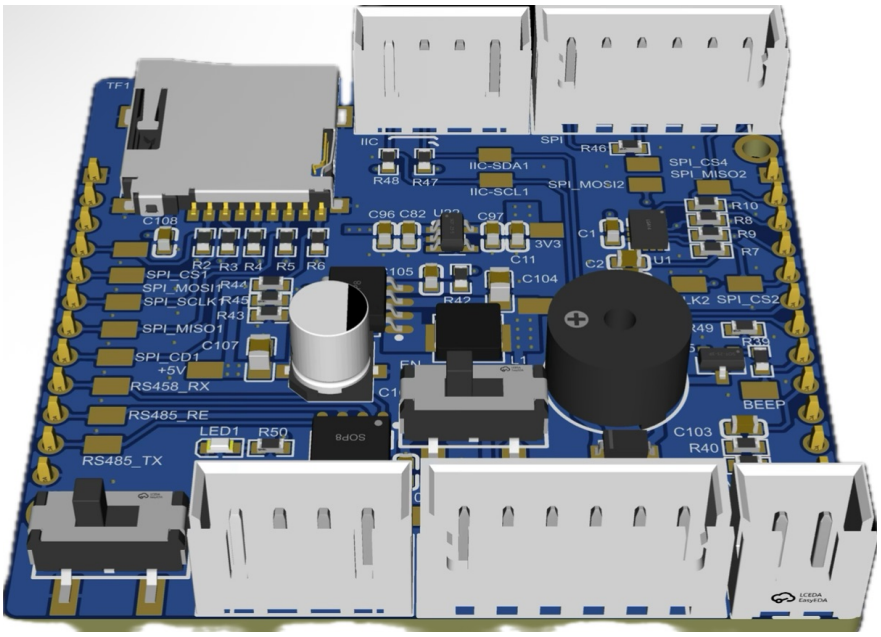


Design and Development of an adaptive robotic rehabilitation exoskeleton for lower limb

Pengcheng Shen

DEPARTMENT OFELECTRICAL ENGINEERING

Pengcheng Shen

DEPARTMENT OFELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Design and Development of an adaptive robotic rehabilitation exoskeleton for lower limb

Pengcheng Shen

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Design and Development of an adaptive robotic rehabilitation exoskeleton for lower limb

NAME Pengcheng Shen

Supervisor: Fabian Just, Department of Electrical Engineering
Examiner: Emmanuel Dean, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering

Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 763188496

Cover:Rendered view of the proposed custom expansion PCB.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Design and Development of an adaptive robotic rehabilitation exoskeleton for lower limb

Pengcheng Shen
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis focuses on the design and development of an adaptive robotic rehabilitation exoskeleton for lower-limb assistance. The main objective of the project is to create a more compact, integrated, and portable electronic system that can be easily embedded into lower-limb exoskeleton structures for practical rehabilitation use.

To achieve this goal, a custom electronic architecture based on the ESP32 Feather S3 and Arduino Portenta H7 microcontrollers was developed. The work includes the design of a custom expansion PCB, two adapter boards for MCU compatibility, and a fully customized IMU module for real-time motion sensing. These hardware components enable improved functionality, reduced system size, and enhanced structural integration. In addition, dedicated firmware was implemented to ensure stable communication, control, and data acquisition across the exoskeleton system.

Keywords:Lower-limb exoskeleton; Embedded system; ESP32 Feather S3; Arduino Portenta H7; Custom IMU; PCB design

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Fabian Just, and my examiner, Emmanuel Dean, for their continuous guidance and support throughout this thesis project. Their valuable advice and constructive feedback greatly contributed to the technical development and academic quality of this work.

I am also grateful to Yuhong Zhou, Zikun Wei, and Domenico Caliandro, whose assistance and collaboration provided many helpful ideas during the design phase of the system. In addition, I would like to thank Axel Sjögren Holtz, Adrian, and Oskar from the ETA laboratory for their support, insightful discussions, and helpful contributions during the course of this work.

Lastly, I sincerely thank my family for their endless encouragement, patience, and love. Their support has always been the motivation that kept me moving forward and enabled me to complete this thesis despite the challenges along the way.

Pengcheng Shen, Gothenburg, December 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

MCU	Micro-controller Unit
IMU	Inertial Measurement Unit
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver–Transmitter
GPIO	General-Purpose Input/Output
SD	Secure Digital memory card
LDO	Low-Dropout Regulator
PCB	Printed Circuit Board
JST	Japan Solderless Terminal connector
SMD	Surface-Mount Device
Li-Po	Lithium-Polymer battery

Contents

List of Figures

List of Tables

1
Introduction

This chapter presents the section levels that can be used in the template.

1.1 Background and Motivation

Lower-limb motor impairments caused by stroke, spinal cord injury, and neurological disorders affect millions of individuals worldwide each year. Conventional physiotherapy often relies on intensive manual assistance from therapists, which can limit training duration, repeatability, and quantitative assessment. In recent years, robotic rehabilitation exoskeletons have emerged as a promising solution to support gait training by providing controlled assistance, motion guidance, and objective motion tracking [1, 2, 3].

Despite these advances, many existing wearable exoskeleton platforms remain bulky and mechanically cumbersome due to distributed electronics, extensive wiring, and external control units. Insufficient integration between sensing electronics and mechanical structures can restrict joint-level sensor placement, reduce wearing comfort, and complicate system deployment in practical rehabilitation settings [4, 5]. In particular, motion sensing in research prototypes frequently relies on commercially available inertial measurement units (IMUs), which may be oversized, difficult to integrate into customized joint housings, or unsuitable for long-term wearable use [6, 7].

Recent studies have highlighted the importance of compact, joint-level sensing architectures that enable reliable motion capture while maintaining mechanical compatibility with wearable orthoses. Fabian Just and colleagues demonstrated that customized, mechanically integrated sensing modules can significantly improve signal stability and reduce system complexity in lower-limb exoskeletons [8, 9]. These findings emphasize the need for application-specific sensor and electronics design rather than reliance on generic breakout boards.

Beyond sensing accuracy, modern rehabilitation systems increasingly demand enhanced computational capability to support multi-sensor synchronization, real-time signal processing, and future algorithmic extensions such as adaptive control and learning-based methods [10, 11]. At the same time, wearable systems must balance processing performance with power efficiency, compact form factor, and mechanical robustness to remain suitable for daily use [12, 13].

Motivated by these challenges, this thesis aims to design a customized embedded

electronic system for a lower-limb exoskeleton that emphasizes reduced size, improved hardware integration, and flexible computational scalability. By consolidating sensing, communication, and data logging functions onto custom-designed PCBs and developing a dedicated compact IMU module, the proposed system seeks to bridge the gap between early-stage research prototypes and more deployable wearable rehabilitation platforms, while enabling future extensions toward higher-performance embedded processing and advanced rehabilitation algorithms [14, 15].

1.2 State of the Art

Over the past decade, lower-limb rehabilitation exoskeletons have progressed from early laboratory prototypes toward clinically relevant assistive systems capable of supporting gait training and functional recovery. Recent studies have demonstrated that wearable exoskeletons can provide repeatable assistance, objective motion assessment, and quantitative feedback during rehabilitation, contributing to improved motor outcomes and reduced therapist workload [16, 17].

In parallel with actively actuated exoskeletons, there has been increasing interest in passive and sensorized orthotic systems that focus on motion monitoring rather than force generation. Such systems aim to achieve lightweight construction, low power consumption, and high wearability while still providing accurate joint-level kinematic information [18]. Passive sensing-based orthoses are particularly suitable for long-term use and unsupervised rehabilitation scenarios, where safety, comfort, and robustness are critical considerations [19].

1.2.1 Embedded Systems in Rehabilitation Exoskeletons

To enable reliable motion monitoring, modern rehabilitation devices increasingly rely on embedded electronic systems integrating inertial measurement units (IMUs), joint encoders, and synchronized data acquisition pipelines [20]. However, many existing platforms continue to employ commercially available breakout boards and distributed electronic modules interconnected through extensive wiring. This approach often results in increased size, reduced mechanical compatibility, and susceptibility to signal interference, limiting practical deployment in compact lower-limb orthoses [21].

Recent research has emphasized the importance of tight hardware integration and custom embedded architectures to improve portability, reliability, and long-term usability of wearable rehabilitation systems [22]. Custom-designed PCBs that consolidate sensing, power regulation, communication, and data logging functions have been shown to significantly reduce wiring complexity while enabling stable multi-sensor synchronization [23].

Beyond sensing performance, emerging rehabilitation applications increasingly require enhanced computational capability to support real-time signal processing, sensor fusion, and future extensions toward adaptive or learning-based algorithms [24].

This trend motivates the exploration of embedded hardware architectures that can balance low-power operation with scalable processing performance, while maintaining compatibility with wearable constraints [25].

Despite these advances, there remains a lack of compact, modular embedded platforms specifically tailored for passive knee rehabilitation orthoses. Addressing this gap motivates the development of a highly integrated sensing system based on custom PCB design and dedicated inertial sensing hardware, as proposed in this thesis.

1.3 Research Objectives and Scope

Research Objectives:
1.To develop a fully customized PCB that integrates IMUs, encoder interfaces, signal conditioning, data acquisition, and wireless communication into a single embedded platform.

2.To design a compact mechanical enclosure tailored to the geometry of the rehabilitation brace, reducing system size and improving joint-level sensor placement compared to the previous multi-layer housing.

3.To ensure reliable, synchronized sensing performance through hardware redesign, embedded-software development, and systematic evaluation of measurement accuracy and system stability.

4.To validate the redesigned system experimentally, demonstrating improvements in integration level, wearability, and usability relative to the original modular prototype.

Research Scope::

This study focuses on the design, integration, and evaluation of a compact embedded sensing system for a knee-rehabilitation orthosis. The work includes the development of a custom PCB integrating two IMUs (one onboard and one external custom module), a microSD interface for local data logging, and adapter boards supporting both ESP32-S3 and Arduino Portenta H7 microcontrollers. The research aims to establish a functional proof-of-concept demonstrating improved integration, reduced size, and reliable synchronized sensing. Mechanical redesign, advanced control strategies, and clinical trials lie outside the scope of this thesis.

1.4 Structure of the Thesis

This thesis is organized into five chapters, each addressing a key stage in the design, integration, and evaluation of the proposed embedded sensing system for knee-rehabilitation applications.

- **Chapter 1 – Introduction**
  Provides the background, motivation, and research context. It outlines limitations of the previous modular prototype and defines the objectives, scope,

and contributions of the integrated embedded system developed in this work.

- **Chapter 2 – Theory**
  Presents relevant theoretical foundations, including sensing principles for IMUs and encoders, embedded-system concepts, and data-logging requirements. It also reviews existing sensorized rehabilitation devices and identifies design considerations guiding the new system architecture.
- **Chapter 3 – Methods**
  Describes the design and implementation of the proposed system. This includes the development of the custom PCB, the integration of two IMUs (one onboard and one external module), the microSD-based data-logging interface, and the adapter boards used for ESP32 Feather S3and Arduino Portenta H7 platforms. Firmware development and system assembly are also detailed.
- **Chapter 4 – Results and Discussion**
  Presents experimental results evaluating sensing accuracy, synchronization performance, data-logging reliability, and improvements in system compactness compared to the previous multi-board prototype. The findings are discussed in relation to the research objectives and identified limitations.
- **Chapter 5 – Conclusion and Future Work**
  Summarizes the main contributions of the thesis and outlines potential future extensions, including advanced data fusion, real-time control integration, and long-term deployment in rehabilitation settings.

In summary, this section provides an overview of the thesis structure, showing how the work progresses from theoretical background and system design to implementation, evaluation, and future development.

2

Theory

2.1 System Requirements and Design Considerations

This chapter outlines the system-level requirements and design considerations that guided the development of the integrated sensing hardware. These requirements include mechanical constraints, electrical and communication needs, modularity of the microcontroller platform, and power-system considerations. Together, they form the foundation for the hardware architecture presented in Chapter 3.

2.1.1 Overview of System Requirements

The objective of this project is to design a compact, reliable, and highly integrated sensing platform for a lower-limb exoskeleton. In the previous prototype, the sensing and processing electronics relied on multiple commercial breakout boards interconnected through extensive wiring. This architecture limited mechanical integration, increased system bulk, and reduced long-term robustness under wearable operating conditions.

To address these limitations and support practical deployment in a wearable rehabilitation device, the new hardware architecture is required to satisfy the following system-level requirements:

- **Reduced system size and improved mechanical compatibility:** The sensing electronics and power components must be physically compact and mechanically compatible with the lower-leg exoskeleton housing.
- **Multi-sensor support with synchronized data acquisition:** The system must support multiple digital sensing modalities, including inertial sensing and joint-level measurements, with reliable synchronization across sensors.
- **Wireless communication capability:** Integrated wireless connectivity is required to enable data transfer, monitoring, and debugging without reliance on physical interfaces.
- **Long-duration battery-powered operation:** The system must operate continuously for extended periods under battery power, suitable for experimental data collection sessions.
- **Simplified wiring and improved assembly reliability:** Internal wiring

complexity should be minimized to enhance mechanical robustness, reduce assembly errors, and improve long-term reliability in a wearable environment.

- **Computational scalability:** The hardware architecture should accommodate varying computational demands, enabling both low-power operation and future extensions toward higher-performance processing when required.

These high-level requirements guided the subsequent mechanical, electrical, and architectural design decisions presented in the following sections.

2.1.2 Mechanical and Form-Factor Requirements

All electronic components of the sensing system must be integrated within the lower-leg segment of the exoskeleton, where available space is highly constrained. As a result, both the physical dimensions and mounting strategy of each PCB play a critical role in ensuring mechanical compatibility, user comfort, and system robustness.

To meet the mechanical and form-factor constraints of a wearable lower-limb exoskeleton, the following requirements were identified:

- **Miniaturized IMU module:** The IMU module must be substantially smaller than the previous commercial breakout board to enable placement close to the knee joint and within narrow structural regions.
- **Reduced internal wiring:** Internal cabling must be minimized to improve mechanical robustness, reduce assembly complexity, and enhance long-term reliability under repeated motion.
- **Compact PCB footprints:** All custom PCBs must be designed with constrained outlines to ensure proper fit within the lower-leg housing while allowing secure mechanical mounting.
- **Lightweight system enclosure:** The main system enclosure must be smaller and lighter than that of the previous prototype to reduce distal mass, improve wearer comfort, and minimize interference with natural gait dynamics.
- **Distributed but integrated placement:** Electronic components must fit within narrow regions near the knee joint and lower leg without protruding beyond the mechanical profile of the exoskeleton.

These mechanical constraints directly motivated the development of a custom compact IMU module and a reduced-size expansion PCB, as well as the redesign of the main system enclosure to achieve improved integration and wearability.

2.1.3 Electrical and Communication Requirements

The sensing subsystem is required to interface with multiple digital peripherals and support reliable data acquisition, storage, and wireless communication. The following electrical and communication requirements were identified:

- **Multi-device SPI communication:** The onboard IMU, external IMU module, rotary encoder, and microSD card interface are all required to operate

over SPI. This necessitates reliable signal routing, stable chip-select control, and robust shared-bus arbitration.

- **Synchronized multi-sensor operation:** The system must support deterministic and synchronized data sampling across multiple sensors to ensure consistent joint-level motion reconstruction.
- **High-throughput data logging:** Continuous multi-sensor acquisition requires sustained write performance to external storage, motivating the inclusion of a dedicated microSD interface.
- **Wireless connectivity:** Wireless communication is required for remote monitoring, data transfer, and debugging. Support for both Wi-Fi and Bluetooth was therefore identified as a core system requirement.
- **Stable power distribution:** All peripherals must be supplied with low-noise and regulated power rails to ensure reliable operation during long-duration measurements.

2.1.4 Modularity and Microcontroller Flexibility

A key design requirement of the system is flexibility in microcontroller selection. Wearable rehabilitation applications span a wide range of use cases, from low-power long-term motion monitoring to more computationally demanding scenarios involving real-time signal processing, sensor fusion, or learning-based algorithms.

To accommodate these diverse requirements, the hardware architecture was designed to remain agnostic to a specific microcontroller platform. Instead of tightly coupling the sensing electronics to a single MCU, a modular interface approach was adopted, allowing different microcontroller platforms to be evaluated and integrated without redesigning the core sensing PCB.

This flexibility enables the use of low-power wireless microcontrollers for energy-efficient data logging and communication, as well as higher-performance embedded processors when increased computational capability is required. Differences in pin configuration, voltage levels, peripheral availability, and development ecosystems across platforms motivated the separation of microcontroller-specific circuitry from the main sensing hardware.

By introducing dedicated adapter boards and standardized electrical interfaces, the proposed system supports scalable hardware evolution while preserving compatibility with existing sensing, power, and communication subsystems. This modular design strategy facilitates future experimentation and platform upgrades without compromising mechanical integration or sensing reliability.

3

Methods

3.1 Overview

The redesigned sensing subsystem aims to replace the previous prototype's modular, multi-board electronic architecture with a compact and fully integrated embedded platform. In the earlier system, sensing and processing functions were distributed across several independent breakout boards, including multiple IMUs, a rotary encoder interface, a wireless communication module, and an external microSD logger. These components were interconnected through extensive wiring and housed within a large multi-layer enclosure. While suitable for early-stage prototyping, this arrangement increased system thickness and weight, introduced wiring fragility, and complicated synchronized data acquisition, ultimately limiting wearability and long-term reliability during rehabilitation use.

To address these limitations, the proposed system consolidates all sensing, processing support, and data-logging interfaces onto a single custom-designed PCB. The board integrates power regulation, dual IMU interfaces, encoder connectivity, microSD storage, and standardized microcontroller expansion connectors within a compact form factor. Two inertial sensing units are incorporated into the design: an onboard IMU that provides a stable reference frame for the system, and a custom external IMU module positioned closer to the knee joint to improve localized motion-capture accuracy.

Beyond mechanical integration, the redesigned architecture was developed to accommodate varying computational requirements across different stages of system deployment. Basic sensing, logging, and wireless communication tasks can be supported by low-power microcontroller platforms, while more advanced applications—such as high-rate sensor fusion, real-time signal processing, and learning-based algorithms—require substantially higher computational performance. To avoid redesigning the sensing hardware for each use case, the system adopts a modular processing interface that decouples sensing from computation, enabling interchangeable microcontroller platforms to be used with the same sensing PCB.

This integrated and modular architecture significantly reduces system size, minimizes external wiring, improves mechanical compatibility with the rehabilitation brace, and enhances synchronized sensing performance. The following sections describe the embedded hardware design, microcontroller interface strategy, external IMU module, and full system assembly in detail.

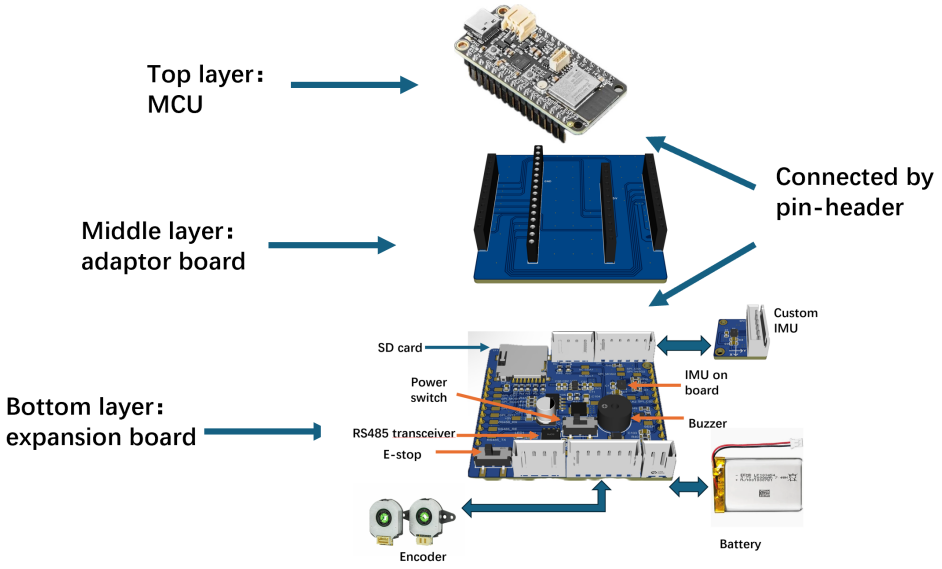


Figure 3.1: System overview diagram.

3.2 Processing Platform Selection and Computational Requirements

While the primary objective of this work is the development of a compact and highly integrated sensing hardware platform, the selection of an appropriate processing unit plays a critical role in determining system capability, scalability, and long-term applicability. Early-stage wearable rehabilitation prototypes typically emphasize low power consumption and basic sensing functionality. However, more advanced experimental and research-oriented scenarios impose substantially higher computational demands.

Beyond basic data acquisition, modern sensorized exoskeletons increasingly require higher computational capability to support synchronized sensing, real-time processing, and advanced algorithms, which exposes the limitations of ultra-low-power microcontroller platforms.

To support future extensions toward closed-loop control and integration with a robotics software stack, the system should be compatible with ROS 2. In particular, microcontroller-side ROS communication (e.g., micro-ROS) is desirable to enable lightweight publishing of sensor data and potential reception of control commands during future development. Therefore, a comparison of representative candidate processing platforms was conducted, focusing on computational capability, wireless connectivity, and ROS compatibility.

These evolving requirements reveal a limitation of relying exclusively on ultra-low-power microcontroller platforms. While such platforms are well suited for continuous sensing and wireless communication under strict energy constraints, they may

struggle to support concurrent peripheral operation, higher sampling rates, and computationally intensive processing without compromising timing stability or system robustness.

To support future extensions toward closed-loop control and integration with a robotics software stack, the system should be compatible with ROS 2. In particular, microcontroller-side ROS communication (e.g., micro-ROS) is desirable to enable lightweight publishing of sensor data and potential reception of control commands during future development. In addition, the ESP32 family has existing support for ROS 2 integration through micro-ROS, which provides a potential path for publishing sensor data to a ROS 2 system and enabling future control-oriented extensions. Candidate Platform Comparison

3.2.1 Processing Platform Comparison

Table 3.1: Comparison of candidate processing platforms (capability overview).

Platform	Core / Processor	Wi-Fi	BT	ROS support
ESP32-S3 Feather	Dual-core ESP32-S3 MCU	Yes	BLE	micro-ROS
Arduino Portenta H7	Dual-core STM32H747 MCU	Yes	Yes	micro-ROS
Raspberry Pi 4 Model B	Quad-core Cortex-A72 SBC	Yes	Yes	Full ROS
STM32F429ZI	Single-core Cortex-M4 MCU	No	No	micro-ROS

3.2.2 Platform Selection Rationale (ESP32-S3 Feather as Baseline)

In this work, the primary processing platform for the sensing prototype was selected based on a balance between computational capability, power consumption, integration complexity, and overall development reliability. Besides supporting multi-sensor acquisition and synchronized communication, the platform must maintain stable peripheral access and repeatable firmware iteration under battery-powered operation, since the system is intended for long-duration wearable experiments.

Several candidate platforms were considered. Ultra-low-power microcontrollers provide excellent energy efficiency but may struggle to support concurrent peripheral operation, higher sampling rates, and continuous onboard data logging without compromising timing stability. High-performance options such as the Arduino Portenta H7 offer significant computational headroom and enable future extensions toward real-time processing and on-device learning. However, such platforms also introduce higher integration complexity, especially due to extensive pin multiplexing and platform-dependent startup behavior.

Based on these considerations, an ESP32 platform was selected as the baseline processing solution due to its strong balance between performance, power efficiency, cost, and ecosystem maturity. In particular, the system was implemented using the Adafruit ESP32-S3 Feather board rather than a generic ESP32 development board.

This board provides practical advantages that directly match the requirements of a compact wearable sensing prototype, including an onboard LiPo battery connector with integrated battery charging circuitry, convenient USB-C power and programming, and a built-in battery fuel gauge that enables monitoring of battery state during long-run experiments. These features reduce integration effort and improve experimental repeatability while still providing sufficient computational capability for multi-sensor acquisition and SD-card-based data logging.

3.2.3 High-Performance Processing Platform Integration

To complement low-power wireless microcontroller platforms and enable future extensions toward more computationally demanding applications, a higher-performance embedded processing platform is introduced into the system architecture. The Arduino Portenta H7 is selected as a representative high-performance option due to its suitability for embedded and industrial applications.

The Portenta H7 features a dual-core architecture combining a high-performance ARM Cortex-M7 processor with a power-efficient Cortex-M4 core. This heterogeneous architecture enables computationally intensive tasks, such as high-rate signal processing and algorithmic computation, to be offloaded to the M7 core, while time-critical or low-power tasks can be handled by the M4 core.

By supporting the Arduino Portenta H7 through a dedicated adapter interface, the proposed sensing platform enables future experimentation involving real-time processing, complex algorithm deployment, and on-device learning, without sacrificing mechanical integration or compatibility with wearable system constraints. This approach provides a scalable hardware foundation that bridges low-power sensing applications and higher-performance research-oriented use cases.

3.3 Embedded Hardware Design

The embedded hardware of the proposed system consolidates sensing, logging, communication, and power management into a custom-designed PCB. The design replaces the modular breakout-board architecture of earlier prototypes with a compact and mechanically robust platform tailored for wearable knee-rehabilitation applications. The following subsections describe the structure and functionality of each hardware component.

3.3.1 Modular Microcontroller Interface Architecture

The system is designed to operate with two interchangeable processing platforms: the ESP32 Feather S3 and the Arduino Portenta H7. To support both microcontrollers without redesigning the main board, the PCB integrates two dedicated adapter interfaces that route all functional signals—SPI, I^2C, UART, GPIO, and power rails—into the corresponding connectors.

This modular MCU interface enables compatibility with different processing architectures, rapid switching during development, and consistent peripheral routing regardless of which platform is installed. The adapters also provide mechanical reinforcement, eliminating the need for external wiring between the PCB and the microcontroller.

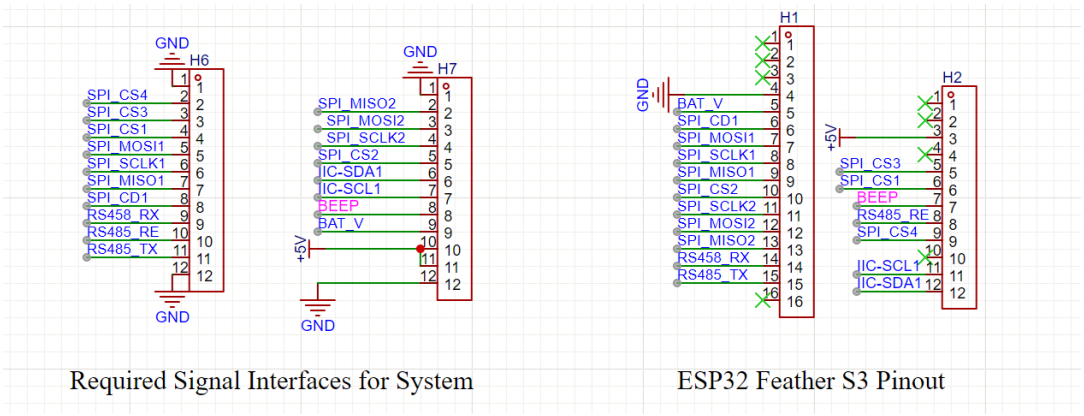


Figure 3.2: System interface requirements and MCU pin mapping. The left portion illustrates the external signal requirements of the sensing system, while the right portion shows the corresponding ESP32 Feather S3 pin resources.

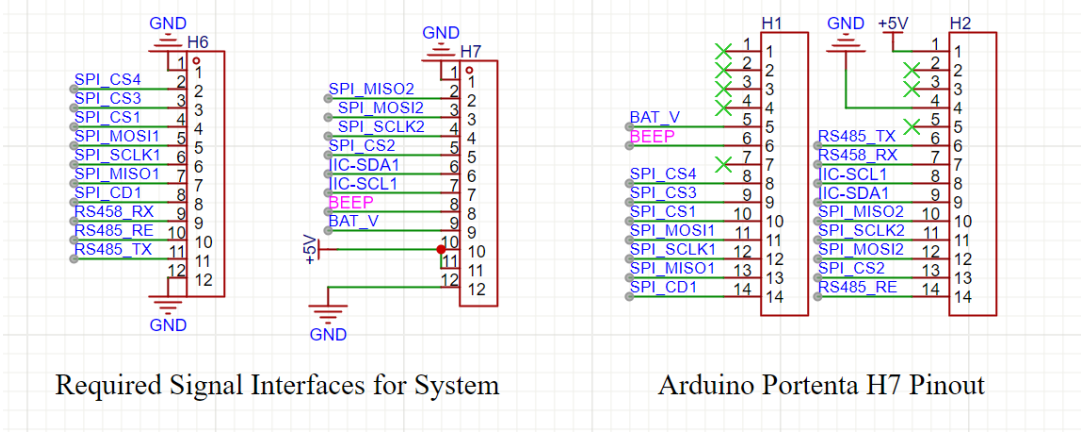
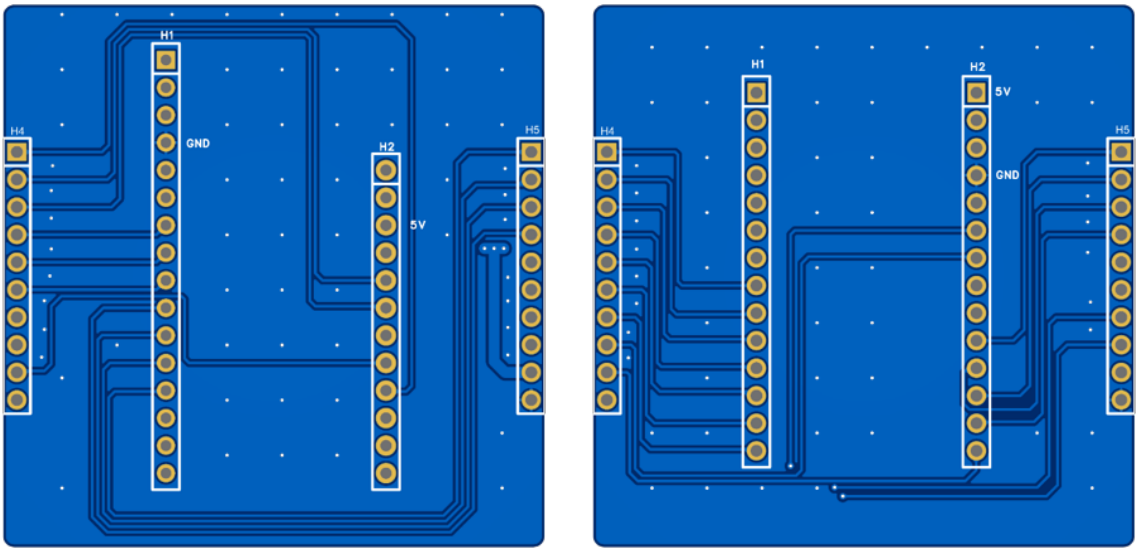


Figure 3.3: System interface requirements and MCU pin mapping for the Arduino Portenta H7 platform.

Based on the required system signal interfaces and the available MCU pin resources, two adaptor boards were designed to bridge the sensing subsystem with different MCU platforms.



(a) Adaptor Board for ESP32 Feather S3 layout.

(b) Adaptor Board for Arduino Portenta H7 layout.

Figure 3.4: Custom adaptor boards designed to map the required system interfaces to different MCU platforms. The boards route SPI, I^2C, RS485, and power signals while maintaining a compact footprint suitable for wearable integration.

3.3.2 Inertial Sensing Architecture

The system incorporates two inertial measurement units to achieve accurate characterization of knee-joint motion.

3.3.2.1 On-board IMU

The primary IMU is integrated directly onto the main PCB and serves as a stable reference sensor for the system. Its rigid mounting within the electronics enclosure provides a consistent orientation reference, reducing sensitivity to cable motion and external disturbances.

The PCB layout incorporates dedicated power decoupling, local filtering, and grounding strategies to ensure low-noise operation and support high-frequency sampling. The IMU interfaces with the microcontroller via the shared SPI bus described above, using a dedicated chip-select line.

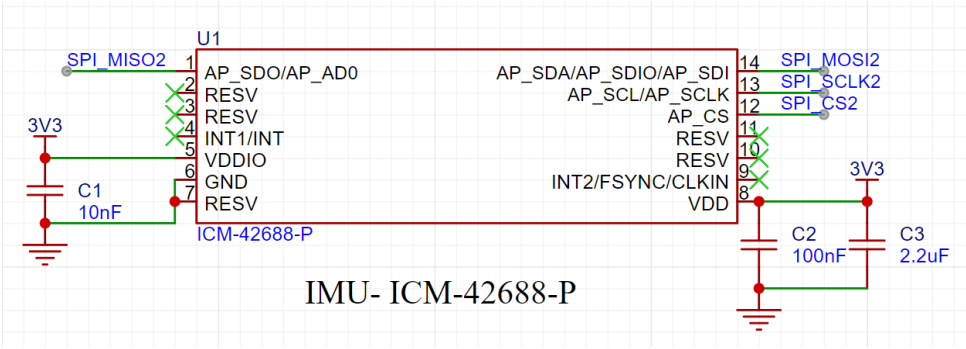


Figure 3.5: Schematic of the on-board IMU.

3.3.2.2 External IMU Module

A second IMU is implemented as a compact external module that can be positioned closer to the knee joint to capture limb motion independently of the brace structure. The module connects to the main PCB via a lightweight cable and a board-mounted JST connector, allowing flexible sensor placement while maintaining synchronized sampling with the on-board IMU.

In early prototyping, the ICM-42688-P sensor was evaluated using a commercial breakout board, namely the MikroElektronika 6DOF IMU 14 Click. This breakout board integrates standardized mikroBUS™ headers, selectable SPI/I^2C communication interfaces, I^2C address configuration jumpers, and on-board indicators, and is primarily intended for rapid evaluation and general-purpose development.

According to the manufacturer specifications, the breakout board measures 28.6 mm $\times$ 25.4 mm. While its configurable interface options and auxiliary components provide flexibility for bench-top testing, these features are not required in the proposed wearable rehabilitation system, where the communication protocol, wiring scheme, and power configuration are fixed. The use of a commercial breakout board would therefore introduce unnecessary footprint and redundant circuitry without providing functional benefits for the final integrated system.

To address these limitations, a custom IMU module was designed based on the ICM-42688-P reference schematic. The custom design retains only the essential sensing circuitry, SPI interface routing, and power decoupling components, while eliminating non-essential headers, jumpers, and indicators. The resulting module measures approximately 18 mm $\times$ 18 mm, representing a substantial reduction in size compared to the commercial breakout board.

The custom IMU interfaces with the main PCB via a board-mounted JST connector, providing a direct electrical and mechanical connection while minimizing wiring complexity. This connector-based integration improves system-level robustness and supports accurate motion measurement in wearable rehabilitation applications.

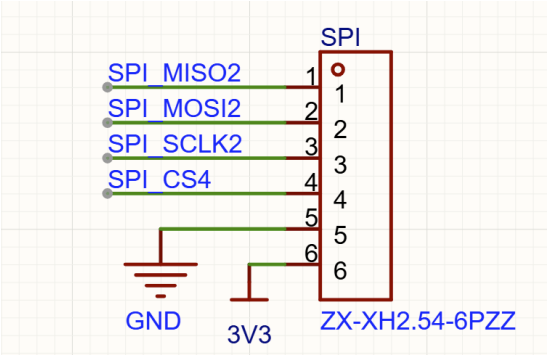


Figure 3.6: Schematic of the connector for external IMU interface.

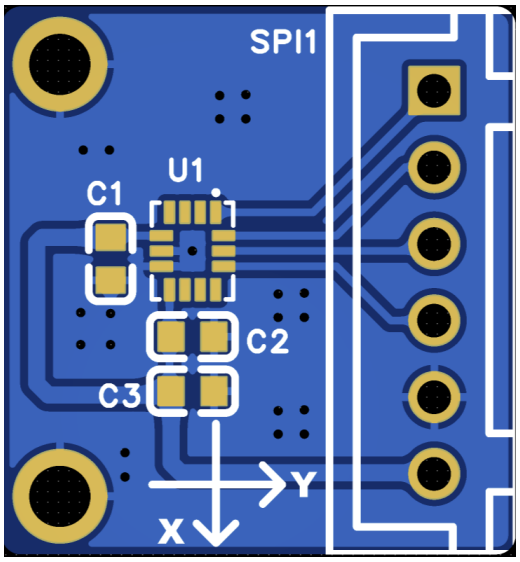


Figure 3.7: PCB layout of the custom external IMU module.

3.3.3 Rotary Encoder Interface

The knee joint angle is measured using an absolute magnetic rotary encoder interfaced with the microcontroller via SPI. In the proposed system, the rotary encoder shares a dedicated SPI bus with the integrated microSD interface, while an independent SPI bus is used for the onboard and external IMUs. Separate chip-select (CS) lines are used to address each device on their respective buses. This shared

SPI architecture reduces interface and GPIO requirements on compact MCU platforms such as the ESP32 Feather S3, while maintaining synchronized and reliable access to all motion sensors. All encoder-related circuitry, including SPI routing and power regulation, is integrated directly onto the main PCB. A dedicated JST

connector provides a direct electrical and mechanical interface to the encoder, replacing splitter cables used in earlier prototypes. This approach simplifies system assembly and results in a compact and mechanically robust interface suitable for wearable applications.

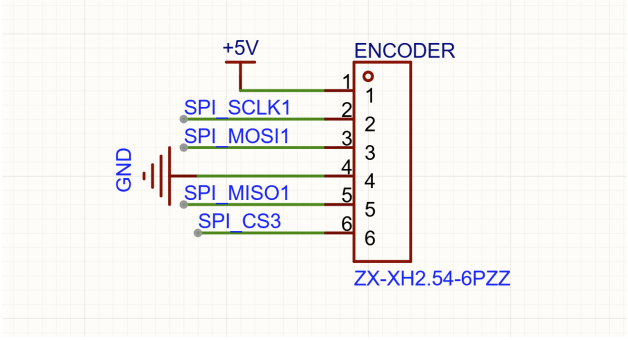


Figure 3.8: Schematic of the rotary encoder interface.

3.3.4 Integrated microSD Logging System

The system integrates a microSD card slot on the main PCB to support local data logging of IMU and encoder measurements. The microSD card is interfaced with the microcontroller via SPI and shares the same SPI bus as the rotary encoder, while the IMUs are connected on a separate SPI bus. Devices on each bus are addressed using dedicated chip-select (CS) lines, enabling reliable access and continuous logging without bus contention. The interface also includes pull-up resistors on the SPI lines and a card-detect (CD) signal to improve robustness and support reliable card insertion detection.

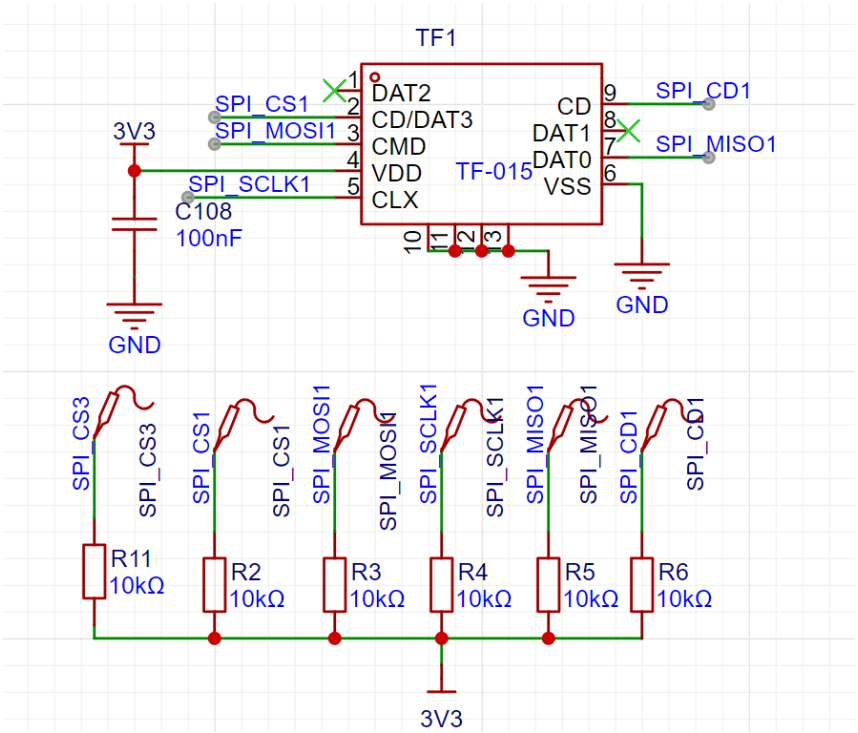


Figure 3.9: Schematic of the integrated microSD interface.

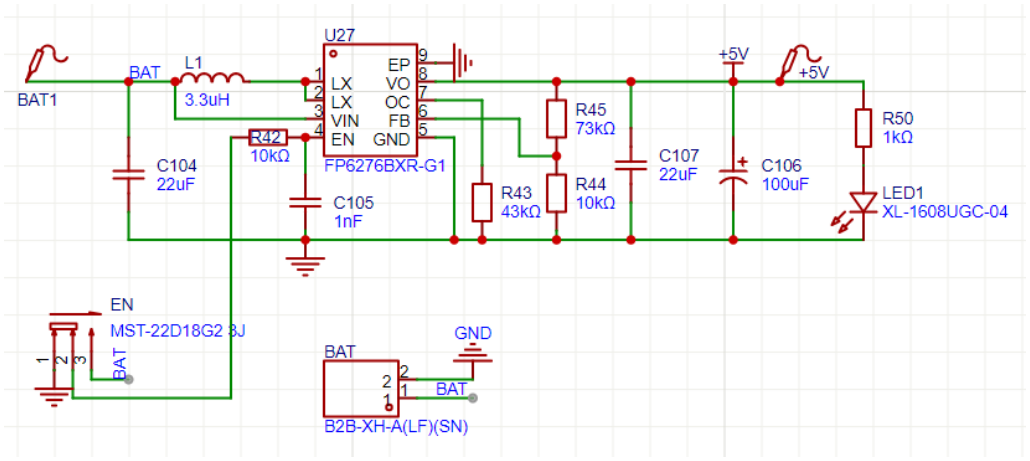
3.3.5 Power Regulation and Distribution

The power subsystem is designed to provide stable and isolated supply rails for the sensing modules and supported microcontroller platforms. The system is powered by a single Li-Po battery, which is first regulated to a 5 V rail using a dedicated boost converter.

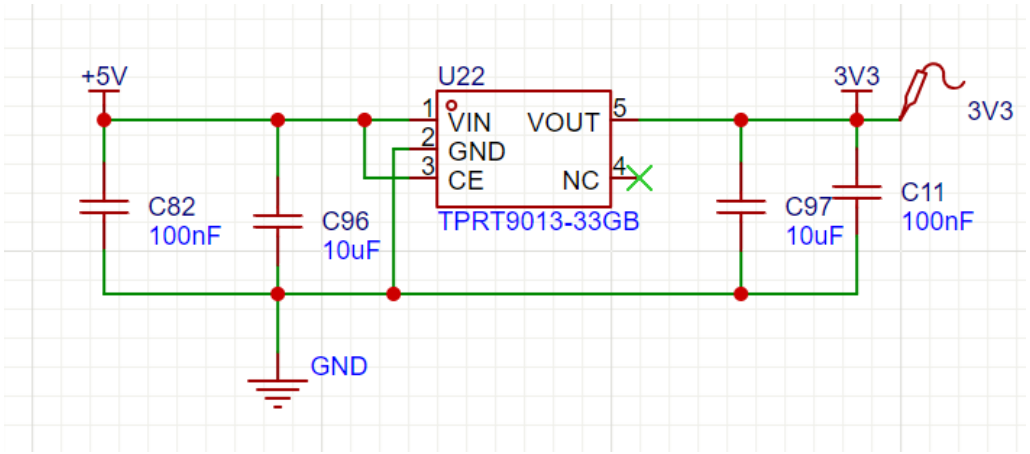
The regulated 5 V supply serves two purposes: it directly supports microcontroller platforms that require a 5 V input, such as the Arduino Portenta H7, and it acts as the input to a low-noise linear regulator. A 3.3 V LDO stage is used to generate the primary logic supply for the IMUs, rotary encoder interface, microSD card circuitry, and other digital components.

Local decoupling capacitors and filtering networks are implemented at each regulation stage to minimize voltage ripple and noise coupling into the sensing subsystem. This cascaded boost–LDO architecture enables compatibility with multiple MCU

platforms while maintaining a clean and stable 3.3 V supply for sensitive sensor electronics.



(a) Boost converter for 5V power generation.



(b) LDO voltage regulation circuit.

Figure 3.10: Power management circuits of the system, including the LDO for 3.3 V regulation and the boost converter for 5 V supply.

3.3.6 Expansion and Debug Interfaces

To support testing, calibration, and firmware development, the PCB integrates several low-profile debugging and expansion interfaces. Instead of conventional pin headers, the board employs surface-mount test points (Harwin S2751-46R) to expose critical signals such as power rails, SPI and I^2C lines, GPIOs, and microcontroller programming pins.

These compact SMD test points allow direct access using oscilloscope probes, logic-analyzer hooks, or pogo-pin clips, enabling non-intrusive measurement of communication and power signals during development. Additional expansion nodes are routed to auxiliary test points to support future sensor integration and hardware extensions while maintaining the compact form factor of the embedded subsystem.

3.4 PCB Manufacturing, Assembly, and Hardware Validation

All PCBs designed in this project, including the main expansion board, the two microcontroller adapter boards, and the external IMU module, were manufactured by a PCB fabrication service using the design files developed in this work. The following subsections describe the fabrication process, manual assembly steps, and hardware validation procedures performed before integrating the electronics into the exoskeleton system.

3.4.1 PCB Fabrication

The printed circuit boards developed in this work were fabricated by a commercial PCB manufacturer JLC using standard FR-4 processes. Different PCB layer configurations were selected according to the functional role and integration requirements of each board within the system.

3.4.1.1 Four-layer Expansion Board

The main expansion board was fabricated as a four-layer PCB. This decision was mainly driven by signal integrity and power distribution considerations. The expansion board integrates multiple high-speed SPI peripherals (e.g., IMUs, SD card interface, and encoder communication) together with mixed-signal routing and distributed power rails in a compact layout.

A four-layer stack-up allows dedicating continuous reference planes (ground and power) underneath critical signal traces, which improves return current paths, reduces loop area, and helps mitigate noise coupling and electromagnetic interference. In addition, using internal planes simplifies routing density and reduces the need for long detours or narrow trace bottlenecks, improving reliability during high-rate sampling and continuous data logging.

3.4.1.2 Two-layer Peripheral Modules

In contrast, the two MCU adapter boards and the external custom IMU module were implemented as two-layer PCBs. These boards primarily serve as interface and sensing modules with limited routing complexity. A two-layer design was sufficient to achieve compact layouts, short signal paths, and reliable operation while minimizing manufacturing cost and design overhead.

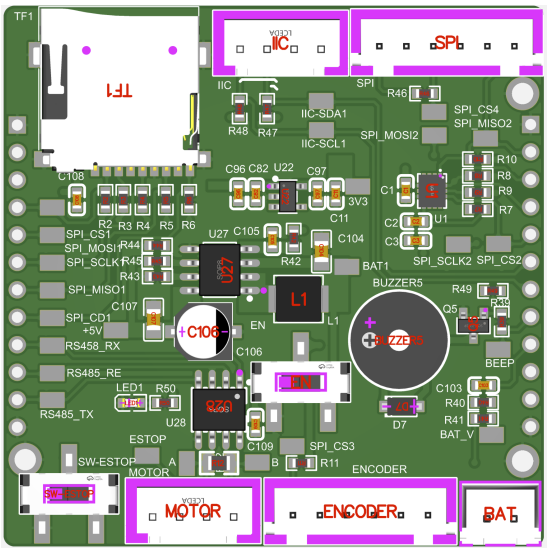


Figure 3.11: Annotated component placement of the expansion board PCB (manufacturing reference).

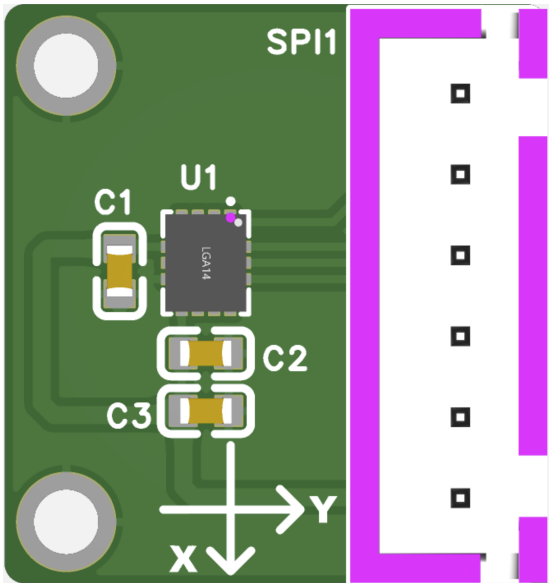
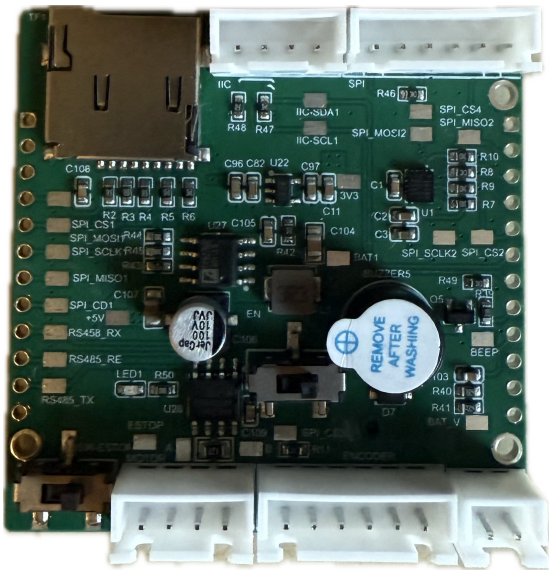


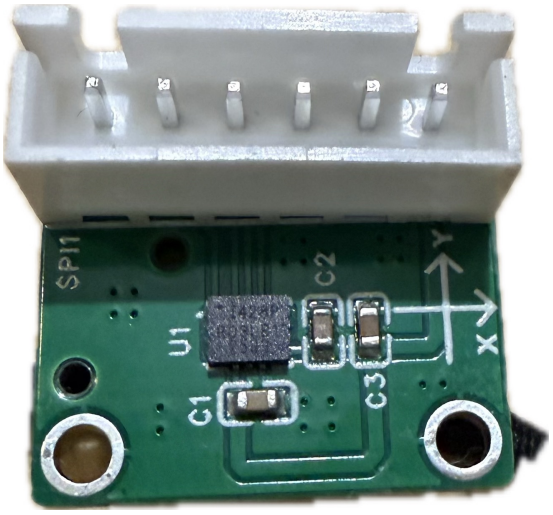
Figure 3.12: Annotated component placement of the custom IMU PCB (manufacturing reference).

All PCBs were manufactured using FR-4 substrate material with a standard thickness of 1.6 mm and finished with a HASL surface finish. Minimum trace width, spacing, via dimensions, and solder-mask parameters were selected in accordance with the manufacturer's design rules to ensure fabrication reliability.

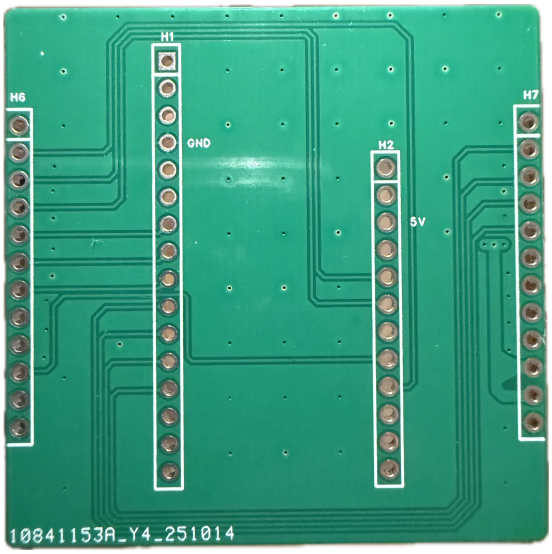
Surface-mount assembly for the main board, adapter boards, and the external IMU module was performed by the manufacturer. Following delivery, the boards were ready for mechanical integration and system-level assembly.



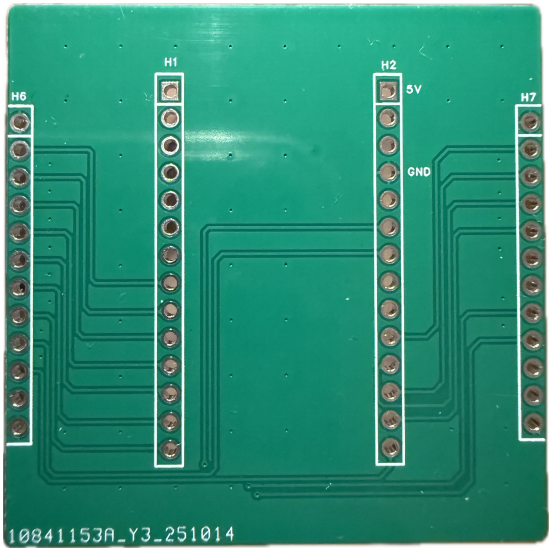
(a) Fabricated expansion board.



(b) Fabricated custom IMU module.



(c) Adapter board for ESP32 Feather S3.



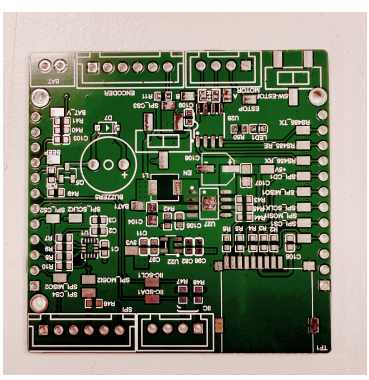
(d) Adapter board for Arduino Portenta H7.

Figure 3.13: Fabricated printed circuit boards developed for the sensing system, including the main expansion board, custom IMU module, and microcontroller adapter boards.

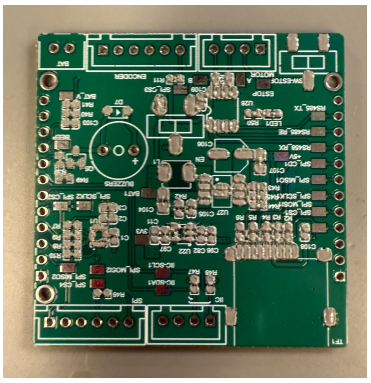
3.4.2 PCB Manual Assembly and Inspection

Following fabrication, the other PCBs were assembled using a combination of manual and stencil-based soldering processes, depending on board complexity and component density. 5 prototype boards was assembled entirely by hand soldering, primarily for early-stage validation and debugging purposes.

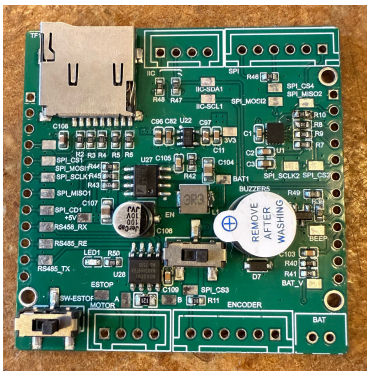
The remaining five boards were assembled using solder paste applied through a stainless-steel stencil, followed by reflow soldering in a temperature-controlled reflow oven. This approach ensured consistent solder joint quality and improved assembly repeatability across boards. For the inertial measurement unit (IMU), which required precise placement and controlled heating, hot-air gun rework was used to solder the device after the main reflow process.



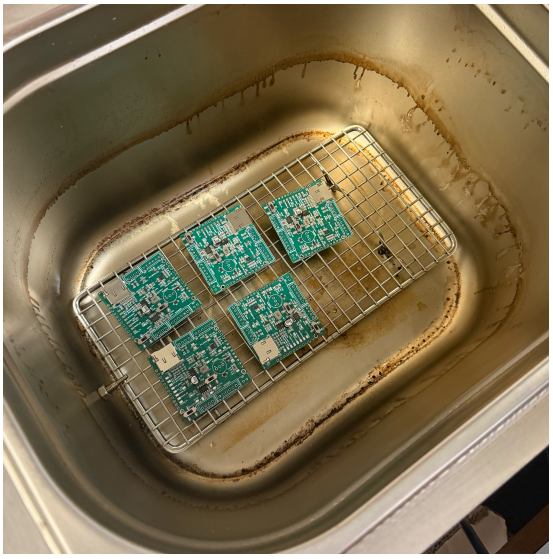
(a) Bare PCB without components.



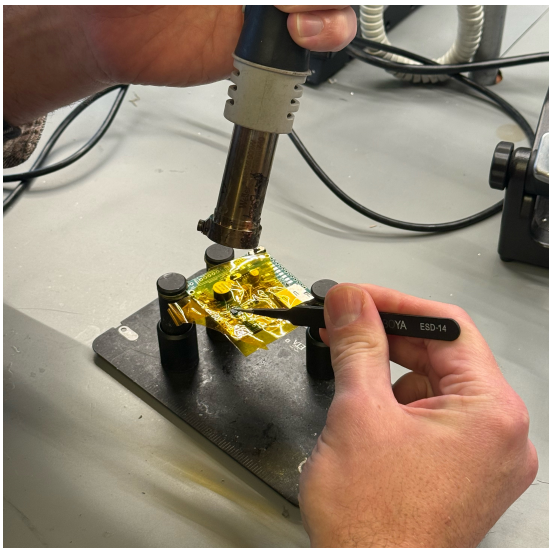
(b) Solder paste applied using stencil.



(c) All components placed prior to reflow.



(d) PCB placed in reflow oven.



(e) IMU soldered using hot-air rework.

Figure 3.14: PCB assembly process, including bare board fabrication, stencil-based solder paste application, component placement, reflow soldering, and hot-air rework for the IMU.

After assembly, all boards underwent visual inspection to verify component alignment, solder joint integrity, and the absence of visible defects such as solder bridges or insufficient wetting. Electrical inspection was subsequently performed using a multimeter probe to check for unintended shorts between power and ground rails as well as continuity across critical signal paths. These inspection steps ensured correct assembly prior to power-up and functional testing.

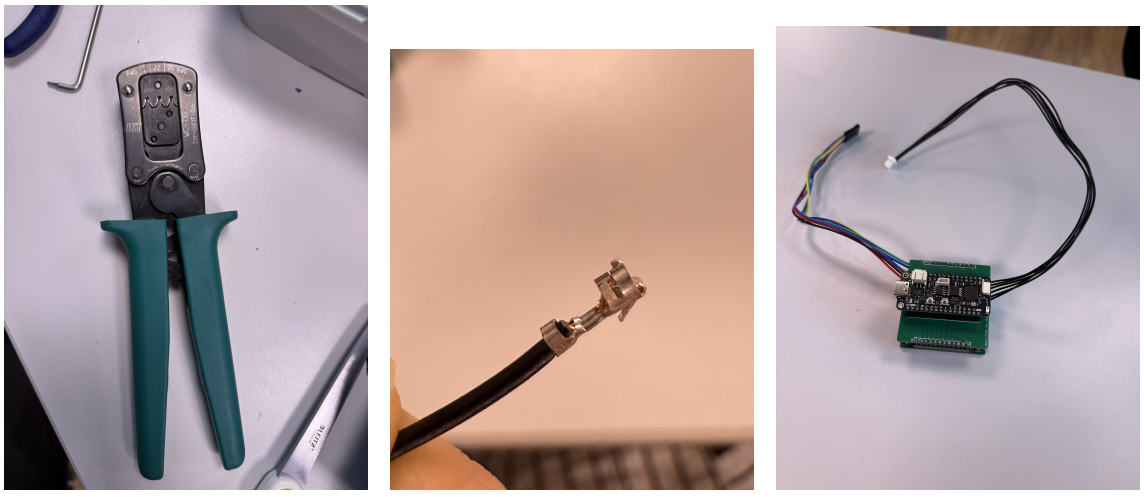
3.4.3 Interconnect Assembly and Cable Preparation

After PCB fabrication and SMT assembly, several minor manual steps were performed to complete system integration. These tasks primarily involved inter-board interconnects and cable preparation for external sensors.

(1) Header installation for adapter boards. Pin headers and board-to-board connectors were soldered to connect the expansion board with the MCU adapter boards, enabling interchangeable microcontroller platforms.

(2) Cable preparation for external sensors. Custom wire harnesses were prepared for the rotary encoder and the external IMU using JST connectors. The work included wire stripping, terminal crimping, and housing insertion to match the required cable lengths.

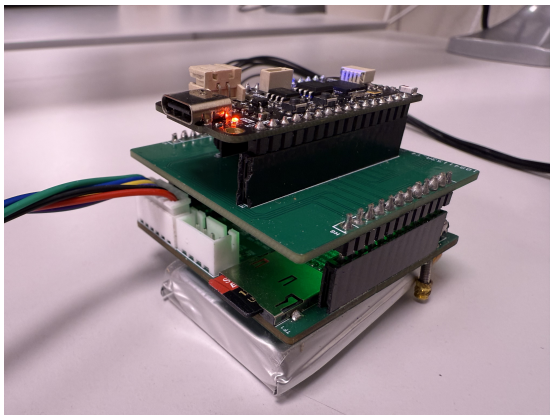
Special attention was paid to pin order during crimping, ensuring that the JST pinout matches the SPI signal positions on the expansion board and that the corresponding SPI lines align correctly with both the encoder connector and the custom external IMU module.



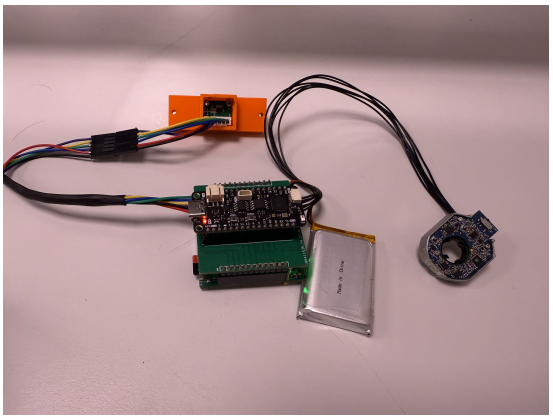
(a) JST crimp tool (WC-110). (b) Crimped JST terminal on AWG26–28 wire. (c) Assembled wiring harness and system view.

Figure 3.15: Cable preparation workflow for external sensors: (a) the JST WC-110 crimp tool used for AWG26–28 wires, (b) the crimped terminal after pressing and inspection, and (c) the assembled harness connected to the sensing system.

(3) Final integration. The expansion board, adapter boards, external IMU module, and encoder cable were assembled into a complete sensing subsystem, with cable routing adjusted to fit within the mechanical enclosure.



(a) Close-up view of the custom board assembly.



(b) Fully assembled sensing subsystem after interconnect assembly and wiring.

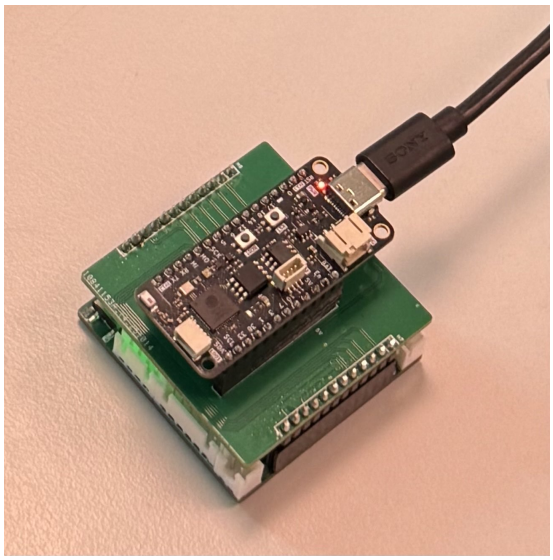
Figure 3.16: Final system integration after manual interconnect assembly and cable preparation.

3.4.4 Power-On Validation

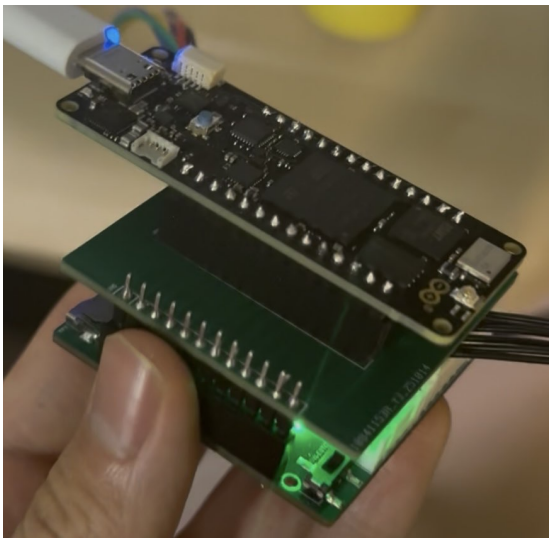
After assembly, the system was powered for the first time using a laboratory supply or USB power. The following checks were performed:

- verification that the 3.3 V LDO output was stable,
- verification of the 5 V boost converter when required by the Portenta H7,
- confirmation that overall current draw fell within the expected range.

Successful power validation confirmed that the power-distribution network and assembled interconnections were functioning correctly.



(a) ESP32 adapter powered via USB



(b) Arduino Portenta H7 powered via USB

Figure 3.17: Power-on validation of the system using USB power with different microcontroller platforms.

After assembly, the system was powered on for initial validation using USB power supplied through the microcontroller development boards. This test was intended to verify the integrity of the power-distribution network and basic system bring-up. Successful power-on was confirmed by the illumination of the onboard status LEDs on both the microcontroller platform and the expansion board. These observations indicate that the 3.3 V LDO regulator operated correctly and that no short circuits or assembly faults were present on the PCB.

3.4.5 Communication Interface Testing

With power validated, each communication interface was tested to ensure correct operation.

To make the communication interface testing repeatable and independent from system-level integration, a dedicated firmware-based validation workflow was implemented. The workflow includes pin mapping and SPI bus initialization, sensor communication handshake, device configuration, raw data acquisition checks, and SD card logging verification. The overall procedure is summarized in the interface test flowchart shown in Fig.3.18

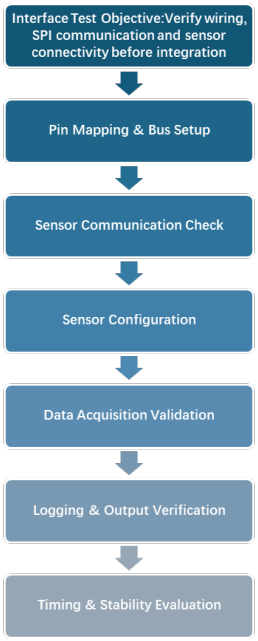


Figure 3.18: Interface test workflow. The workflow verifies pin mapping and SPI setup, confirms sensor connectivity through handshake checks, applies device configuration, validates data acquisition, and verifies SD logging consistency before evaluating timing stability.

(1) IMU communication testing

Both the onboard IMU and the external IMU module were verified through SPI communication tests.

In the validation firmware, each IMU was accessed through a shared SPI bus using independent chip-select signals. Communication was verified by reading the WHO_AM_I register, followed by basic configuration of accelerometer and gyroscope settings. Raw accelerometer and gyroscope outputs were then sampled to confirm that the sensor values were stable and responsive.

(2) Rotary encoder testing

The encoder interface was tested by rotating the joint and monitoring the angle output.

The encoder readings were streamed continuously in firmware, and the measured angle was verified to change smoothly without discontinuities during joint rotation.

(3) microSD card testing

The SPI-based microSD interface was tested by detecting the card, initializing the file system, and writing a short test log to confirm successful data storage.

During logging experiments, it was observed that SD card write operations can introduce non-deterministic latency, leading to occasional timing jitter when logging is performed directly inside the sampling loop.

To address this, a buffered logging strategy was later adopted in the battery-powered long-run test, where samples are stored in RAM and written to the SD card in periodic burst operations.

3.4.5.1 Validation results

Prior to platform comparison, the communication interfaces were first validated on the platform IO used in the proposed system. All required interfaces, including SPI communication with the IMUs, rotary encoder reading, and microSD card access, were verified to operate correctly and reliably under the intended operating conditions. Based on these results, the ESP32 Feather S3 was confirmed to fully support the required multi-sensor communication and data logging tasks.

However,after flashing the firmware, the Portenta H7 board consistently exhibited a persistent solid orange LED and remained unresponsive, even after reset. In this state, the bootloader could not be accessed.

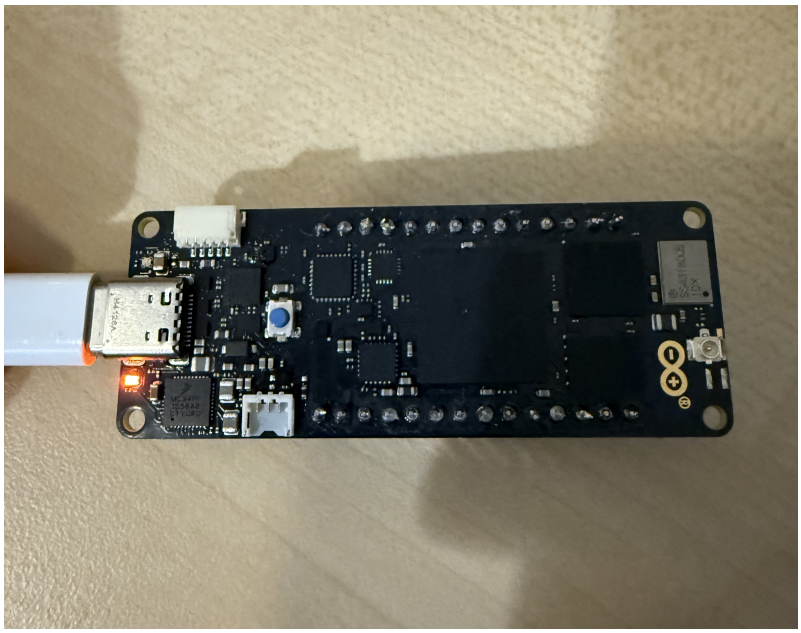


Figure 3.19: Observed solid orange LED state on the Arduino Portenta H7 after flashing, indicating an abnormal startup condition. In this state, the bootloader was unresponsive and external debugging access was unavailable.

3.4.6 Issue Encountered on Arduino Portenta H7

During the platform evaluation phase, the Arduino Portenta H7 was considered as an alternative MCU candidate. However, after repeated flashing attempts, the board eventually entered an abnormal startup state indicated by a persistent solid orange LED and became unresponsive. In this condition, the bootloader could not be accessed and normal upload procedures failed.

3.4.6.1 Bring-up and firmware upload timeline

To clarify how the issue developed over time, the firmware upload process is summarized chronologically below.

- **Week 1 — Basic bring-up validation (Blink program):** A minimal `Blink` program was uploaded and executed to verify basic board functionality and power delivery through the adaptable board and expansion board. At this stage, the board behaved normally and the blue LED blinked as expected.
- **Week 1 — Initial peripheral integration attempts:** After the `Blink` test, pin assignments were mapped and interface setup was started in order to follow a similar validation sequence as used on the ESP32 platform. During this phase, firmware upload attempts started to fail intermittently. Notably, the orange LED symptom had *not* appeared yet, and the primary issue observed was unsuccessful uploads.

- **Week 2 — Persistent upload failure and onset of orange LED state:** In the second week, firmware uploads continued to fail consistently and the board could no longer be flashed successfully. After continued unsuccessful flashing attempts, the board entered an abnormal state where the orange LED became persistent and the device became unresponsive. Once the orange LED condition appeared, recovery using standard reset/upload procedures was unsuccessful.

3.4.6.2 Likely causes reported by the community

Based on community reports and platform documentation, a solid or blinking orange LED on the Portenta H7 is commonly associated with bootloader- or power-management-related failures. A frequent interpretation is that the board has entered a "bricked"-like state in which the startup firmware responsible for initializing the Power Management IC (PMIC) is missing or corrupted. In that case, the PMIC is not configured correctly, internal power rails are not enabled, and the board cannot complete its normal boot sequence. In addition, some reports indicate that I^2C bus conflicts can also trigger the orange LED, particularly if external I^2C devices are connected to the same bus used by the PMIC on the breakout board.

3.4.6.3 Recommended checks (non-invasive)

Before attempting advanced recovery, the following checks are commonly recommended:

- **Reset/bootloader check:** perform a quick double-press of the reset button to test whether the bootloader is still accessible. If no bootloader behavior is observed, the board is likely in a boot-critical failure state.
- **Disconnect external I^2C devices:** remove any connected Qwiic/I^2C peripherals to rule out conflicts with the PMIC I^2C bus.

The recovery sequence typically includes:

1. Connect the required I^2C lines to access the PMIC configuration interface.
2. Attach an external debugger (e.g., ST-LINK) via SWD/JTAG.
3. Issue PMIC configuration commands over I^2C to restore the power rails.
4. Reflash the bootloader/firmware image once the board is powered correctly.

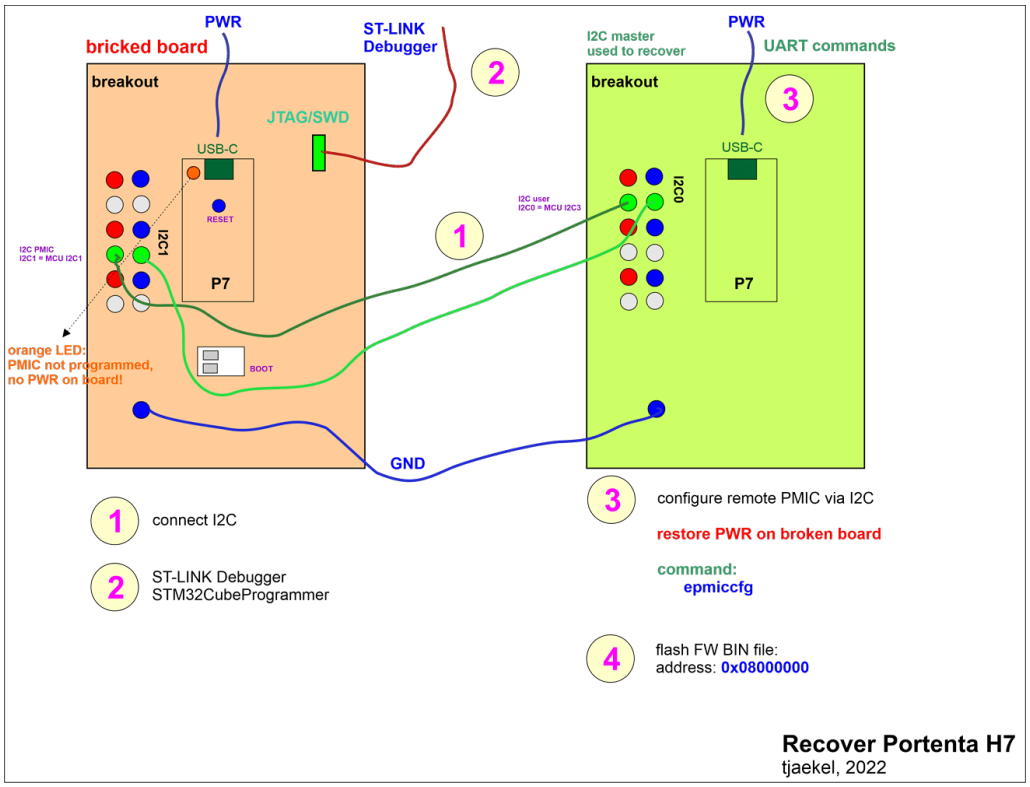


Figure 3.20: Example recovery setup for an Arduino Portenta H7 showing a typical approach for restoring a "bricked" board: I^2C access is used to reconfigure the PMIC, and an external debugger is used to reflash the firmware/bootloader image.

3.4.6.4 Impact on this project

Although the issue encountered with the Portenta H7 appears to be platform-specific rather than directly related to the proposed sensing hardware design, it has methodological implications for this project.

Recovering the Portenta H7 required dedicated recovery hardware and additional configuration steps (breakout access, I^2C PMIC configuration, and external debugging tools). Due to time constraints and the focus of this thesis on validating the custom sensing and logging architecture, the Portenta H7 platform was excluded from subsequent communication interface validation and experimental evaluation.

As a consequence, the experimental results presented in this thesis correspond exclusively to the implemented ESP32-S3 based system. While the sensing architecture is conceptually platform-agnostic at the design level, its practical performance characteristics — including timing determinism, interrupt handling behavior, DMA scheduling, and bus arbitration latency — are inherently dependent on the microcontroller architecture and firmware implementation.

Therefore, the synchronization stability and timing robustness demonstrated in later chapters cannot be automatically generalized to alternative hardware platforms without independent validation. The current work confirms reliable behavior for the implemented platform, but it does not establish universal robustness across

different MCU families.

Future work should include cross-platform validation to assess the portability and robustness of the custom sensing architecture under varying hardware and firmware environments.

4

Results

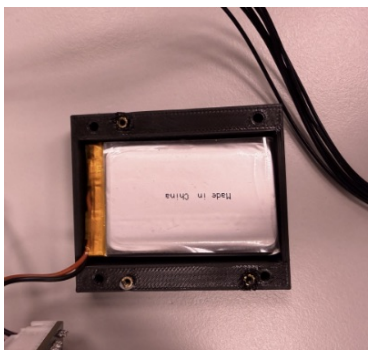
This chapter presents the experimental results obtained from the proposed sensing system. All experiments were conducted using multiple MCU platforms in order to evaluate system functionality, data acquisition capability, and classification performance.

Experimental validation was successfully completed using the ESP32-based platform and the Feather S3 adapter board. Initial validation was also attempted using the Arduino Portenta H7 platform; however, platform-specific flashing and power initialization issues prevented its use for experimental data collection. Unless otherwise stated, all results presented in this chapter were obtained using the ESP32-based platform and the Feather S3 adapter board.

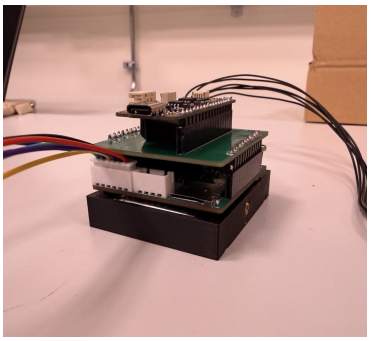
4.1 Prototype Assembly and System Integration

The electronic components were integrated into a custom-designed 3D-printed enclosure developed through an iterative design process.

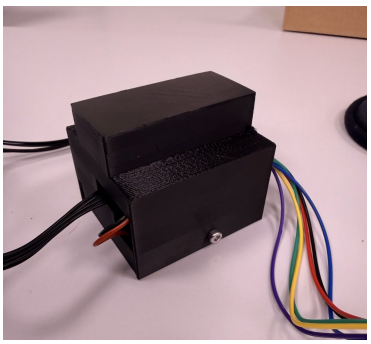
The final prototype consists of a dedicated battery enclosure, a stacked electronics assembly, and a fully enclosed housing. Figure(a f) illustrates the main mechanical components and the final assembled state of the prototype. The resulting system provides a compact and mechanically stable platform suitable for experimental data collection.



(a) Battery case (bottom).

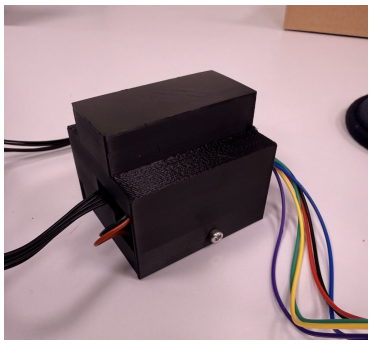


(b) Board stack mounted on the case base.

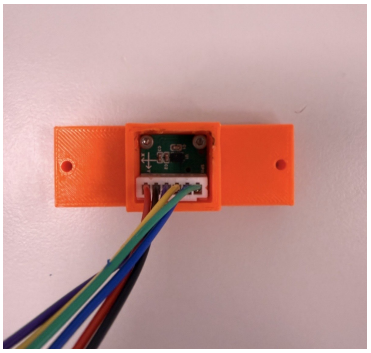


(c) Fully assembled enclosure.

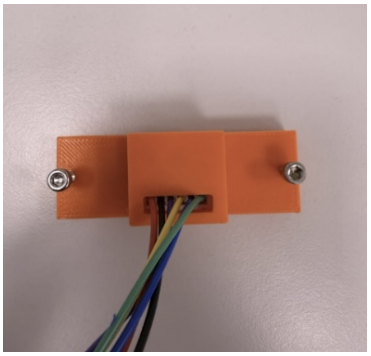
Figure 4.1: Mechanical enclosure components and final assembly of the electronics module.



(d) Fully assembled enclosure with electronics.



(e) Secondary IMU placed inside the enclosure.



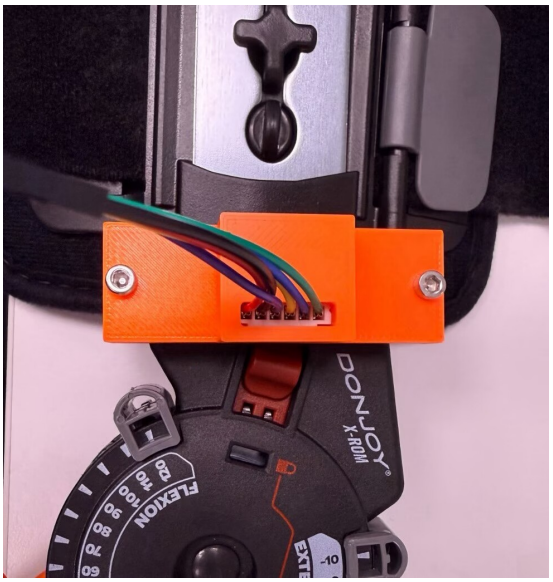
(f) IMU enclosure with click-lock lid assembled.

Figure 4.2: Integration and assembly of the secondary IMU and enclosure components.

4.1.1 Integration and Wiring Complexity Reduction

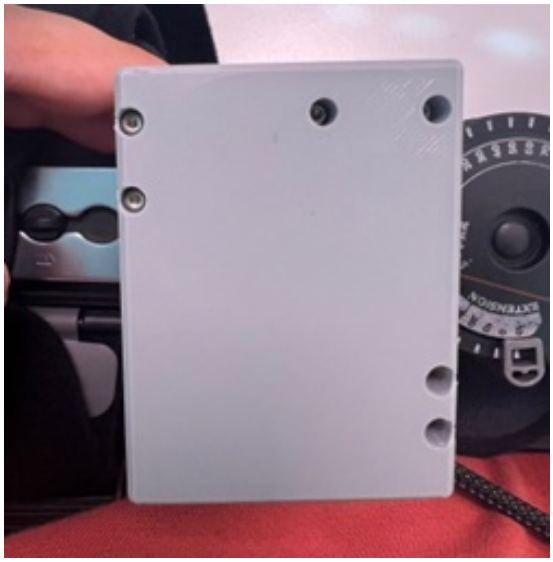
The proposed IMU and system enclosures are more compact than the previously used designs, as shown in Figures 4.3 and 4.4. The reduced dimensions simplify mechanical integration into the exoskeleton structure and enable shorter cable routing with fewer interconnections.

(a) Previously used IMU enclosure (28 mm × 45 mm).

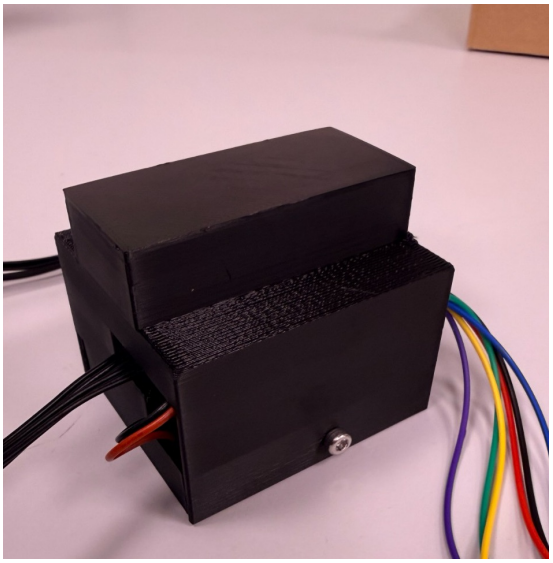


(b) Proposed custom IMU enclosure (19 mm × 19 mm).

Figure 4.3: Size comparison between the previously used IMU enclosure and the proposed custom IMU enclosure. The proposed design achieves a substantial reduction in footprint, enabling improved integration into the exoskeleton structure.



(a) Previously used exoskeleton system enclosure (55 mm × 72 mm × 53 mm).



(b) Proposed system enclosure (59 mm × 53 mm × 50 mm).

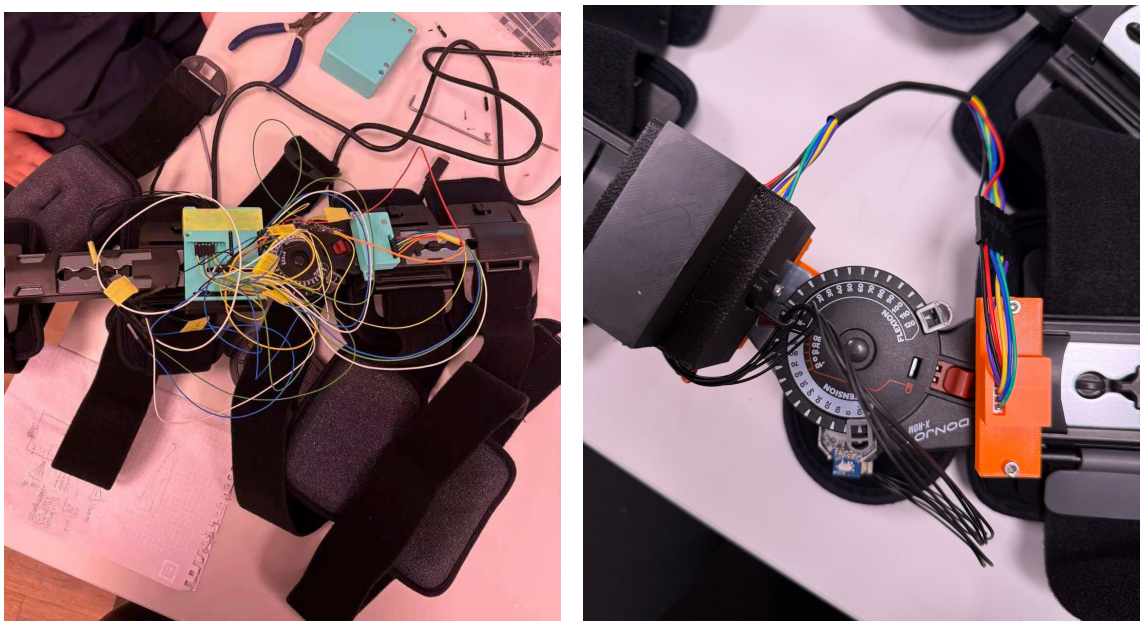
Figure 4.4: Size comparison between the previously used exoskeleton system enclosure and the proposed system enclosure. The proposed design achieves a more compact and integration-friendly form factor for wearable exoskeleton applications.

To experimentally validate improvements in integration level and usability, the redesigned system was compared with the original modular prototype in terms of electronics partitioning, external module dependency, and wiring complexity.

Table 4.1: Integration comparison between the original modular prototype and the redesigned prototype.

Metric	**Original prototype**	**Redesigned prototype**
Boards used in the system	MCU + 2 IMU breakout boards + microSD board	MCU expansion + adapter + customized sensor board+custom IMU board
External modules	Encoder + 2 IMUs	Encoder + 1 secondary IMU
SPI wiring (total)	24 wires	12 wires
Overall integration	Distributed modules with extensive wiring	Compact stacked architecture with reduced wiring

As summarized in Table 4.1, the original system relied on a modular architecture with four separate electronics boards and multiple external interfaces. This resulted in a relatively complex wiring setup, especially due to multiple SPI buses and additional power/ground connections routed between boards. In contrast, the redesigned system consolidates sensing and logging functions into a compact stacked architecture using an MCU expansion board, an adapter board, and a customized four-layer sensor board. As a result, the required SPI bus groups were reduced from four to two, which halves the amount of 12 SPI-related wires.



(a) Original modular prototype with multiple inter-board connections.

(b) Redesigned prototype with reduced wiring and improved integration.

Figure 4.5: Qualitative comparison of wiring complexity between the original modular prototype and the redesigned integrated prototype.

Figure 4.5 provides a qualitative illustration of the reduced wiring complexity enabled by the redesign. Fewer external connections simplify assembly and debugging, improve repeatability during experiments, and reduce the risk of intermittent failures caused by cable movement or connector instability. Therefore, the integration improvement is not only reflected in physical compactness, but also contributes to higher usability and reduced connection-related variability in long-term data collection.

4.2 System Stability Evaluation

System stability was evaluated under continuous operation with simultaneous acquisition from both IMUs, the rotary encoder, and periodic data logging to the microSD card. The system was operated for extended periods to assess communication reliability and runtime robustness.

In this work, system stability is evaluated from three complementary aspects: (i) data continuity and loss, (ii) signal quality and statistical consistency, and (iii) inter-sensor synchronization. These aspects directly reflect whether the proposed system can support reliable long-term and multi-sensor measurements.

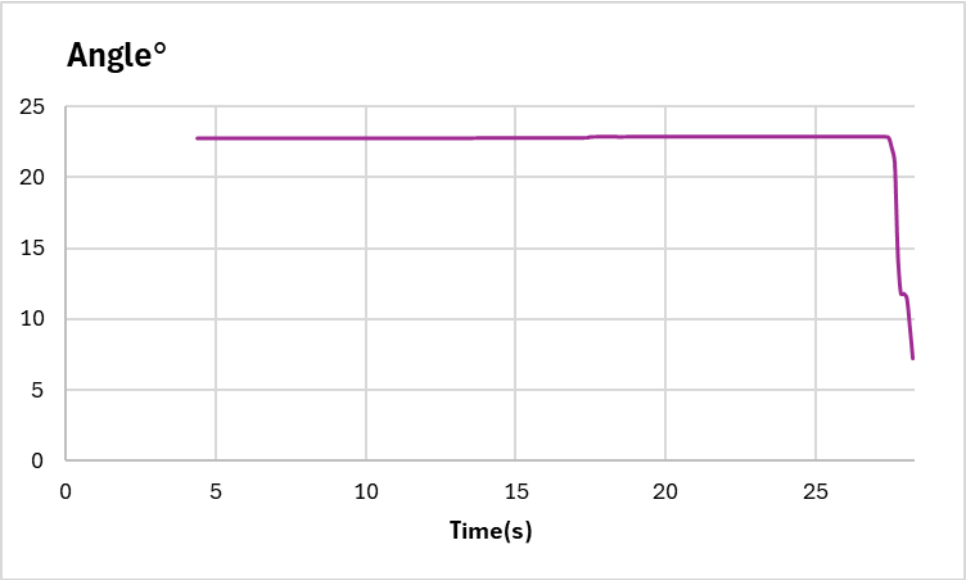


Figure 4.6: Recorded encoder angle under static condition.

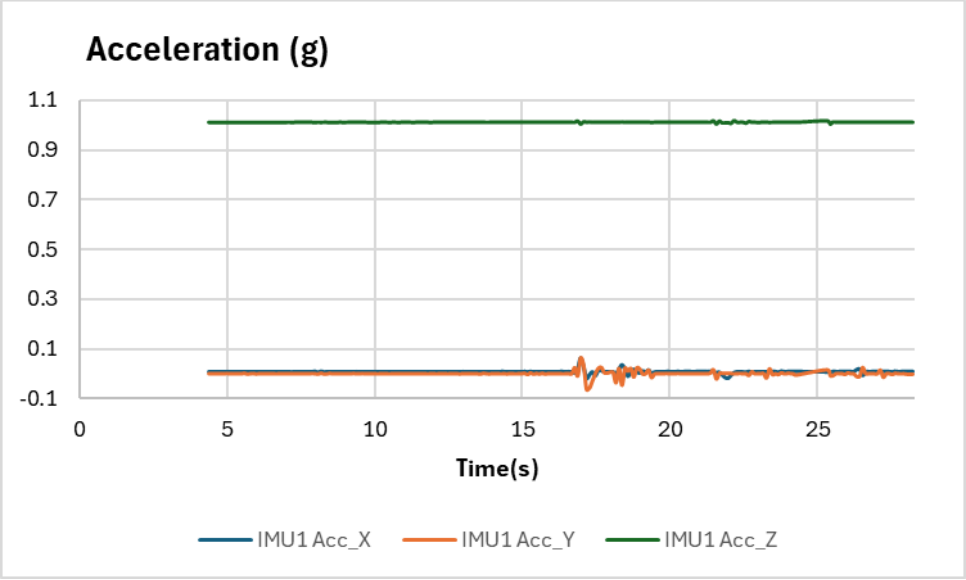


Figure 4.7: Tri-axial acceleration measured by IMU2 during continuous multi-sensor acquisition.

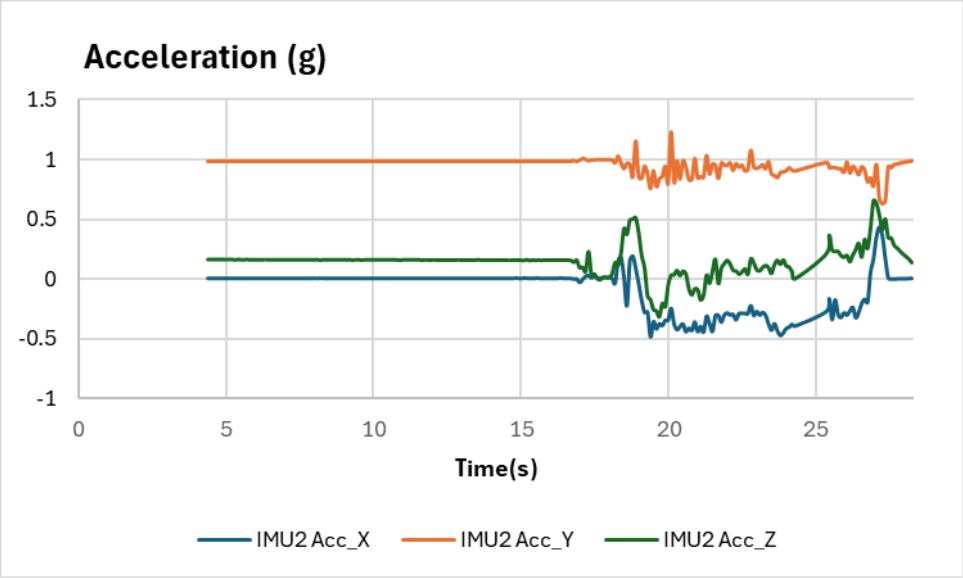


Figure 4.8: Tri-axial acceleration measured by IMU1 during continuous multi-sensor acquisition.

Figures 4.5–4.7 mainly serve as a functional validation of the multi-sensor acquisition and microSD logging pipeline. The recorded encoder and IMU signals confirm that the system can continuously sample data from multiple sensors and store it to the microSD card without interruptions. A more detailed evaluation of measurement stability and noise characteristics is presented in the next section.

4.2.1 Noise level under different sampling configurations

To quantify measurement dispersion under nominally static conditions, the sensors were kept fixed on the test setup without any intentional motion, while the system operated in continuous dual-IMU acquisition and microSD logging mode.

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} a_{x,i}. \tag{4.1}$$

The sample standard deviation is defined as

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(a_{x,i}-\mu_x)^2}, \tag{4.2}$$

and similarly for the y- and z-axes. In this static test, the true acceleration is expected to remain approximately constant. Therefore, σ serves as a practical proxy for the dispersion/noise level and enables a direct comparison between sampling configurations.

For IMU1, the measured standard deviations (in g) were approximately the following.

$$\text{10 Hz:} \quad \sigma_{AX} \approx 0.065,\ \sigma_{AY} \approx 0.070,\ \sigma_{AZ} \approx 0.004. \tag{4.3}$$

$$\text{1000 Hz:} \quad \sigma_{AX} \approx 0.027,\ \sigma_{AY} \approx 0.038,\ \sigma_{AZ} \approx 0.035. \tag{4.4}$$

As an example, the change on the x-axis can be written as

$$\Delta\sigma_{AX} = 0.065 - 0.027 = 0.038 \text{ g}, \tag{4.5}$$

which corresponds to an approximate relative reduction of

$$\frac{0.038}{0.065} \times 100\% \approx 58\%. \tag{4.6}$$

For IMU2, the measured standard deviations (in g) were approximately the following.

$$\text{10 Hz:} \quad \sigma_{AX} \approx 0.150,\ \sigma_{AY} \approx 0.100,\ \sigma_{AZ} \approx 0.147. \tag{4.7}$$

$$\text{1000 Hz:} \quad \sigma_{AX} \approx 0.030,\ \sigma_{AY} \approx 0.050,\ \sigma_{AZ} \approx 0.035. \tag{4.8}$$

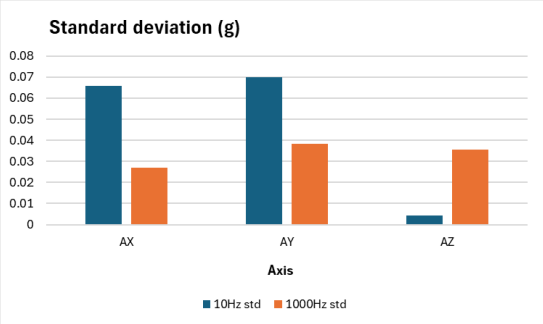


Figure 4.9: IMU1: standard deviation of acceleration (10 Hz vs. 1000 Hz).

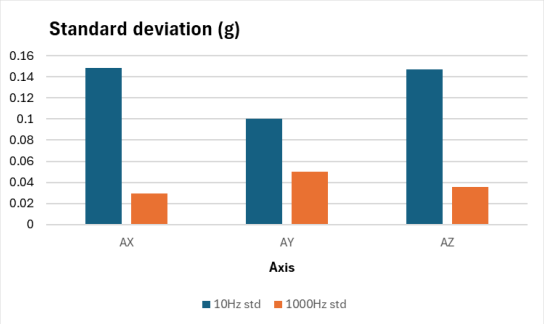


Figure 4.10: IMU2: standard deviation of acceleration (10 Hz vs. 1000 Hz).

Compared to the 10 Hz configuration, a consistently lower standard deviation is observed for the 1000 Hz configuration on both IMUs and along all axes. This indicates a reduced short-term dispersion of the measured acceleration signals under static conditions.

This improvement is mainly attributed to two factors. First, the higher sampling rate allows the sensor noise to be sampled more densely, which reduces the influence of quantization and short-term timing irregularities on the estimated variance. Second, the higher-rate acquisition provides a larger number of samples within the same observation interval, leading to a more stable estimate of the signal statistics.

It should be noted that the 10 Hz and 1000 Hz results are obtained from separate recordings under nominally identical static conditions, rather than from a resampled version of the same dataset. Therefore, the comparison reflects the combined effect of sampling configuration and system operating conditions, including processing load and logging behaviour.

Importantly, the reduced standard deviation at 1000 Hz should be interpreted as an improvement in measurement dispersion and short-term noise characteristics. It does not, by itself, imply improved long-term system stability or the absence of timing interruptions, which are evaluated separately in the timing stability analysis.

4.2.2 Acceleration magnitude distribution

To complement the axis-wise standard deviation comparison, the acceleration magnitude was analyzed as an orientation-independent indicator of static measurement stability. The magnitude was computed from the tri-axial accelerometer outputs as

$$a_{\mathrm{mag}}(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)}. \tag{4.9}$$

where $a_x(t)$, $a_y(t)$ and $a_z(t)$ denote the accelerometer measurements along the three sensor axes, and $a_{\mathrm{mag}}(t)$ represents the Euclidean norm of the acceleration vector.

The acceleration magnitude a_{mag} represents the Euclidean norm of the tri-axial accelerometer output and provides an orientation-independent representation of the measured acceleration vector. Under static conditions, a_{mag} is expected to remain approximately constant and close to the gravitational acceleration $1\,g$. Therefore, its distribution reflects the combined noise and consistency of the three axes without being affected by sensor orientation.

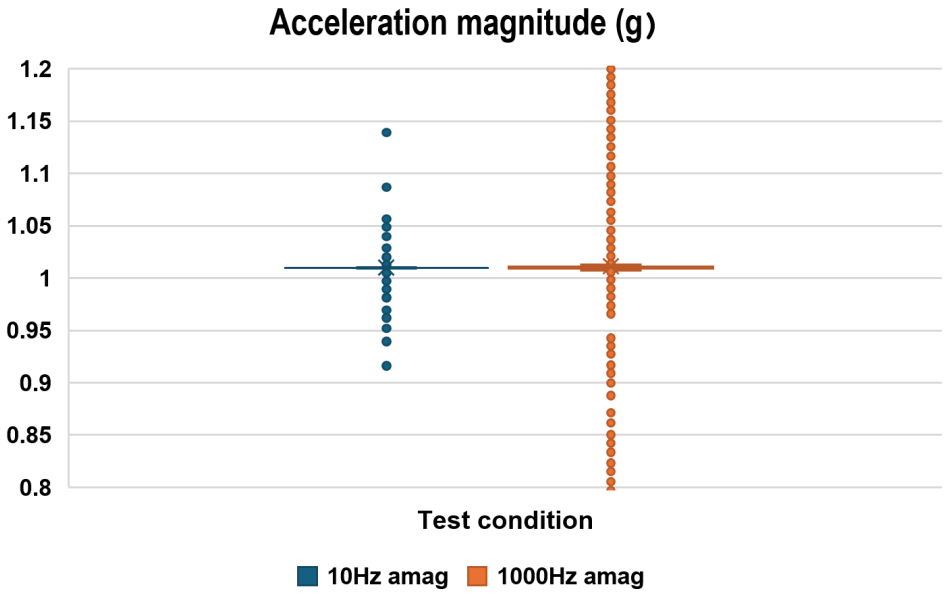


Figure 4.11: Distribution of acceleration magnitude under different test conditions.

Under static conditions, the acceleration magnitude is expected to remain close to the gravitational acceleration, i.e.

$$a_{\mathrm{mag}} \approx 1\,g.$$

Therefore, the distribution of a_{mag} provides an orientation-independent indicator of measurement consistency and sensor noise.

Figures4.11 shows the acceleration magnitude distributions obtained at 10 Hz and 1000 Hz, respectively. In both configurations, the mean magnitude remains close to $1\,g$, indicating correct sensor operation and consistent tri-axial measurements. The observed spread around the nominal value mainly reflects sensor noise and quantization effects.

It should be noted that the 10 Hz and 1000 Hz data were recorded in separate measurement runs and do not originate from the same underlying raw signal. The larger dispersion observed in the 1000 Hz case is mainly attributed to the higher sampling bandwidth, which captures additional high-frequency noise components.

Importantly, this analysis characterizes the quality and consistency of the acceleration measurements under static conditions. It does not provide information about sampling-time stability, data loss, or timing jitter, which are evaluated separately using the timestamp-based metrics in Section 4.3.

4.2.3 Inter-IMU synchronization and consistency

In addition to intra-sensor stability, system-level reliability for a dual-IMU platform requires that IMU1 and IMU2 remain synchronized and mutually consistent during simultaneous acquisition. This was evaluated using the axis-wise acceleration difference between the two IMUs:

The inter-IMU synchronization and consistency are evaluated using the axis-wise acceleration differences between IMU1 and IMU2.

$$\Delta a_k(t) = a_k^{\mathrm{IMU1}}(t) - a_k^{\mathrm{IMU2}}(t), \qquad k \in \{x, y, z\}. \tag{4.10}$$

A stable and bounded distribution of $\Delta a_k(t)$ indicates consistent sampling timing and comparable sensor behaviour between the two IMUs.

$$\begin{aligned} \Delta a_x(t) &= a_{x,\mathrm{IMU1}}(t) - a_{x,\mathrm{IMU2}}(t), \\ \Delta a_y(t) &= a_{y,\mathrm{IMU1}}(t) - a_{y,\mathrm{IMU2}}(t), \\ \Delta a_z(t) &= a_{z,\mathrm{IMU1}}(t) - a_{z,\mathrm{IMU2}}(t). \end{aligned} \tag{4.11}$$

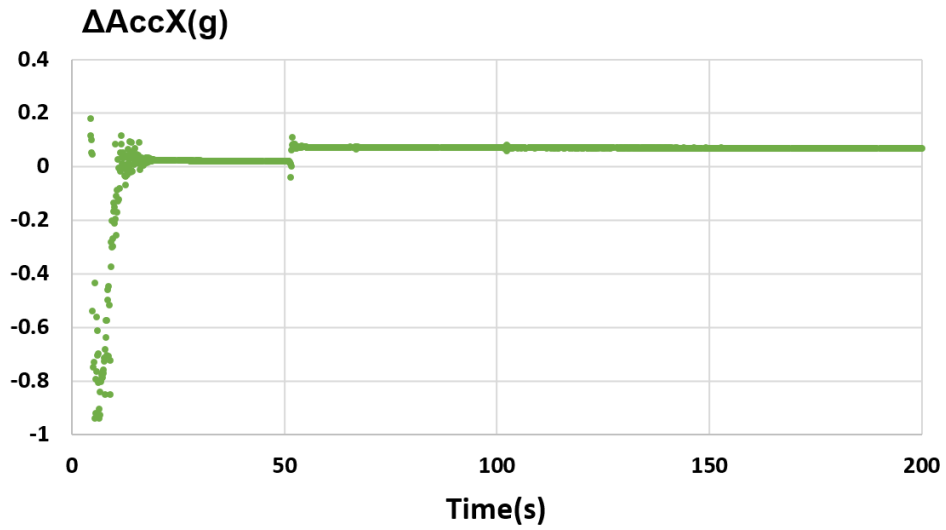


Figure 4.12: Time evolution of acceleration deviation on the X axis, $\Delta AccX$ (g).

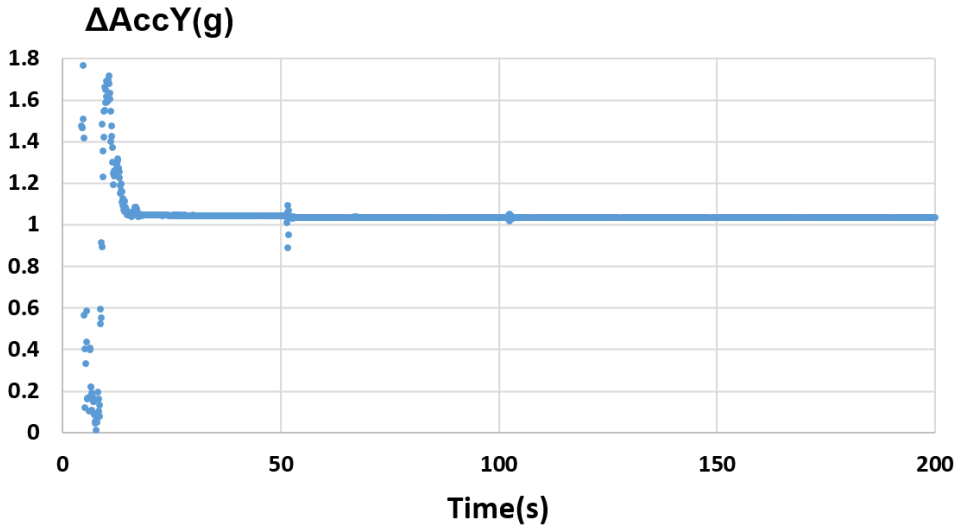


Figure 4.13: Time evolution of acceleration deviation on the Y axis, $\Delta AccY$ (g).

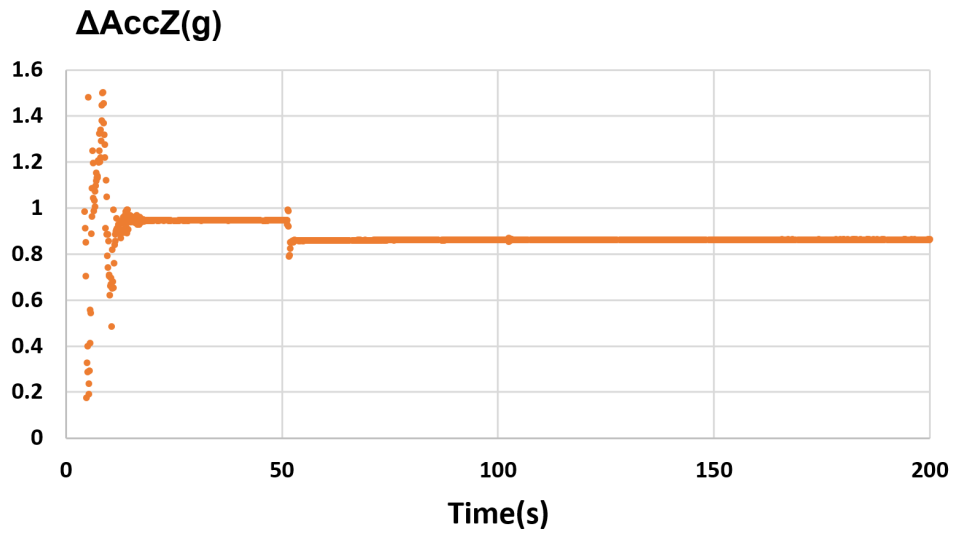


Figure 4.14: Time evolution of acceleration deviation on the Z axis, $\Delta AccZ$ (g).

Figures 4.12, 4.13, and 4.14 show the time evolution of the inter-IMU acceleration differences Δa_x, Δa_y, and Δa_z during a stationary experiment.

During the test, the system operated in continuous dual-IMU acquisition and logging mode, while data were streamed to the storage subsystem under the same software configuration as used in the subsequent noise and timing evaluations. No intentional motion was applied to the sensors, and both IMUs remained fixed in a static configuration throughout the measurement.

For a stationary test under constant operating conditions, a synchronized acquisition pipeline is expected to produce inter-sensor differences that remain bounded and approximately constant over time. As shown in the figures, a short transient phase is observed after system start-up, after which the Δa signals converge to steady levels with limited fluctuations and without visible drift, burst-like deviations, or discontinuities. This indicates that the two IMUs are sampled and logged in a temporally consistent manner during continuous operation.

Importantly, the observed non-zero steady offsets in Δa do not imply synchronization errors. They can be attributed to constant sensor-to-sensor bias, small axis misalignment, and mounting orientation differences between the two IMUs. The relevant synchronization criterion in this experiment is the temporal stability of the inter-IMU differences rather than their absolute magnitude. The absence of long-term drift and abrupt changes under continuous acquisition and logging load therefore supports stable inter-IMU synchronization during runtime operation.

4.2.4 Synchronization between IMUs and encoder

To evaluate the temporal synchronization performance of the implemented ESP32 feather S3 based sensing system under dynamic operating conditions, a motion-based validation experiment was conducted using the joint encoder as an external

reference sensor.

The experimental setup is illustrated in Figure 4.15. The wearable orthosis equipped with the sensing system was worn by the subject, and cyclic knee flexion–extension motion was manually performed. The joint angle was repeatedly moved between approximately 0° and 100°, generating a periodic back-and-forth trajectory representative of typical rehabilitation movement patterns.

Figure 4.15: Experimental setup for synchronization validation during cyclic knee motion.

All sensing channels, including the encoder and the two IMUs, were acquired within the same synchronized firmware loop and recorded using a common timestamp generated by the ESP32-S3 microcontroller. The effective sampling frequency during this experiment was 250 Hz, corresponding to a nominal sampling interval of 4 ms.

Motion Signal Consistency

Figure 4.16 presents the encoder angle trajectory together with the angular velocity derived from the encoder and the z-axis gyroscope signals from both IMUs.

The encoder angle demonstrates a stable periodic trajectory between 0° and 100°, confirming consistent excitation conditions across cycles. The encoder-derived angular velocity exhibits clear direction reversals and peak magnitudes at flexion and extension transitions.

The gyroscope signals of both IMUs show strong temporal alignment with the encoder velocity profile. Peak angular velocities occur at consistent time instances relative to the encoder extrema, and zero-crossings correspond closely to motion direction changes. Across multiple cycles, no visible phase drift between the encoder and IMU signals is observed, indicating stable synchronization throughout the experiment.

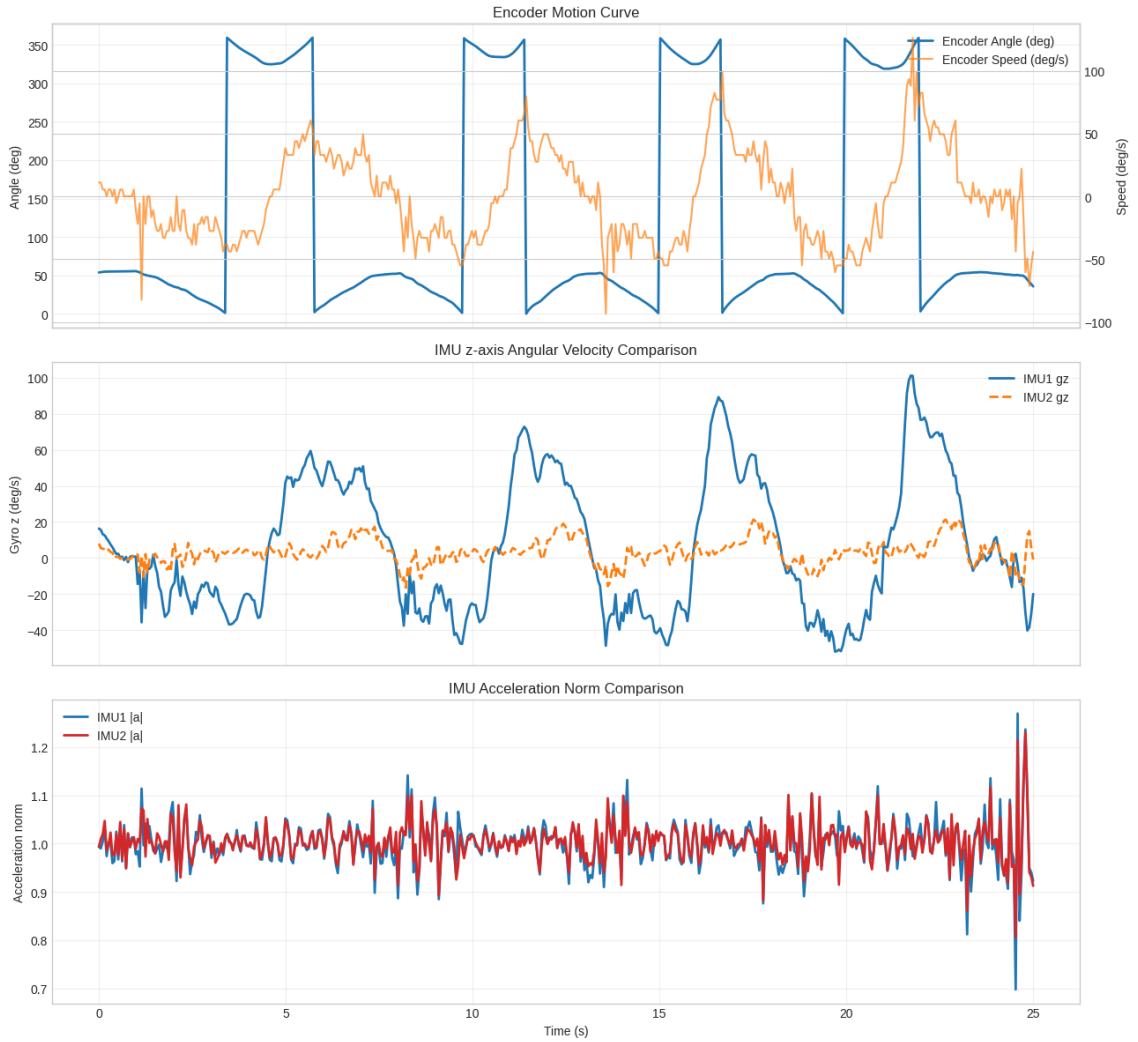


Figure 4.16: Encoder angle, encoder angular velocity, and IMU gyroscope signals during cyclic motion.

Jitter and Inter-Sensor Delay Analysis

Figure 4.17 summarizes timing stability metrics across the experiment.

The sampling period remains tightly clustered around the nominal 4000 μs. The average sampling interval shows minimal deviation, confirming that the ESP32-S3 acquisition loop maintains deterministic timing under dynamic motion conditions.

Occasional jitter peaks are observed in the maximum jitter trace. These peaks are isolated and do not propagate across consecutive cycles, indicating that they are transient scheduling effects rather than systematic timing drift. The inter-sensor

delay analysis reveals: - Encoder $\rightarrow$ IMU1 delay remains stable around approxi-

mately 41–43 μs - IMU1 $\rightarrow$ IMU2 delay remains stable around approximately 48–50 μs - Encoder $\rightarrow$ IMU2 delay remains centered near 90 μs Importantly, these delay

values remain constant across the entire motion duration. No increasing offset or cumulative time skew is detected. This confirms that the synchronized acquisition loop maintains fixed ordering and deterministic execution timing.

Delay Distribution Characteristics

The high-resolution delay histograms (Figure 4.18) provide quantitative validation of timing consistency.

The encoder-to-IMU1 delay exhibits a mean value of approximately 41.18 μs with a standard deviation of 1.74 μs. The IMU1-to-IMU2 delay shows a mean of 48.59 μs with a standard deviation of 2.37 μs. The encoder-to-IMU2 delay remains centered at 89.77μswith a standard deviation of 4.02 μs.

The distributions are well approximated by a Gaussian profile, with no evidence of multimodal behavior or heavy tails. The relatively small standard deviations compared to the mean delay values indicate tightly bounded timing dispersion.

The cycle jitter histogram further confirms that the sampling interval deviation remains centered close to zero (6.09 μs), without systematic offset accumulation. Although occasional extreme jitter events are observed in time-domain traces, the statistical distribution demonstrates that the majority of samples remain concentrated within a narrow temporal band.

These results statistically confirm deterministic execution order and bounded timing variability of the synchronized acquisition loop.

Synchronization Performance Summary

Under cyclic knee motion between 0° and 100° at 250 Hz sampling frequency, the implemented ESP32-Feather S3 based sensing system demonstrates statistically bounded and deterministic timing behavior.

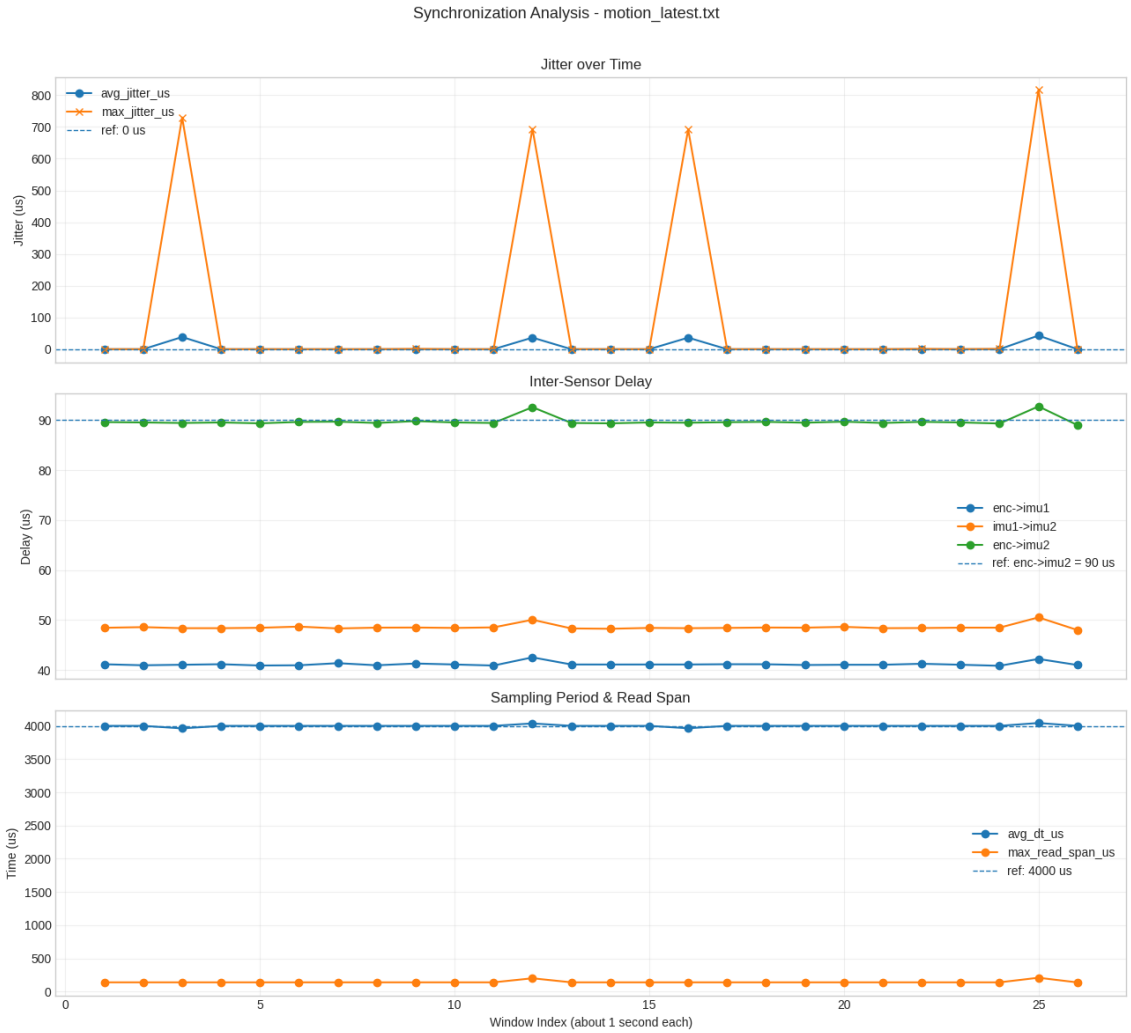


Figure 4.17: Sampling period stability, jitter behavior, and inter-sensor delay during motion experiment at 250 Hz.

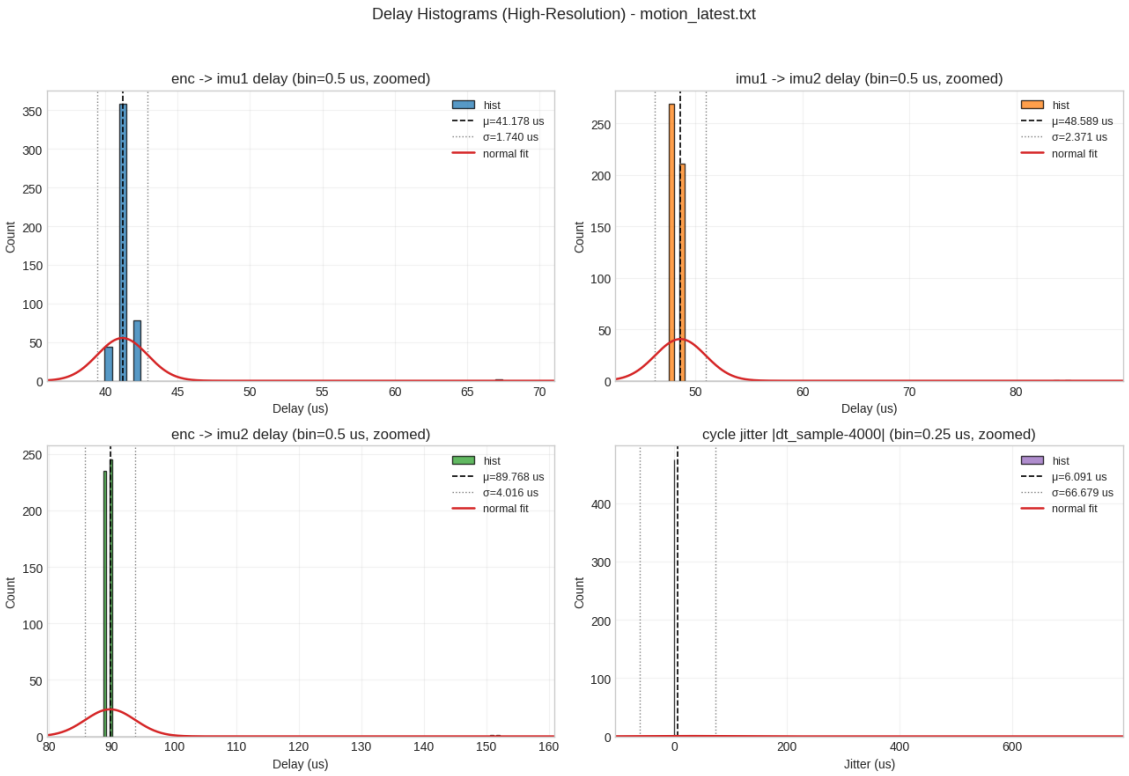


Figure 4.18: Delay histograms and sampling interval distribution during synchronization experiment.

- Stable sampling period centered at 4000 μs with mean jitter $\mu \approx 6.09$ μs and no cumulative drift
- Encoder $\rightarrow$ IMU1 delay: $\mu = 41.18$ μs, $\sigma = 1.74$ μs
- IMU1 $\rightarrow$ IMU2 delay: $\mu = 48.59$ μs, $\sigma = 2.37$ μs
- Encoder $\rightarrow$ IMU2 delay: $\mu = 89.77$ μs, $\sigma = 4.02$ μs
- Approximately Gaussian delay distributions without multimodal behavior or long-tailed dispersion

These results confirm bounded timing variability and fixed execution ordering of the synchronized acquisition loop under dynamic operating conditions representative of practical wearable use.

4.3 Battery Life Evaluation

To assess the endurance of the system during untethered operation, a series of battery-life experiments were conducted using a 2000 mAh lithium-ion battery. The experiments corresponded to different sampling and logging configurations.

The battery life evaluation followed a consistent workflow to ensure that both runtime and long-run sampling stability were assessed under comparable conditions. Figure 4.19 summarizes the overall test procedure, from battery power setup and continuous sensor acquisition to buffered microSD burst logging and timing stability monitoring.

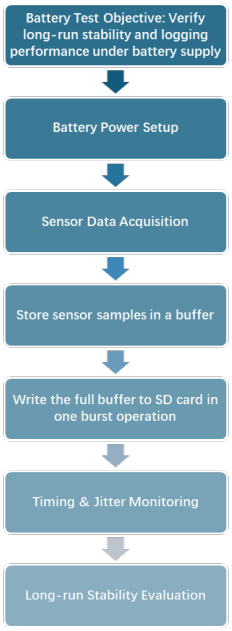


Figure 4.19: Battery test workflow for evaluating long-run stability and logging performance under battery supply.

Based on this workflow, three operating configurations were evaluated to quantify how sampling frequency and microSD logging affect the achievable continuous runtime and timing stability. The tested configurations are described in the following paragraphs.

4.3.1 Experimental Setup

The system was powered solely by the battery throughout each experiment. Sensor data were acquired at a fixed sampling rate and stored into a RAM buffer before being written to the microSD card in burst transfers. A buffer size of $N = 200$ samples was used in all configurations. Under this buffering scheme, the microSD write interval is given by

$$T_{\text{write}} = \frac{N}{f_s}, \tag{4.12}$$

where f_s denotes the sensor sampling frequency. This design reduces frequent small writes and instead concentrates storage activity into periodic burst events.

4.3.2 Test Configurations and Results

Three operating configurations were evaluated:

(1) High-frequency mode (1000 Hz sensor communication)
In the high-frequency configuration, the MCU sampled the sensors at $f_s = 1000$ Hz, corresponding to a target sampling period of

$$T_{\text{target}} = \frac{1}{f_s} = 1.000 \text{ ms}. \tag{4.13}$$

With $N = 200$ samples per buffer, a burst write to the microSD card occurred every

$$T_{\text{write}} = \frac{200}{1000} = 0.2 \text{ s}. \tag{4.14}$$

Under this configuration, the system achieved a continuous runtime of:

10 hours 44 minutes

(2) Low-frequency mode (10 Hz sensor communication)
In the low-frequency configuration, the sensor sampling rate was reduced to $f_s = 10$ Hz, resulting in a target sampling period of

$$T_{\text{target}} = \frac{1}{10} = 100 \text{ ms}. \tag{4.15}$$

Using the same buffer size of $N = 200$ samples, the microSD write interval became

$$T_{\text{write}} = \frac{200}{10} = 20 \text{ s}, \tag{4.16}$$

which reduces microSD activity by a factor of 100× compared to the 1000 Hz mode. Under this configuration, the system achieved a continuous runtime of:

13 hours 18 minutes

(3) Low-frequency mode with limited data logging
To further isolate the impact of microSD write operations on power consumption, an additional experiment was performed at $f_s = 10$ Hz where microSD logging was enabled only during the first three hours of operation and disabled thereafter, while sensor sampling and internal communication remained active. This approach provides an estimate of the energy overhead associated with sustained storage writes. Under this condition, the system achieved a total continuous runtime of:

13 hours 34 minutes

4.3.3 Timing stability metrics

To quantitatively characterize the timing behaviour of the acquisition and logging pipeline under different operating modes, the sampling timestamps were recorded and analysed. Rather than directly claiming timing stability, the following metrics are used to describe the average sampling behaviour and the occurrence of occasional timing disturbances caused by system load and microSD write operations.

For each mode, the instantaneous sampling period is computed from consecutive timestamps as

$$T_i = t_{i+1} - t_i.$$

From the sequence of sampling periods, the peak-to-peak timing jitter is defined as

$$J_{\mathrm{pp}} = T_{\max} - T_{\min}, \tag{4.16}$$

where $T_{\min}$ and $T_{\max}$ denote the minimum and maximum observed sampling intervals, respectively. In addition, the standard deviation of the sampling period, σ_T, is computed to describe the dispersion of the sampling intervals around their mean value.

These metrics are intended to capture both the average sampling behaviour and the influence of rare blocking events, which are mainly introduced by burst microSD write operations and shared system resources. The extracted timing statistics for all tested configurations are summarized in Table 4.2.

Table 4.2: Timing stability metrics under different battery test configurations.

Mode	$\bar{T}$ (ms)	σ_T (ms)	$T_{\min}$ (ms)	$T_{\max}$ (ms)	J_{pp} (ms)
1000 Hz logging (0.2 s burst write)	1.587	8.469	0.0	229.0	229.0
10 Hz logging (20 s write period)	103.019	44.013	100.0	765.0	665.0
10 Hz limited logging	103.321	46.677	93.0	1164.0	1071.0

Table 4.2 summarizes the timing statistics of the sampling periods extracted from the logged timestamps. For all configurations, the mean sampling period $\bar{T}$ remains close to the nominal target value (1 ms for 1000 Hz and 100 ms for 10 Hz), indicating that the scheduler maintains the intended average sampling rate.

However, the standard deviation σ_T is noticeably larger than the deviation of the mean from the nominal period. This behaviour is caused by a small number of rare but long sampling intervals. Although the majority of sampling periods are clustered close to the target value, occasional blocking events significantly increase the dispersion of the distribution. As a result, σ_T is dominated by these outliers and therefore does not reflect the typical short-term sampling behaviour.

This effect is also visible in the maximum sampling interval $T_{\max}$ and in the peak-to-peak jitter J_{pp}. For the two 10 Hz configurations, isolated long gaps of up to 765 ms and 1164 ms are observed. These events are attributed to temporary blocking of the acquisition task during microSD write operations and shared system resource usage. Consequently, the limited-logging mode exhibits the largest worst-case stall.

In the 1000 Hz configuration, the mean sampling period increases to $\bar{T} = 1.587$ ms, which reflects the increased execution and logging overhead at high sampling rates. The worst-case interval of 229 ms remains substantially smaller than the worst-case gaps observed at 10 Hz, but still indicates the presence of system-level blocking events.

The reported minimum sampling interval $T_{\min}$ includes values close to zero. These values originate from consecutive timestamps with identical or near-identical time

stamps caused by the limited timer resolution and timestamp quantization, rather than by an actual physical zero sampling interval.

Importantly, no missing samples or corrupted records were observed in the logged data. All samples are present in the log files and the observed long intervals correspond to delayed acquisitions rather than data loss. Therefore, the results indicate that the proposed system reliably records all samples, while rare blocking events introduce timing irregularities in the acquisition and logging pipeline.

4.3.4 Discussion

Across the three experiments, reducing the sampling rate and decreasing the frequency of microSD write bursts resulted in a noticeable increase in continuous runtime. The improvement from 10 hours 44 minutes (1000 Hz logging) to 13 hours 18 minutes (10 Hz logging) indicates that the combined overhead of high-rate communication and frequent storage activity contributes significantly to average power draw. Furthermore, the limited-logging experiment achieved the longest runtime of 13 hours 34 minutes, supporting the conclusion that microSD write operations impose measurable additional power consumption even at low sampling rates. Overall, the battery tests demonstrate that the system can operate continuously for approximately 10–13 hours depending on the sensor sampling frequency and the applied data logging strategy.

4.3.5 Power Budget Analysis and Loss Breakdown

To interpret the measured battery runtime results, a module-level power budget was derived from the expansion board schematic and relevant component datasheets. The goal of this analysis is to identify dominant power consumers, quantify secondary losses, and explain the gap between estimated peripheral loads and the measured total system current.

4.3.5.1 Power distribution architecture.

The system is powered from a single-cell Li-ion battery with a nominal voltage $V_{\text{bat}} \approx 3.7$ V. The ESP32 Feather S3 microcontroller is supplied directly from the battery rail (`BAT_V`). A boost converter generates a regulated 5 V rail, which powers the rotary encoder and other 5 V loads. The 5 V rail is further converted to 3.3 V via an LDO, supplying the IMUs, microSD card, and logic pull-up networks.

The boost converter output voltage is configured by a feedback divider ($R_{\text{upper}} = 73.1$ kΩ, $R_{\text{lower}} = 10$ kΩ). Using a feedback reference voltage $V_{\text{FB}} = 0.6$ V, the expected boost output voltage becomes:

$$V_{out} = V_{\text{FB}} \left(1 + \frac{R_{\text{upper}}}{R_{\text{lower}}}\right) \approx 0.6 \left(1 + \frac{73.1}{10}\right) \approx 5.0 \text{ V}. \tag{4.17}$$

This confirms that the peripheral supply rail is nominally 5 V.

4.3.5.2 Battery-equivalent current estimation.

To compare individual module contributions with the measured system current, each load is expressed as an equivalent battery-side current. For a load drawing power P_{load} supplied through the boost converter, the corresponding battery current is approximated as:

$$I_{\text{bat}} \approx \frac{P_{\text{load}}}{V_{\text{bat}} \cdot \eta_{\text{boost}}}, \tag{4.18}$$

where η_{boost} is the boost conversion efficiency. In this work, a typical efficiency of $\eta_{\text{boost}} = 0.9$ is assumed for engineering-level estimation.

4.3.5.3 Major peripheral loads.

Rotary encoder (AMT222B-V, 5 V rail). The AMT222B-V rotary encoder is powered from the 5 V rail. Using a typical current consumption of $I_{\text{enc}} \approx 16$ mA, the encoder power draw is:

$$P_{\text{enc}} = 5 \text{ V} \cdot 16 \text{ mA} = 80 \text{ mW}. \tag{4.19}$$

This corresponds to an estimated battery-equivalent current of:

$$I_{\text{bat,enc}} \approx \frac{80 \text{ mW}}{3.7 \text{ V} \cdot 0.9} \approx 24 \text{ mA}. \tag{4.20}$$

Therefore, the encoder represents one of the largest fixed peripheral loads in the system.

Power indicator LED (5 V rail, always enabled). A 5 V power indicator LED is connected through a 1 kΩ series resistor. Assuming a typical LED forward voltage $V_f \approx 2.0$ V, the LED current is approximately:

$$I_{\text{LED}} \approx \frac{5 - V_f}{1 \text{ k}\Omega} \approx 3 \text{ mA}. \tag{4.21}$$

The corresponding battery-equivalent current becomes:

$$I_{\text{bat,LED}} \approx \frac{5 \text{ V} \cdot 3 \text{ mA}}{3.7 \text{ V} \cdot 0.9} \approx 4.5 \text{ mA}. \tag{4.22}$$

Although the LED is required for operational feedback, its continuous draw is non-negligible compared to other peripheral loads.

Dual IMU sensing (2× ICM-42688-P, 3.3 V rail). Two ICM-42688-P IMUs are powered from the 3.3 V domain. Assuming a typical 6-axis current of 0.88 mA per IMU, the combined IMU current is:

$$I_{\text{IMU,3V3}} \approx 2 \times 0.88 \text{ mA} = 1.76 \text{ mA}. \tag{4.23}$$

The IMU output power is:

$$P_{\text{IMU,out}} = 3.3 \text{ V} \cdot 1.76 \text{ mA} \approx 5.8 \text{ mW}. \tag{4.24}$$

Since the 3.3 V rail is generated by a 5 V→3.3 V LDO, the LDO input power is approximately:

$$P_{\text{IMU,in}} \approx 5\ \text{V} \cdot 1.76\ \text{mA} = 8.8\ \text{mW}, \tag{4.25}$$

and the dropout loss is:

$$P_{\text{LDO,loss}} \approx (5 - 3.3)\ \text{V} \cdot 1.76\ \text{mA} \approx 3.0\ \text{mW}. \tag{4.26}$$

The corresponding battery-equivalent IMU current is:

$$I_{\text{bat,IMU}} \approx \frac{8.8\ \text{mW}}{3.7\ \text{V} \cdot 0.9} \approx 2.6\ \text{mA}. \tag{4.27}$$

4.3.5.4 Small but non-zero passive losses (quantified).

Boost feedback divider loss. The boost feedback divider draws a constant DC current of:

$$I_{\text{FB}} = \frac{V_{out}}{R_{\text{upper}} + R_{\text{lower}}} \approx \frac{5\ \text{V}}{83.1\ \text{k}\Omega} \approx 60\ \mu\text{A}, \tag{4.28}$$

resulting in:

$$P_{\text{FB}} = V_{out} \cdot I_{\text{FB}} \approx 5\ \text{V} \cdot 60\ \mu\text{A} = 0.30\ \text{mW}. \tag{4.29}$$

Battery voltage sensing divider loss. The battery voltage sensing divider (300 kΩ and 220 kΩ) draws:

$$I_{\text{div}} = \frac{V_{\text{bat}}}{R_{top} + R_{bottom}} \approx \frac{3.7\ \text{V}}{520\ \text{k}\Omega} \approx 7.1\ \mu\text{A}, \tag{4.30}$$

with a corresponding dissipated power of:

$$P_{\text{div}} \approx 3.7\ \text{V} \cdot 7.1\ \mu\text{A} = 0.026\ \text{mW}. \tag{4.31}$$

Pull-up / pull-down resistor worst-case upper bound. The PCB includes multiple pull-up and pull-down resistors (commonly 10 kΩ) for SPI and control signals. The worst-case static loss occurs when a pull-up is continuously forced low, producing a DC path to ground. For a 10 kΩ pull-up at 3.3 V:

$$I_{10k} = \frac{3.3\ \text{V}}{10\ \text{k}\Omega} = 0.33\ \text{mA}, \tag{4.32}$$

corresponding to:

$$P_{10k} = 3.3\ \text{V} \cdot 0.33\ \text{mA} \approx 1.1\ \text{mW}. \tag{4.33}$$

Even if $N = 14$ pull resistors were simultaneously driven into this worst-case condition, the upper bound would remain:

$$I_{\text{pull,worst}} \approx 14 \times 0.33\ \text{mA} \approx 4.6\ \text{mA}, \tag{4.34}$$

which is still small compared to the measured total current. In practice, only a subset of signals remain in such a worst-case state, and the typical DC loss is significantly lower.

Table 4.3: Estimated module-level battery-equivalent current contributions for the 10 Hz limited-logging configuration ($V_{\text{bat}} = 3.7$ V, $\eta_{\text{boost}} = 0.9$).

Subsystem / loss source	**Supply domain**	**Estimated current**	$I_{\text{bat,eq}}$
AMT222B-V encoder	5 V (boost)	16 mA @ 5 V	$\approx$ 24 mA
Power indicator LED (1 kΩ, $V_f \approx 2$ V)	5 V (boost)	$\approx$ 3 mA @ 5 V	$\approx$ 4.5 mA
2× IMU (ICM-42688-P) + LDO loss	3.3 V (via LDO)	1.76 mA @ 3.3 V	$\approx$ 2.6 mA
Boost feedback divider (73.1k/10k)	5 V (boost)	$\approx$ 60 μA @ 5 V	$\ll$ 0.1 mA
Battery sensing divider (300k/220k)	BAT	$\approx$ 7.1 μA @ 3.7 V	$\ll$ 0.1 mA
Pull-up/down resistors (upper bound)	3.3 V	$\leq$ 4.6 mA @ 3.3 V	$\leq$ 5 mA
Identified subtotal (typical)	–	–	$\approx$ **31 mA**
Measured total (10 Hz limited logging)	–	–	$\approx$ **147 mA**
Remainder (Feather S3 + misc)	BAT	–	$\approx$ **116 mA**

4.3.5.5 Explaining the gap to the measured system current.

In the 10 Hz limited-logging configuration, the measured average system current is approximately 147 mA. The identified peripheral loads contribute approximately 24 mA (encoder), 4–5 mA (power indicator LED), and 2–3 mA (two IMUs including LDO losses). The calculated passive losses (voltage dividers, typical pull-up dissipation, and reactive element conduction/ESR losses) are orders of magnitude smaller.

As a result, the remaining current (approximately 147 mA$-$31 mA $\approx$ 116 mA) is primarily attributed to the ESP32 Feather S3 microcontroller and firmware-dependent activity, including CPU load, sensor communication overhead, microSD idle power, and background bus transactions. This remainder dominates the overall power budget, explaining why measured runtime is strongly influenced by the microcontroller operating state and data logging workload rather than by IMU sensing or passive losses.

4.3.5.6 Opportunities for power optimization.

Based on this breakdown, several optimization directions are suggested:

- **Encoder power reduction:** The encoder contributes approximately 24 mA battery-equivalent current. Using a lower-current encoder model or implementing duty-cycled measurement (if compatible with the sensing objective) would significantly improve runtime.
- **Improve the 5V→3.3V conversion path:** If the 3.3 V load increases in future iterations, replacing the LDO with a high-efficiency buck regulator would reduce dropout losses and improve overall efficiency.

- **Firmware-level ESP32 optimization:** The largest unaccounted current is associated with the ESP32 Feather S3. Reducing serial debug output, optimizing sensor acquisition routines, minimizing active CPU time, and enabling low-power operation between acquisitions can substantially reduce average current.
- **Optimize microSD logging strategy:** Buffered burst logging reduces average storage overhead. Further improvements may be achieved by minimizing write frequency, increasing buffer size, and reducing filesystem overhead.

Overall, the quantitative budget confirms that passive component losses are second-order effects, while the encoder subsystem and microcontroller activity dominate the measured system power consumption.

4.4 Discussion with respect to the research questions

4.4.1 Discussion of Research Question 3

Research Question 3: To ensure reliable, synchronized sensing performance through hardware redesign, embedded-software development, and systematic evaluation of measurement accuracy and system stability.

The objective of Research Question 3 is to evaluate whether the redesigned hardware architecture and embedded software can provide reliable and synchronized multi-sensor data acquisition with stable measurement performance. This objective is addressed through the experimental results presented in Section 4.2.

The noise and stability characteristics of the inertial measurements under different sampling configurations are analysed in Sections 4.2.1 and 4.2.2. The reported acceleration standard deviation and magnitude distributions indicate that both IMUs exhibit stable measurement behaviour under static and dynamic conditions. These results show that the redesigned sensing hardware and power distribution architecture do not introduce observable degradation in signal quality. Furthermore, the customized external IMU module demonstrates comparable performance to the on-board IMU, despite its reduced size and simplified circuitry.

Synchronization and consistency between the two IMUs are investigated in Section 4.2.3 using the inter-sensor acceleration deviation over time. The small and bounded deviation observed on all three axes indicates that the two sensing channels remain temporally aligned during continuous acquisition. This confirms that the adopted shared clocking strategy, SPI-based communication architecture and synchronized firmware acquisition scheme are sufficient to support multi-sensor operation without noticeable temporal drift.

In addition to the hardware design, the embedded software architecture plays an important role in ensuring timing stability. As discussed in Section 4.3, direct microSD write operations introduce non-deterministic latency that affects the sampling loop.

By introducing a buffered logging mechanism and separating sampling from storage operations, the timing jitter caused by file-system access is effectively reduced. The resulting timing stability metrics demonstrate that long-duration acquisition can be performed without significant disruption of the sampling process.

Overall, the experimental results demonstrate that the redesigned multi-bus hardware architecture, combined with synchronized firmware and buffered logging, enables reliable and synchronized sensing under the tested conditions. Therefore,

Research Question 3 is addressed for the implemented ESP32-S3 based system and the evaluated operating scenarios. However, broader generalization across alternative hardware platforms and extended dynamic conditions would require additional validation.

4.4.2 Discussion of Research Question 4

Research Question 4: To validate the redesigned system experimentally, demonstrating improvements in integration level, wearability, and usability relative to the original modular prototype.

Research Question 4 focuses on validating whether the proposed system provides practical improvements in integration level, wearability and usability compared to the previous modular prototype. These aspects are mainly evaluated through the integration and structural comparison results presented in Section 4.1.

The integration level is significantly improved by consolidating sensing, data logging, power regulation and communication interfaces onto a single expansion board and a compact external IMU module. As shown in Table 4.1 and Figures 4.3–4.5, the number of separate boards and interconnecting cables is substantially reduced compared to the original modular prototype. The redesigned stacked architecture eliminates several intermediate breakout boards and long cable connections, resulting in a more compact and mechanically robust system layout.

From a wearability perspective, the reduced footprint of both the main electronics enclosure and the customized IMU module directly supports improved placement within the exoskeleton structure. The smaller enclosure dimensions and reduced internal wiring enable closer mounting to the knee joint and minimise protrusion from the mechanical profile of the orthosis. These improvements reduce mechanical interference with the user and facilitate a more comfortable and unobtrusive integration into the wearable rehabilitation device.

Usability is improved primarily through simplified assembly and deployment. The reduced number of connectors and the elimination of splitter cables decrease the risk of wiring errors and shorten the system setup time during experimental preparation. The integrated architecture also improves mechanical robustness during handling and repeated attachment to the exoskeleton, which is particularly beneficial for iterative laboratory experiments and long-duration data collection sessions.

In summary, the experimental integration and structural comparison results demon-

strate clear improvements in integration level, wearability and usability with respect to the original modular prototype. Therefore, Research Question 4 is successfully validated by the proposed redesigned sensing system.

4.4.3 Summary

This chapter evaluated the redesigned sensing system through static, long-duration logging, and dynamic motion experiments. The results demonstrate stable multi-sensor acquisition and synchronized operation under the tested experimental conditions.

While the findings confirm reliable performance for the implemented ESP32-S3 based system, broader cross-platform robustness and long-term operational validation remain subjects for future investigation.

5

Conclusion and Future Work

5.1 Conclusion

This thesis presented the redesign and experimental validation of a compact multi-sensor data acquisition system for a lower-limb knee rehabilitation exoskeleton.

The redesigned PCB-based architecture successfully integrates sensing, storage, and communication into a consolidated platform, reducing system size and wiring complexity compared to the original modular prototype.

Experimental evaluation demonstrated deterministic acquisition timing at 250 Hz, with bounded jitter behavior and statistically consistent inter-sensor delays. High-resolution delay analysis confirmed Gaussian-like delay distributions with limited dispersion (σ in the order of a few microseconds) and no cumulative timing drift during cyclic knee motion between 0° and 100°.

The results validate the temporal synchronization architecture for the implemented ESP32-Feather S3 platform under the tested laboratory conditions.

However, the demonstrated synchronization stability and timing robustness are specific to the evaluated hardware and firmware implementation. Cross-platform validation, long-term operation testing, and performance characterization under more complex dynamic loading conditions were not conducted within the scope of this thesis.
Overall, the redesigned platform improves integration level, wearability, and usability while providing a validated foundation for synchronized multi-sensor acquisition in wearable rehabilitation systems.

5.1.1 Future Work

While the developed sensorized orthosis system has demonstrated stable operation and synchronized sensing performance under the evaluated experimental conditions, further validation and refinement are necessary to strengthen robustness, portability, and long-term reliability.

To make the next development phase actionable, the proposed future work is structured as a roadmap consisting of smaller milestones with estimated time requirements.

The time estimates assume part-time work in parallel with documentation and testing.

Post-Fabrication Debugging and Electrical Validation (1–2 weeks)

While the fabricated PCBs were functionally validated at the system level, further post-fabrication inspection and debugging are necessary to improve reliability and repeatability across multiple boards.

Extended Development Using Arduino Portenta H7 (2–4 weeks)

Future work will revisit integration with the Arduino Portenta H7 to evaluate higher-performance onboard processing and assess cross-platform portability of the sensing architecture.

The roadmap for Portenta integration includes:

– Milestone 1 (3–5 days): Minimal firmware bring-up and stable sensor readout at low sampling rate with serial debugging enabled.

– Milestone 2 (3–5 days): Establish a stable high-rate sampling loop and validate timing consistency relative to the ESP32-S3 baseline.

– Milestone 3 (3–5 days): Implement microSD burst logging and verify end-to-end logging integrity.

– Milestone 4 (3–7 days): Power management refinement and long-run battery testing to compare Portenta H7 versus ESP32-S3 baseline.

Power Consumption Optimization and Battery Life Extension (1–3 weeks)

Battery testing indicates that sampling frequency and microSD write operations significantly affect average power consumption.

Future work will focus on reducing energy overhead while maintaining sensing quality.

Planned steps include:

– Milestone 1 (2–3 days): Build a detailed power budget by measuring current draw of individual subsystems (MCU, IMU, microSD) in representative modes.

– Milestone 2 (3–5 days): Implement adaptive sampling and/or duty-cycling strategies.

– Milestone 3 (3–5 days): Reduce microSD logging overhead by increasing buffer size, lowering write frequency, or applying lightweight compression.

– Milestone 4 (2–4 days): Validate improvements through repeated runtime tests and timing stability measurements.

Bibliography

[1] A. Esquenazi, M. Talaty, A. Packel, and M. Saulino, "The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury," *American Journal of Physical Medicine & Rehabilitation*, vol. 91, no. 11, pp. 911–921, 2017.

[2] A. J. Young and D. P. Ferris, "State of the art and future directions for lower limb robotic exoskeletons," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 2, pp. 171–182, 2017.

[3] R. Riener, L. Lünenburger, and G. Colombo, "Human-centered robotics applied to gait training and assessment," *Journal of Rehabilitation Research and Development*, vol. 53, no. 5, pp. 679–694, 2016.

[4] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in Biotechnology*, vol. 31, no. 5, pp. 287–294, 2015.

[5] J. L. Pons, *Wearable Robots: Biomechatronic Exoskeletons*, Wiley, 2015.

[6] D. Novak and R. Riener, "A survey of sensor fusion methods in wearable robotics," *Robotics and Autonomous Systems*, vol. 73, pp. 155–170, 2015.

[7] A. M. Sabatini, "Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing," *Sensors*, vol. 11, no. 2, pp. 1489–1525, 2015.

[8] F. Just, M. Bogert, and R. Riener, "Joint-level sensor integration for wearable lower-limb exoskeletons," *IEEE Transactions on Medical Robotics and Bionics*, vol. 1, no. 2, pp. 75–85, 2018.

[9] F. Just, T. Nef, and R. Riener, "Embedded sensing and mechanical integration for wearable rehabilitation robots," *Frontiers in Robotics and AI*, vol. 7, pp. 1–14, 2020.

[10] S. DelPreto and H. Herr, "Design of a powered knee–ankle prosthesis with embedded sensing and control," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1581, 2018.

[11] N. Hogan and S. P. Buerger, "Impedance and interaction control," *Robotics and Automation Handbook*, CRC Press, 2016.

[12] A. T. Asbeck, S. M. De Rossi, K. G. Holt, and C. J. Walsh, "A biologically inspired soft exosuit for walking assistance," *The International Journal of Robotics Research*, vol. 34, no. 6, pp. 744–762, 2015.

[13] H. Kawamoto and Y. Sankai, "Power assist method based on phase sequence and muscle force condition for HAL," *Advanced Robotics*, vol. 19, no. 7, pp. 717–734, 2016.

[14] J. Tucker, K. MacDonald, and H. Kazerooni, "A review of exoskeletons for rehabilitation and human performance augmentation," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 2, pp. 1–15, 2015.

[15] S. Kim, J. Bae, and M. Tomizuka, "A human-centered approach to wearable robot design," *IEEE Robotics & Automation Magazine*, vol. 26, no. 2, pp. 68–79, 2019.

[16] A. Esquenazi, M. Talaty, and A. Packel, "The role of robotic exoskeletons in gait rehabilitation," *Journal of Rehabilitation Medicine*, vol. 51, no. 4, pp. 1–9, 2019.

[17] S. Calabrò, A. Accogli, and L. F. Molteni, "Wearable robotic exoskeletons for lower limb rehabilitation: A clinical perspective," *Frontiers in Neurorobotics*, vol. 14, pp. 1–14, 2020.

[18] F. Just, T. Nef, and R. Riener, "Sensor-based assessment of movement quality in rehabilitation robotics," *Sensors*, vol. 19, no. 18, pp. 1–17, 2019.

[19] F. Just and R. Riener, "Passive sensorized orthoses for long-term motion monitoring," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 3, pp. 331–340, 2020.

[20] D. Novak, M. Mihelj, and R. Riener, "Real-time sensor fusion for wearable robotic systems," *IEEE Reviews in Biomedical Engineering*, vol. 13, pp. 1–15, 2020.

[21] J. Camacho, A. Jardón, and C. Balaguer, "Embedded electronics design challenges in wearable rehabilitation robots," *Robotics*, vol. 9, no. 4, pp. 1–18, 2020.

[22] Y. Zhang, H. Wang, and X. Chen, "Miniaturized embedded systems for wearable robotics applications," *IEEE Access*, vol. 8, pp. 145672–145684, 2020.

[23] L. Rossi, F. Lunardini, and N. Vitiello, "Design of integrated sensing electronics for wearable lower-limb robots," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1205–1215, 2021.

[24] S. Mohammed, Y. Amirat, and H. Rifai, "Adaptive control and learning techniques in rehabilitation exoskeletons: A review," *Advanced Robotics*, vol. 34, no. 19, pp. 1231–1248, 2020.

[25] A. H. Khan, M. R. Hossain, and J. Kim, "Embedded intelligence for wearable rehabilitation devices: Opportunities and challenges," *Sensors*, vol. 22, no. 3, pp. 1–20, 2022.

A

Appendix 1

To support replication of this project, the implementation files have been made available in the following GitLab repository:

- **GitLab repository:** `https://git.chalmers.se/phric/aurora_phi/aurora_exoskeleton/supervision/masters/2025/exopcb/exopcb`

The shared repository contains the firmware used for the experimental validation presented in this thesis, including burst-based microSD logging configurations and communication interface test routines. In particular, the implementation covers multiple buffered logging setups:

- **0.2 s burst-write configuration:** buffer size $N = 200$ at 1000 Hz, resulting in one full-buffer microSD write operation every 0.2 s.
- **20 s burst-write configuration:** buffer size $N = 200$ at 10 Hz, resulting in one full-buffer microSD write operation every 20 s.
- **Additional 20 s logging test:** a supplementary low-frequency test configuration based on the same 10 Hz buffering scheme (one write every 20 s), intended for extended runtime validation. At the time of writing, this test code is planned to be uploaded to the repository.

In addition, the repository includes **communication interface testing** code used to verify stable sensor communication and reliable data transfer over the selected bus interfaces (e.g., IMU and storage interfaces) prior to long-run operation.

In addition, a video demonstration of the experimental setup and measurements is provided to support the experimental validation presented in this thesis. The video shows the user performing periodic leg motion with the orthosis while the real-time sensor plots are displayed on the computer screen.
The video recording is available at the following link:

- **Experimental demonstration video:** `https://www.youtube.com/watch?v=7c7nNLONAs8`

The video contains two recordings of the same experimental procedure performed using the same hardware setup and system configuration. The second recording was conducted to include explicit logging of the sensor output timestamps while keeping the same experimental protocol.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY