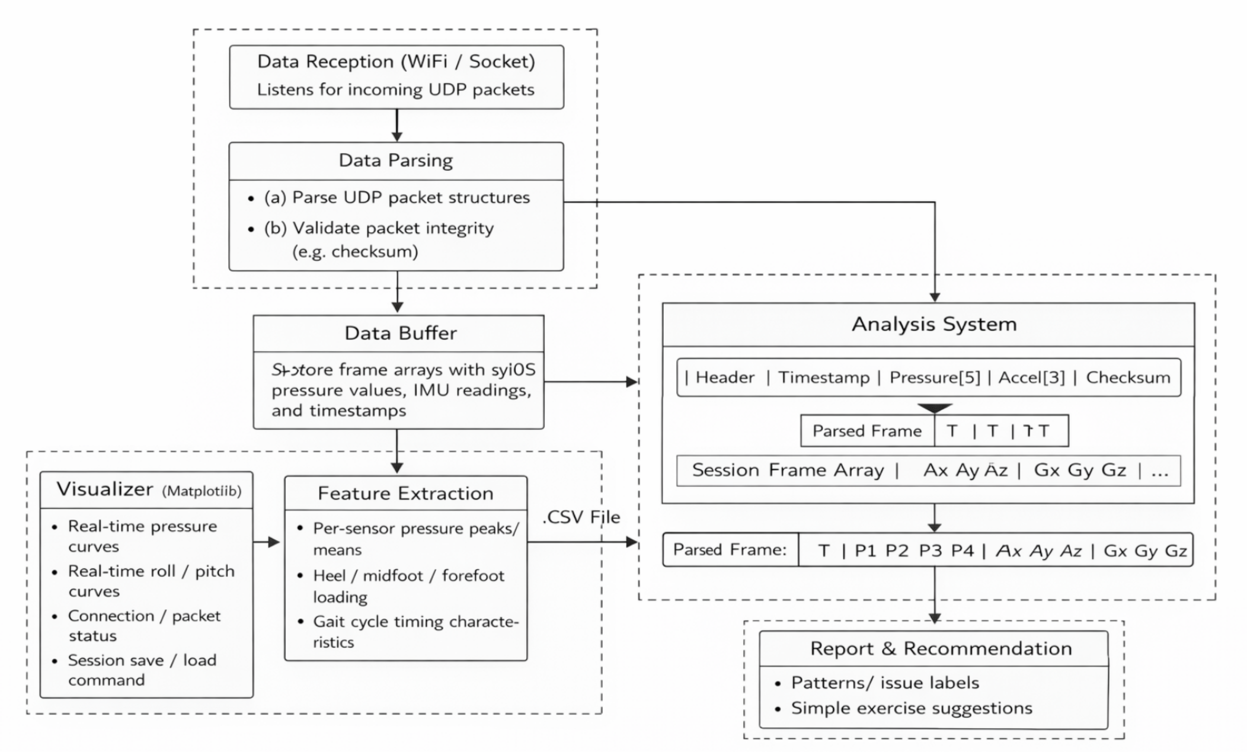

Figure 13: Detailed architecture of the PC software system for data processing and gait analysis

Finally, the software generates a compact result summary for each session. The output includes synchronized signal plots, extracted feature values, a simple pattern label, and short exercise suggestions associated with the detected result. This allows the PC software to act as a practical screening and rehabilitation-support tool without over-claiming clinical capability.

PC Software and Analysis Subsystem Requirements

The PC Software and Analysis subsystem is responsible for receiving transmitted session data, storing session records, visualizing sensor signals, and performing basic offline gait screening.

PC Software and Analysis Subsystem Verification

The subsystem will be verified through functional tests using simulated datasets in the early stage and real transmitted data after hardware integration. Table 2 summarizes the requirement–verification mapping.

Requirements	Verification
1. The PC software shall correctly receive and parse transmitted session data from the ESP32-side subsystem. For each valid session, at least 95% of transmitted frames shall be parsed successfully without program interruption or data-format failure. 2. The PC software shall save each completed walking session as a structured local file containing timestamped pressure and IMU data. For successfully parsed frames, no required field shall be missing in the saved record. 3. The PC software shall display all five pressure channels and selected IMU signals within 3 seconds after a completed session is loaded or ends. 4. The analysis system shall perform basic offline screening using interpretable rule-based logic and achieve at least 90% accuracy on selected project-defined screening tasks (e.g., normal-like loading, flat-foot-like loading, and loading-asymmetry-like patterns). 5. If the received data stream is incomplete, corrupted, or interrupted, the software shall mark the session as invalid rather than outputting a potentially misleading analysis result.	1. Generate and transmit valid test packets and session streams, while injecting malformed/incomplete packets to test parser robustness. Pass if at least 95% of valid frames are parsed successfully and malformed packets do not crash the program. 2. Create and save mock walking sessions, then reload and compare with original records for field completeness, ordering, and session-length consistency. Pass if structure and field count match, with no missing required field in successfully parsed frames. 3. Load a completed session file and check whether all required pressure and IMU curves are rendered with correct time axes and labels. Pass if all required plots are displayed within 3 seconds and remain readable. 4. Process a labeled validation set of simulated/recorded sessions and compare predicted labels with reference labels. Pass if screening accuracy is at least 90% on the selected pattern classes. 5. Provide interrupted, truncated, or intentionally corrupted session streams and inspect system behavior. Pass if invalid or incomplete sessions are correctly flagged and no misleading final screening result is generated.

Table 2: Requirements and verification for the PC software and analysis subsystem.

2.4 Schematics

2.5 Tolerance Analysis

3D-Printing Structure Tolerance

A critical question for the 3D-printed insole structure is whether fabrication and assembly tolerances can be controlled tightly enough to preserve meaningful plantar-pressure measurements. Since the StepWise system uses five pressure sensors to represent key plantar loading regions, excessive geometric or placement error may shift a sensor away from its intended anatomical location and reduce the interpretability of the measured gait pattern. Therefore, the tolerance analysis for this subsystem focuses on two issues: sensor placement deviation and arch-geometry deviation.

Sensor Placement Tolerance Let the nominal center of a pressure sensor be (x_0, y_0) and the actual installed center be (x, y). The placement error is defined as

$$e = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{1}$$

where e is the Euclidean distance between the designed sensor position and the actual installed position.

To ensure that a sensor remains within its intended plantar region, the placement error must be smaller than the minimum available distance from the nominal sensor center to the boundary of the target anatomical region. This requirement can be written as

$$e < d_{\text{min}} \tag{2}$$

where d_{min} is the minimum geometric margin between the nominal sensor center and the nearest boundary of the intended sensing region.

If the pressure sensor has a non-negligible effective sensing radius r_s, a more conservative condition is

$$e + r_s < d_{\text{region}} \tag{3}$$

where d_{region} is the distance from the nominal sensor center to the boundary separating adjacent functional plantar regions. Equation 3 ensures that the sensing footprint remains inside the intended region instead of overlapping significantly with a neighboring region.

This analysis shows that the heel and forefoot sensors can generally tolerate slightly larger placement deviations because their target areas are relatively larger, while the midfoot or arch-region sensor requires tighter positional control due to the smaller anatomical target area and its higher sensitivity to local geometry changes. Based on this reasoning, a preliminary design goal of approximately $\pm 2\,\mathrm{mm}$ placement tolerance is reasonable for most sensors, while tighter control may be desirable in the arch region.

Arch-Height Tolerance In addition to sensor placement, the geometric accuracy of the ergonomic insole profile must also be controlled. The insole is not designed as a completely flat plate; instead, it includes a mild arch-support feature and a heel-containment profile so that the structure better follows the natural shape of the human foot. However, if the printed arch feature deviates excessively from its intended shape, the insole may begin to act as a corrective orthotic and artificially alter the plantar loading pattern that the system is intended to measure.

Let the designed arch elevation be h_0 and the actual printed arch elevation be h. The arch-height deviation is defined as

$$\Delta h = |h - h_0| \tag{4}$$

where Δh is the absolute geometric deviation of the printed arch feature from its intended design value.

To maintain the role of the insole as a sensing platform rather than a corrective support device, the arch-height deviation should remain below a small tolerance threshold:

$$\Delta h \leq h_{\text{tol}} \tag{5}$$

where h_{tol} is the maximum allowable arch-height deviation. A preliminary engineering target of approximately $1\,\mathrm{mm}$ is reasonable for prototype evaluation, since this limit is small enough to preserve ergonomic fit while reducing the risk of introducing major measurement artifacts during walking trials.

Circuits and Measurement Tolerance

Power Prove that the 3.3V output remains within the required $\pm 5\%$ tolerance (3.135V to 3.465V) despite manufacturing variances of physical components.

Mathematical Analysis:
The output voltage (V_{out}) of the SY8088AAC step-down converter is determined by its internal reference voltage (V_{ref}) and the external feedback resistor divider network (R_{top} and R_{bot}), defined by:

$$V_{\text{out}} = V_{\text{ref}} \times \left(1 + \frac{R_{\text{top}}}{R_{\text{bot}}}\right) \tag{6}$$

According to the SY8088 datasheet, the typical internal reference voltage is $V_{\text{ref}} = 0.60$V with an internal variance of $\pm 2\%$. Therefore:

- $V_{\text{ref(max)}} = 0.612$V

- $V_{\text{ref(min)}} = 0.588\text{V}$

To target a nominal 3.3V output, we select 1% tolerance external resistors: $R_{\text{top}} = 450\text{k}\Omega$ and $R_{\text{bot}} = 100\text{k}\Omega$. Applying the 1% tolerance yields the worst-case resistor values:

- $R_{\text{top(max)}} = 454.5\text{k}\Omega \quad | \quad R_{\text{top(min)}} = 445.5\text{k}\Omega$
- $R_{\text{bot(max)}} = 101.0\text{k}\Omega \quad | \quad R_{\text{bot(min)}} = 99.0\text{k}\Omega$

Worst-Case Maximum Voltage:
This occurs when V_{ref} is maximum, R_{top} is maximum, and R_{bot} is minimum:

$$V_{\text{out(max)}} = 0.612\text{V} \times \left(1 + \frac{454.5\text{k}\Omega}{99.0\text{k}\Omega}\right)$$
$$V_{\text{out(max)}} = 0.612\text{V} \times (1 + 4.591)$$
$$V_{\text{out(max)}} = 0.612\text{V} \times 5.591 = \mathbf{3.421}\text{V}$$

Worst-Case Minimum Voltage:
This occurs when V_{ref} is minimum, R_{top} is minimum, and R_{bot} is maximum:

$$V_{\text{out(min)}} = 0.588\text{V} \times \left(1 + \frac{445.5\text{k}\Omega}{101.0\text{k}\Omega}\right)$$
$$V_{\text{out(min)}} = 0.588\text{V} \times (1 + 4.411)$$
$$V_{\text{out(min)}} = 0.588\text{V} \times 5.411 = \mathbf{3.181}\text{V}$$

Through worst-case mathematical simulation, the maximum output boun**Conclusion of Feasibility:**
dary is 3.421V and the minimum boundary is 3.181V. Because this calculated worst-case range lies inside the specified requirement of 3.135V to 3.465V, the design is feasible. The power subsystem can reliably support accurate ADC sensing even when components drift to tolerance extremes.

Pressure Sensor **Objective:** Prove that electrical component variances do not cause the analog voltage (V_{out}) to exceed the $\pm 0.15\text{V}$ error margin at a nominal mid-pressure state.

Mathematical Analysis:
The voltage at the ADC pin (V_{out}) is defined by the voltage divider equation, where R_{fixed} is the pull-up resistor and R_{FSR} connects to ground:

$$V_{\text{out}} = V_{\text{cc}} \times \left(\frac{R_{\text{FSR}}}{R_{\text{fixed}} + R_{\text{FSR}}}\right) \tag{7}$$

Assume a mid-step pressure where the ideal FSR resistance drops to $10\text{k}\Omega$.

- Nominal $V_{\text{cc}} = 3.3\text{V}$ (with $\pm 5\%$ tolerance from the power subsystem: 3.135V to 3.465V)

- $R_{\text{fixed}} = 10\text{k}\Omega \pm 1\%$ ($9.9\text{k}\Omega$ to $10.1\text{k}\Omega$)
- $R_{\text{FSR}} = 10\text{k}\Omega \pm 5\%$ part-to-part electrical repeatability ($9.5\text{k}\Omega$ to $10.5\text{k}\Omega$)

Ideal Nominal Voltage:

$$V_{\text{out(ideal)}} = 3.3 \times \left(\frac{10}{10+10}\right) = \mathbf{1.650}\text{V}$$

Worst-Case Maximum Voltage: (Max V_{cc}, Max R_{FSR}, Min R_{fixed})

$$V_{\text{out(max)}} = 3.465 \times \left(\frac{10.5}{9.9+10.5}\right) = 3.465 \times 0.5147 = \mathbf{1.783}\text{V}$$

Worst-Case Minimum Voltage: (Min V_{cc}, Min R_{FSR}, Max R_{fixed})

$$V_{\text{out(min)}} = 3.135 \times \left(\frac{9.5}{10.1+9.5}\right) = 3.135 \times 0.4846 = \mathbf{1.519}\text{V}$$

Conclusion of Feasibility:
The maximum deviation from the nominal 1.650V is $+0.133\text{V}$ and -0.131V. Because both worst-case scenarios fall inside the required $\pm 0.15\text{V}$ error boundary, the voltage divider design is feasible and electrically reliable for the ADC.

Tilt Sensor **Objective:** Verify that inherent micro-electromechanical (MEMS) sensor tolerances do not exceed the $\pm 2^\circ$ static angle requirement.

Mathematical Analysis:
Static tilt angle (θ) is calculated using the gravity vector on the accelerometer's X and Z axes:

$$\theta = \arcsin\left(\frac{A_x}{1g}\right) \tag{8}$$

According to the MPU6050 datasheet, the maximum zero-g initial calibration tolerance (offset error) on the accelerometer is $\pm 20\text{mg}$ ($0.02g$) at room temperature.

Worst-Case Uncalibrated Angle Error: If the sensor is perfectly flat (0° actual tilt), the maximum erroneous acceleration read due to offset is $A_x = 0.02g$.

$$\theta_{\text{error}} = \arcsin(0.02)$$
$$\theta_{\text{error}} = 1.146^\circ$$

Conclusion of Feasibility:
The worst-case inherent MEMS offset produces an angular error of 1.146°. Because $1.146^\circ < 2.0^\circ$, the MPU6050 meets the uncalibrated angular accuracy requirement for detecting basic foot valgus/varus.

Control Objective: Verify that the control subsystem still satisfies the 50Hz sampling requirement after accounting for practical timing uncertainty, interface overhead, and firmware margin.

Mathematical Analysis:
For tolerance analysis, the control loop is modeled using a conservative worst-case timing budget instead of ideal conversion times alone. Let

$$T_{\text{loop,worst}} = 5T_{\text{ADC,worst}} + T_{\text{I2C,worst}} + T_{\text{pack}} \tag{9}$$

where $T_{\text{ADC,worst}}$ is the worst-case per-channel analog read time including mux settling and firmware overhead, $T_{\text{I2C,worst}}$ is the worst-case IMU read time including protocol overhead, and T_{pack} is the software time used for buffering, timestamping, and packet assembly.

A conservative timing budget is chosen as follows:

- **ADC budget:** use $T_{\text{ADC,worst}} = 50\mu$s per channel. This is five times larger than the nominal 10μs conversion time and covers channel switching, settling, and firmware-call overhead.
- **I^{2}C budget:** use $T_{\text{I2C,worst}} = 250\mu$s. This covers a 6-byte MPU6050 read together with addressing, ACK bits, start/restart conditions, and moderate clock or firmware latency.
- **Packaging budget:** use $T_{\text{pack}} = 500\mu$s to cover local buffering, timestamp insertion, and packet formatting before WiFi transmission.

Worst-Case Loop Time:

$$\begin{aligned}
T_{\text{loop,worst}} &= 5(50\mu\text{s}) + 250\mu\text{s} + 500\mu\text{s} \\
T_{\text{loop,worst}} &= 250\mu\text{s} + 250\mu\text{s} + 500\mu\text{s} \\
T_{\text{loop,worst}} &= \mathbf{1{,}000}\boldsymbol{\mu}\text{s}
\end{aligned}$$

The 50Hz system requirement corresponds to a frame period of

$$T_{\text{req}} = \frac{1}{50} = 20\text{ms} = 20{,}000\mu\text{s} \tag{10}$$

Therefore, the remaining timing margin is

$$\begin{aligned}
T_{\text{margin}} &= T_{\text{req}} - T_{\text{loop,worst}} \\
T_{\text{margin}} &= 20{,}000\mu\text{s} - 1{,}000\mu\text{s} \\
T_{\text{margin}} &= \mathbf{19{,}000}\boldsymbol{\mu}\text{s}
\end{aligned}$$

Conclusion of Feasibility:
Even under this intentionally conservative timing budget, the control loop uses only $1{,}000/20{,}000 = 5\%$ of the available frame time. This shows that realistic timing variation and interface overhead do not threaten the required 50Hz acquisition rate, and the controller retains substantial margin for scheduling jitter and communication-stack activity.

Wireless Communication **Objective:** Prove that the chosen WiFi link can handle the generated gait data without creating a transmission bottleneck.

Mathematical Analysis:
First, calculate the payload size (S_{payload}) per sample iteration:

- 5 pressure sensors (12-bit ADC, padded to 16-bit / 2 bytes): $5 \times 2 = 10$ bytes
- 3-axis accelerometer data used for tilt estimation (16-bit / 2 bytes each): $3 \times 2 = 6$ bytes
- Timestamp/header: 4 bytes

$S_{\text{payload}} = 10 + 6 + 4 = 20$ bytes per sample.

At a 50Hz sampling rate, the required raw data rate (R_{req}) is:

$$
\begin{aligned}
R_{\text{req}} &= 20 \text{ bytes/sample} \times 50 \text{ samples/sec} \\
R_{\text{req}} &= 1{,}000 \text{ bytes/sec} \\
R_{\text{req}} &= \mathbf{8} \text{ kbps}
\end{aligned}
$$

Conclusion of Feasibility:
The required data throughput is 8 kbps. A typical embedded WiFi link provides practical application throughput far above 1 Mbps. Because 8 kbps $\ll 1$ Mbps, the wireless subsystem can transmit real-time diagnostic data without bandwidth limitations.

3 Cost

Part / Value	Designator	Qty	Cost (RMB)
ESP32-S3-WROOM-1-N8	U3	1	
MPU6050	U4	1	
CH340C	U1	1	
SY8088AAC	U2	1	
DB125-2.54-2P-GN-S	U5,U6,U7,U8,U9	5	
U262-161N-4BVC11	USB2	1	
SRV05-4	D1	1	
WS2812B-2020	RGB	1	
TSA343G00D-250J2	BOOT,EN	2	
MSK12C02	SW1	1	
UMH3N	Q3	1	
2.2uH	L3	1	
2.54-1*4P	H1	1	
PZ254V-11-02P	H2	1	
1N5819HW-7-F	D2	1	
LED-0603_R	LED1,LED2	2	
10uF / 10UF	C2,C3,C7,C10	4	
1uF	C1, C8	2	
100nF / 100NF	C4,C6,C9,C11,C12	5	
22pF	C5	1	
100K / 100k	R6, R7	2	
10K / 10k	R3,R4,R5,R11,R12,R13	6	
22k	R9	1	
5.1K	R15,R16	2	
2k	R1	1	
1k	R2,R8	2	
NC	R10	1	
Electronic Components Total			**500.00**
PCB Fabrication			**200.00**
3D Printing (Insole Housing)			**1500.00**
		Estimated Total Cost	**2200.00**

Table 3: Consolidated Bill of Materials and Manufacturing Budget

4 Schedule

The project schedule is organized by subsystem development, integration, testing, and final deliverables. Tasks are divided by technical focus so that each team member has a clear responsibility throughout the semester. The timeline below summarizes the main work plan and major course milestones. Weeks W1–W13 correspond to 3/2, 3/9, 3/16, 3/23, 3/30, 4/6, 4/13, 4/20, 4/27, 5/4, 5/11, 5/18, and 5/25.

Figure 14: Gantt-style schedule for StepWise, showing team responsibilities and major project milestones from early design to final report submission.

5 Ethics and Safety

5.1 Ethics

The StepWise system involves the collection and analysis of human gait data, which raises important ethical considerations related to privacy, data security, and responsible use of feedback.

First, the system collects sensitive personal data such as pressure distribution and motion signals. Although this data does not directly identify individuals, it may still reveal behavioral patterns. Therefore, all collected data should be anonymized and stored securely to prevent unauthorized access. Data transmission between the insole and the PC should also avoid storing unnecessary personal identifiers [1].

Second, user consent is essential. Participants must be informed about what data is collected, how it is used, and for what purpose. The system should only operate after obtaining clear user permission, following basic principles of informed consent in human-centered engineering systems [2].

Third, the feedback provided by the system should be carefully designed. Since StepWise gives gait-related suggestions, there is a risk that incorrect analysis may mislead users. Therefore, the system should clearly state that it is intended for assistive or educational purposes, not as a medical diagnostic tool [3].

5.2 Safety

Safety considerations in the StepWise system mainly involve electrical safety, mechanical reliability, and user comfort.

From an electrical perspective, the system is powered by a low-voltage battery and regulated circuits. Proper insulation and stable voltage regulation are necessary to prevent overheating or short circuits. All components should operate within their rated voltage and current limits to ensure safe operation [4].

From a mechanical perspective, the insole must withstand repeated loading during walking. The 3D-printed structure should be designed to tolerate continuous pressure cycles without deformation or failure. In addition, sharp edges and rigid components should be avoided to prevent injury or discomfort to the user.

From a usability perspective, long-term wearing comfort must be considered. The placement of sensors and electronic modules should not interfere with natural walking patterns. The system should also be tested under normal usage conditions to ensure it does not introduce instability or affect balance.

Finally, basic validation tests, including repeated mechanical loading checks for the insole structure and electrical stability checks for the powered system, should be conducted before user trials to reduce potential risks during operation [5, 4].

References

[1] European Union, "General Data Protection Regulation (GDPR)." [Online]. Available: https://gdpr.eu/, 2016.

[2] IEEE, "IEEE code of ethics." [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html, 2020.

[3] World Health Organization, "Ethics and governance of artificial intelligence for health." [Online]. Available: https://www.who.int/publications/i/item/9789240029200, 2021.

[4] International Electrotechnical Commission, "IEC 62368-1: Safety of electrical equipment," 2018.

[5] ASTM International, "ASTM D7791: Standard test method for uniaxial fatigue properties of plastics," 2022.