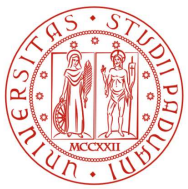

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITY OF PADUA
DEPARTMENT OF INFORMATION ENGINEERING
MASTER'S DEGREE IN ELECTRONIC ENGINEERING

Design and Development of an Optimized Wearable Device for Paralympic Athlete Performance and Prosthesis Monitoring (Olympia Project)

Supervisor: Prof. Claudio Narduzzi
Co-supervisors: Prof. Alberto Morato
Prof. Nicola Petrone
Ing. Andrea Giovanni Cutti

Candidate: Vincent Leone
Student ID: 2128860

ACADEMIC YEAR 2025–2026

Graduation date: 13 April 2026

Abstract

This work focuses on the design and development of a low-energy wearable device for the biomechanical analysis of Paralympic athletes, as part of the Olympia Project, a collaboration between the University of Padua and the INAIL Prosthetic Center in Budrio (BO). In this context, the constraints and objectives for the development of the device to be inserted into lower limb sports prostheses have been defined, including costs, form factor, interrogability, processing capacity, and energy consumption. The aim of the work is to create a compact electronic device capable of evaluating the performance of Paralympic athletes and the durability of prostheses through the acquisition and analysis of inertial data from IMU sensors. The document presents a detailed comparative analysis of the best low-power commercial platforms, such as ESP32 and nRF, providing quantitative comparisons between the performance of MCUs, inertial sensors and wireless communication in each platform, with a careful focus on the power consumption associated with each functional block. The Seeed XIAO nRF52840 Sense Plus board, identified as the most suitable solution, is described in detail and integrated with a 102050 lithium battery, after considerations regarding the energy profile. The robustness of this architecture and the achievement of the excellent performance trade-off guaranteed by this HW combination are demonstrated. This is followed by the development of a dedicated firmware for the nRF, structured around multiple scenarios and operating states, which allow the system to work in different operating modes, compatible with all the activities performed by the Paralympic athlete, processing data related to their biomechanics. To this end, algorithms for Activity Detection, Step Counting and extraction of significant biomechanical features are proposed, all designed with an embedded approach. The firmware is completed by the integration of the BLE module, which allows the transmission of data acquired and processed by the XIAO. The work concludes with experimental measurements carried out with a Keithley 6485 picoammeter and empirical tests on the device's autonomy in a real context, within the scenarios implemented in the firmware. Estimates and validations of energy consumption demonstrate the robustness of the proposed solution for use in monitoring sports prostheses in the Paralympic context.

Keywords: wearable devices, prosthetics, IMU, BLE, low-power, Olympia Project

Riassunto

Questo lavoro riguarda la progettazione e lo sviluppo di un dispositivo indossabile a basso consumo energetico per l'analisi biomeccanica di atleti paralimpici. L'attività si inserisce nel Progetto Olympia, una collaborazione tra l'Università di Padova e il Centro Protesi INAIL di Budrio (BO). In questo contesto sono stati definiti i principali vincoli progettuali, tra cui costi, fattore di forma, interrogabilità, capacità di elaborazione e consumo energetico, per l'integrazione del dispositivo in protesi sportive per arti inferiori. L'obiettivo è realizzare un sistema elettronico compatto in grado di valutare le prestazioni degli atleti e la durabilità delle protesi, attraverso l'acquisizione e l'analisi di dati inerziali provenienti da sensori IMU. A tal fine, viene presentata un'analisi comparativa delle principali piattaforme commerciali a basso consumo, come ESP32 e nRF. Il confronto considera le prestazioni di microcontrollori, sensori e moduli di comunicazione wireless, con particolare attenzione ai consumi energetici dei diversi blocchi funzionali. Tra le soluzioni analizzate, la scheda Seeed XIAO nRF52840 Sense Plus è risultata la più adatta. Essa viene descritta nel dettaglio e integrata con una batteria al litio 102050, sulla base di valutazioni legate al profilo energetico. Questa combinazione hardware consente di ottenere un buon compromesso tra prestazioni e consumi, garantendo al tempo stesso robustezza e affidabilità. Successivamente, è stato sviluppato un firmware dedicato per la piattaforma nRF. Il sistema è organizzato in diversi scenari e stati operativi, così da adattarsi alle differenti attività dell'atleta. In questo modo è possibile elaborare informazioni rilevanti sulla biomeccanica del movimento. A questo scopo sono stati implementati algoritmi di riconoscimento delle attività, conteggio dei passi ed estrazione di feature biomeccaniche significative, progettati secondo un approccio embedded. Il firmware include inoltre l'integrazione del modulo BLE, che consente la trasmissione dei dati acquisiti ed elaborati dal dispositivo. Il lavoro si conclude con una campagna di misure sperimentali, effettuate tramite un picoamperometro Keithley 6485, e con test di autonomia in condizioni reali. I risultati ottenuti confermano la validità della soluzione proposta e la sua idoneità per il monitoraggio di protesi sportive in ambito paralimpico.

Contents

Chapter 1

Introduction

1.1 Context and Motivation

In recent years, the diffusion of wearable devices has experienced significant growth in fields such as healthcare, rehabilitation, sports, and biomechanics [1]. The main reason for this development has been the progressive miniaturization of electronic systems, together with the widespread availability of low-cost inertial sensors. These advancements have enabled the acquisition and processing of motion signals directly on the device, reducing or, in some cases, eliminating the reliance on external data processing systems.

The application of wearable sensors to support clinical and rehabilitative assessments overcomes the limitations of traditional assessments relying solely on controlled environments, allowing for the measurement of motor activity in real-world situations [2]. This is especially beneficial for amputees and prosthetic users [3], for whom it is extremely important to understand their movement patterns and amount of activity and exertion in order to accurately assess their functioning and aid in identifying ways to improve their prosthetic device.

The implementation of integrated embedded systems has enabled further research into the capabilities of these systems, including interrogation of the device and real-time feedback. Therefore, wearable sensor systems are no longer limited to passive monitoring; rather, they can also evolve into systems providing corrective feedback to support different use cases.

However, with these benefits also come increased complexity due to traditional constraints associated with embedded systems when using motion analysis algorithms. Several algorithms that have been proposed in the scientific literature for activity classification and step detection have been validated primarily in offline settings or on high-performance computers [2], [4], [5]; therefore when implementing them on battery powered microcontrollers, all architectural and implementation decisions must be evaluated again.

This implementation must be carried out while staying within very tight constraints related

to energy use, size, cost, function and memory available. The goal of this research study is to create a wearable device that is optimised to help monitor the performance and use of Paralympic athletes as well as their prostheses during their daily activity.

1.2 Olympia Project and INAIL Prosthetic Center

This study is conducted as part of the Olympia Project, a collaborative research project created between the University of Padua and the INAIL Prosthetic Center of Budrio (BO). The objective of this project is to evaluate and investigate the functional performance of limb prostheses made for elite Paralympic athletes. The Olympia Project employs a multidisciplinary approach, bringing together expertise from biomedical, biomechanics, and engineering fields, and conducting experimental investigations in controlled sporting environments such as the Pala Indoor, a building dedicated to indoor athletics, for high-level training, by using dedicated equipment to analyze motions [6], [7].

Prostheses for the lower limb are of particular interest because they are exposed to a range of complex dynamic forces during various modes of movement, as well as in a variety of athletic activities. Measurements collected during an experimental session permit detailed analysis of both athlete performance and prosthesis behavior in standardized and reproducible test conditions and provide high-quality biomechanical data.

Alongside these methodologies, there is an increasing demand for systems which can collect individual data continuously, by lightweight and non-invasive devices, without interfering with training activities.

In this regard, the integration of inertial sensing systems mounted on the lower-limb prosthesis allows the collection of data that are complementary to those obtained through instruments such as SmartTrack, a sensorized track equipped with 7 meters of force plates, which measure the force the athlete applies to the ground while running. The integration of IMUs gives the ability to gather data over a longer period of time and across a greater number of training sessions.

The use of this long-term monitoring method also complements short-term measurements, thus providing valuable insights into how athletes' performance evolves over time and how the prosthesis holds up to the stresses of prolonged use.

This research fits into the electronics/embedded systems section of the Olympia Project and is aimed at building a wearable system that can receive and analyze inertial signals, then provide information about the physical tasks that are performed, as well as the use of the prosthesis. This system will eventually become a complementary tool to Smart Track and all biomechanical monitoring tools that require data collection to take place in a controlled environment. The availability of data collected in real-world settings will enable the identification of new features,

thanks to the device's ease of integration, its low invasiveness, and its low cost, especially when compared to the mechanical instruments typically used in laboratory experiments.

The device developed in this thesis work contributes to expanding the potential of the Olympia Project, gaining a better understanding of prosthesis behavior during sports activity and supporting the analysis of performance and device durability within a long-term monitoring perspective, consistent with the usage scenarios encountered by athletes during training.

Figure 1.1: Different positioning of the AX6 sensor across subjects: on the left, a Transfemoral (TF) configuration; in the center, a standard Transfemoral (TF) setup; on the right, a custom positioning for a Transtibial (TT) subject.

1.3 Previous Work

The experimental work carried out prior to this thesis is described here to clarify the complexity of the procedure and the individual steps that comprise it. The process shows the transition from the acquisition phase of the sampled inertial signal to the extraction of significant features, such as the type of activity performed by the athlete or the strides associated with it. The process begins with the acquisition of signals via the Axivity AX6 wearable inertial unit [8], configured to continuously record accelerations along the three axes (Anterior Posterior (AP), Cranio Caudal (CC), Medio Lateral (ML)), leaving the gyroscope deactivated for energy saving considerations. This phase is characterized solely by the measurement and storage of accelerations, without any type of processing. The AX6 is applied to the prosthesis with careful orientation of the device's axes.

The correct interpretation of these signals requires the construction of a reliable reference, which is not the result of statistical processing but of conscious and controlled labeling. To this end, a supervised session is held, in which the data recorded by the sensor is accompanied by a video recording, which is synchronized with the data to form a gold standard. Each recording is analyzed, manually counting the number of steps and distinguishing the activities shown. This phase provides a basis for comparing the algorithms developed subsequently. The

supervised sessions are divided into time intervals, which are subdivided by activity type; each interval is associated with the corresponding number of steps and video segments. This results in the creation of a structured dataset on which to base the algorithmic processing. This is followed by a calibration phase based on the individual athlete. Each subject has different physical, biomechanical, or prosthetic characteristics. It is necessary to teach the system to work according to the specifications of each subject, dividing the dataset into training and testing portions and thus implementing specific adaptation capabilities. Once the parameters have been defined, a validation phase follows. This is based on the consistency of the estimated results with respect to the gold standard. The correctness of the validation phase is a stringent requirement for proceeding in a real context. The system must ensure accuracy in detecting activities and the number of steps, starting from inertial data, adapting correctly to each subject. In the final phase, the algorithm is tested independently. The gold standard is no longer present, and the system must be able to classify biomechanical features over the long term. This marks the transition from the acquisition of raw data to the visualization of data that provides information on the load of activities and the resulting effects on the athlete's performance or the condition of the prosthesis.

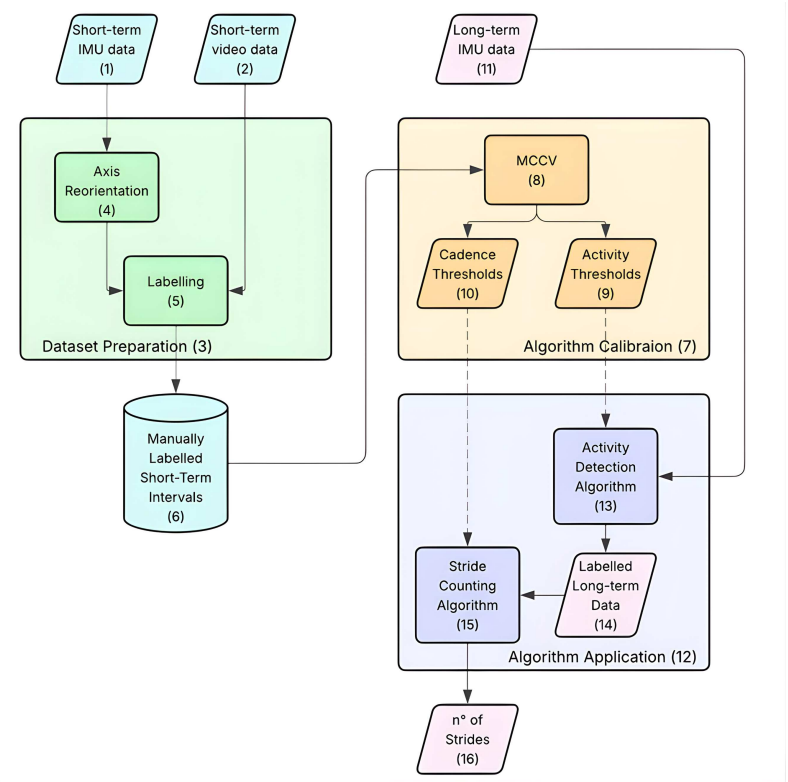

Figure 1.2: Workflow of the previous work underlying this study [9]: short-term acquisition (light blue), long-term monitoring (pink), dataset preparation (green), algorithm calibration (orange), and application (purple). Blocks are labeled from 1 to 16.

Chapter 2

State of the Art and Reference Work

The use of signals acquired by an inertial measurement unit, during a person's physical activity [10], [11], has been researched within the Olympia Project and proved to have great potential for understanding how well lower limb prostheses function and Paralympic athletes can move. In particular, research has demonstrated that using inertial sensors to obtain data related to the classification of activities and detection of steps can provide reliable indicators for future monitoring of prosthetic use and athletic performance [2], [12]. This provided the foundations upon which this thesis will be built. The work involved the use of accelerometric data and the development of algorithms to identify the various types of motion that a person with lower limb prosthetics will engage in, and provide an algorithmic estimate of the number of steps taken during physical activity. The results from these studies provided proof of concept that inertial data analysis can be used as an objective measure of the function of both the individual and the prosthesis.

This work has been designed for and validated within a setting where computational resources and energy supply are much more plentiful than would be true in a low powered wearable embedded system.

Performance levels of the algorithms conceived have shown good results with respect to accuracy, however they were developed in an offline environment, where stringent constraints related to device interrogability, local processing capabilities, and the associated energy consumption were not present.

The transfer of these results into a wearable device capable of being integrated into a lower limb prosthesis presents many important engineering challenges: first, we need to provide a hardware platform that has the capability of meeting the requirements of cost, form factor and autonomy; second, we must amend and adapt the current software solutions to be able to reliably work on low powered microcontrollers, keeping in mind the balance required between the algorithmic complexity, the quality of information extracted and the energy sustainability of

the processing algorithm.

A complete examination of the existing standards for hardware and software is now necessary because the limitations of each type may prevent their actual use in an embedded system. The analysis will provide context in which the reference work may be compared to other current solutions and show its restrictions in order to determine if those restrictions may prevent the use of the reference work with embedded systems. On these premises, the present work aims to bridge the gap between algorithmic validation in an experimental setting and the development of an efficient, robust wearable device capable of being utilized continuously over an extended period of time as part of the Olympia Project.

2.1 Inertial Acquisition System Used in the Olympia Project

2.1.1 Acquisition Device: Axivity AX6

During the preliminary experimental procedures for the Olympia Project, the Axivity AX6 was used to obtain the inertial measurements required for the experiments conducted. The Axivity AX6 is a data logger frequently used in biomechanics and clinical research; it contains a 6-axis inertial measurement unit (IMU), composed of a tri-axial accelerometer and a tri-axial gyroscope, for continuous logging mode, with the data being stored in its on-board internal memory. The AX6 has 1024 MB of non-volatile flash memory on board.

The embedded accelerometer provides acceleration measurements with 16-bit resolution, ensuring adequate sensitivity for detecting subtle variations in human movement. It is a MEMS-based inertial sensor capable of measuring acceleration along three orthogonal axes, with configurable sampling frequencies that can reach several hundred Hz depending on the acquisition settings.

MEMS (Micro-Electro-Mechanical Systems) technology enables the miniaturization of inertial sensors by integrating microscopic mechanical structures that act as transducers, converting inertial accelerations generated by body movement into electrical signals. The minimal footprint, low cost, and high reliability of MEMS have made it possible to widely use lightweight and compact wearable devices in clinical and sports contexts.

The physical size of the AX6 is roughly 23 × 32.5 × 8.9 mm and the weight is approximately 11 g; these factors allow for easy installation on a wearable device and do not interfere with the mechanical performance of the prosthetics.

The decision to use the Axivity AX6 in the initial research and development phase is justified by the validation of the device in scientific literature. The use of an already established platform has allowed to focus on data analysis and the development of algorithms for specific uses, based on a solid and validated hardware structure.

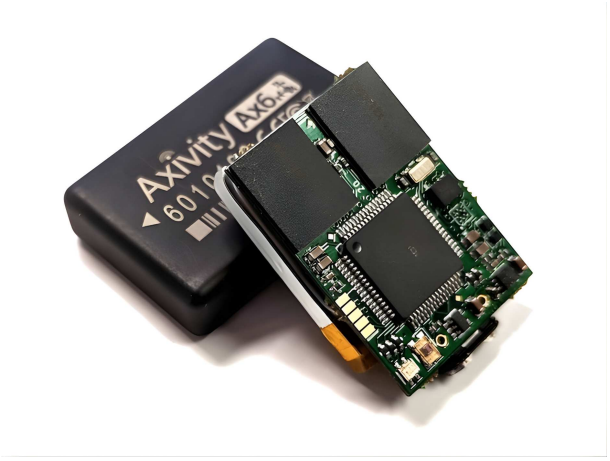

Figure 2.1: Axivity AX6 wearable inertial measurement unit used for data acquisition, configured to record triaxial accelerations.

2.1.2 Sensor Placement and Experimental Setup

The Olympia Project's Axivity device was placed directly on the lower limb prosthesis in a way that would allow for a maximal sensitivity to acceleration while simultaneously minimizing influences from extraneous vibrations/non-representative movements;

More precisely, the sensor is applied in a position proximal to the prosthetic knee and fixed near the socket, i.e., the component connecting the residual limb to the prosthetic structure. Stable installation is essential in order to reduce unwanted rotation of the sensor or unwanted movement of the device relative to the prosthetic structure. The quality of the inertial signal collected is closely related to this fixing phase.

To standardise integration of the sensors across different prosthetic configurations or subjects and help ensure repeatability of results, the sensor mounting procedures adhered to established lab standard protocols as defined in the experimental design phase.

For this reason, it is essential to define a consistent axis orientation, a standardised sensor attachment method, and clear anatomical reference points. Additionally, all procedural steps must be specified to ensure repeatability of the experimental conditions under which the measurements are performed and to minimize variability across subjects and prosthetic configurations.

It should be noted that the position of the sensor may vary depending on the subjects being examined. These subjects have different prosthetic configurations, which are also related to the anthropometric characteristics of the individual athlete. This variability requires special precautions to be taken in the subsequent phase of realigning the sensor to a common anatomical reference point.

2.1.3 Acquisition Configuration and Sampling Frequency

Continuous collection of tri-axial acceleration signals in the Olympia Project is done using the Axivity AX6, with no onboard signal processing or filtering. The sampling rate used during collection was 12.5 Hz, which is an adequate rate to characterize the different modes of locomotion and limit the amount of data recorded, as well as facilitating analysis after data collection, since the dynamic components associated with human locomotor activities occur in a very low frequency band, below a few Hz.

This configuration of the Axivity AX6 supports long-term logging capability, however, the device operates in a fixed acquisition environment, and it is not capable of adapting the sampling frequency based on the activity being measured or conditions during operation. The configuration could introduce complications in terms of quantitative analysis of the impact of individual movements during sprinting activities. Here, the step frequency reaches the acceptable limit for signal analysis that is not affected by aliasing. Nevertheless, for the purposes of activity classification and step detection, the frequency of 12.5 Hz proves to be sufficiently accurate.

2.1.4 Data Flow and Transfer Procedures

The Axivity device records data using internal storage until the acquisition process is completed. Data transfers from the device to an external host occur through a wired connection, at which point the raw data may be downloaded and converted into a standard format (typically CSV) for use in subsequent data analysis.

The exported data contain time series describing the triplets of axial accelerations measured by the sensor. It also includes time-related information, which allows the signal trend to be reconstructed over time, as required for processing and extracting features related to motor activity.

This workflow clearly distinguishes between acquiring data from an experiment and analyzing those data after all the experiments have been completed. While this approach is sufficient for conducting controlled experimental studies, it introduces a considerable amount of time between when an experiment is conducted and when researchers have access to information and can perform an analysis. Consequently, there is no opportunity for real-time feedback or interaction with the Axivity device during the experiment.

2.1.5 Synchronization with External Analysis Systems

Part of the experimental protocol includes aligning inertial data collected from the inertial sensors with an external system of motion analysis, specifically video recordings, for biomechanical validation. Through visual analysis and manual labeling of the various locomotor activities,

these recordings allow to establish a reliable reference for the classification phase and for the evaluation of the algorithms developed. For time synchronizing the accelerometer signal to the video stream, dedicated methods are used (e.g., using reference events or time alignment strategies).

Data synchronization is an important step in the experimental process and requires detailed data management, as well as additional processing both during offline data preparation and signal processing, before detailed and accurate motion analysis can be performed. The use of video as a reference improves the biomechanical validation of inertial data, but adds complexity to the analysis process and makes this approach unsuitable for continuous monitoring systems or studies conducted in uncontrolled environments.

2.1.6 Advantages and Limitations of the Adopted System

By the use of the Axivity AX6, it has been shown experimentally that motion analysis algorithms created as part of the Olympia Project can be validated through the high quality inertial data acquired. The Axivity AX6's ease of use, robustness and long working time make it particularly well suited to biomechanical studies in a controlled setting.

The Axivity AX6 boasts an exceptional battery life of up to 140 days, even when doing continuous measuring. The high battery life is due primarily to the fact that it's a "stand-alone", so it doesn't communicate wirelessly or perform any onboard computations. The absence of these features significantly reduces energy consumption, as the device only acquires and stores unprocessed data. This configuration is particularly suitable for long-term applications based on continuous acquisition.

However, there are some limitations with regards to how well a system like this could be integrated into a wearable embedded system for long-term and continuous use. Most notable among these limitations would be the inability to interrogate what is happening with the sensor; no processing capability on-board the Axivity AX6; a requirement for wired data transfer; and an offline method of synchronizing data. These limitations significantly complicate operation and impair the possibilities for using such systems for real-time feedback, dynamic adaptation of the system based upon feedback from the environment in which they operate, and/or operation with a minimal amount of energy.

These limitations have motivated the need for exploring alternative solutions that provide sufficient quality of information while also reducing the operational complexity of an embedded system, and allowing for efficient, occasional interrogation of the system and/or adaptive behaviour. The following chapters will address these issues in more detail.

2.2 Software State of the Art: Activity Classification and Step Counting Algorithms

2.2.1 Objectives of Software Analysis

Within this context, the software proposed by Olympia Project, which preceded the work in this thesis, aims to extract reliable quantitative information on motor activity and lower-limb prosthesis usage from low-frequency inertial signals. In particular, the work focuses on: the automatic classification of the activity performed by paralympic athletes (Stop, Walk, Jog, Sprint or Swing); the estimation of the number of steps associated with each activity; the availability of the data across an extended period of time for long-term tracking purposes.

These assessments allow us to estimate the functional load to which the prosthesis is subjected, providing relevant information on possible modifications that can be made during the prosthesis design phase or effective interventions in the athlete's training schedule, in order to avoid overloading areas that are excessively sensitive to stress.

The entire development of these software tools took place within an offline environment and has no strict limitations on latency, energy usage or CPU processing power.

Signal Acquisition and Pre-processing

The inertial signals used by the algorithm consist of tri-axial accelerations acquired at 12.5 Hz using an Axivity AX6 sensor. Due to variability in sensor placement across different athletes and prosthetic configurations, an axis realignment phase is introduced, mapping the raw sensor axes to a standardized anatomical reference frame as indicated by axes labelled as follows: antero-posterior -> (AP) axis, cranio-caudal -> (CC) axis, medio-lateral -> (ML) axis.

Realigning the axes allows you to compensate for sensor rotations due to the mounting phase on the prosthesis or inter-subject variability, which causes slightly different positions. Once a reference system has been defined that is compatible with the biomechanics of human movement, comparisons can be made between different training sessions and different athletes.

The realignment of raw data will allow data comparisons across subjects and represents an important prerequisite for future feature extraction.

Time Segmentation and Definition of Motion Intervals

A sliding window of 48 samples (approximately 3.84 s) with 50% overlap (step size of 24 samples) was used to segment the accelerometric signal. Within the framework, Motion Intervals (MIs) were defined as temporal segments that exhibit a single homogeneous motor activity. MIs

serve as the basic unit for calibration of the algorithm and for its validation against a gold standard.

The use of an overlapping sliding window allows the signal to be analyzed continuously over time, reducing the likelihood of losing relevant information, especially in the transition phase between consecutive windows, and obtaining stable features that are not subject to classifications based on sporadic, short-lived events.

In each time window, three time domain features were derived by using the maximum absolute acceleration value within each of the three axes (APmax, CCmax and MLmax). These time-domain features are effective for distinguishing between different conditions: immobile conditions (Stop); low-dynamic activities; higher mechanical stress activities. In particular, CCmax is the primary feature that differentiates Stop conditions from locomotor conditions.

2.2.2 Frequency-Domain Analysis via Continuous Wavelet Transform

Since time-domain statistics alone do not adequately capture the periodic structure of the motion, it is reasonable to assume that time-domain information does not best represent the spectral differences between the various speeds (walking, jogging, and sprinting).

Therefore, time-domain analysis is complemented by the use of a Continuous Wavelet Transform (CWT) on the accelerometric signal along the anterior-posterior (AP) direction.

The Continuous Wavelet Transform is a very suitable tool for analyzing non-stationary signals, such as those related to movement. By combining representations in the time and frequency domains, it allows local variations in the spectral structure of the signal to be identified.

More specifically:

- a Morlet Complex Wavelet (cmor 1.0–0.5) is employed, because the wavelet allows capturing the slow modulations of the signal;
- the CWT is performed using 48 samples of data;
- from the transform the feature L_{AP} is extracted, derived as the amplitude of the wavelet peak within the low frequency range 0–1 Hz.

The feature reflects variations in the periodic structure of the anterior–posterior acceleration associated with different locomotion speeds and accurately differentiates between a Walk and a Jog or between a Jog and a Sprint.

2.2.3 Activity Classification Algorithm

Activity classification is implemented using a binary decision tree, hierarchically structured and modelled on subject-specific threshold values. This makes the classification process easy to

interpret and the structure very clean. The nodes of the tree represent logical conditions based on simple features and ensure relatively lightweight algorithms, even from a computational point of view.

The algorithm follows a deterministic path that: excludes non-relevant activities (e.g. swing phases); identifies Stop conditions using CCmax; discriminates Walk, Jog, and Sprint using L_{AP}. The relevant thresholds T1, T2, and T3 are estimated through a Stratified Monte Carlo Cross-Validation (MCCV) procedure, using 50 iterations.

MCCV is a statistical validation technique used to estimate the performance of a classification model. It is often used when limited datasets are available and, unlike classic k-fold cross validation, it is based on repeated random sampling. Each iteration divides the dataset into two subsets: the Training Set and the Test Set. These allow the classifier to be trained and the model to be validated, returning average performance across all iterations and thus providing robust and generalized results. Furthermore, thanks to its random division structure, MCCV reduces the risk of overfitting, provides stable performance estimates, and allows for the efficient use of small datasets. In the case in question, the computational cost is relatively low thanks to the reduced complexity of the classifier. However, this analysis and classification structure is suitable for an offline processing context and would not be implementable in an integrated, low-power solution for real-time monitoring purposes [3].

2.2.4 Cadence Estimation and Step Counting Algorithm

Step counting is not performed using event-by-event signal analysis, instead, it has been accomplished using the estimation of the average cadence of each of the three activity classes (Walk, Jog, Sprint). This design choice is also dictated by the fact that signals originating from the prosthesis' contact with the ground can be more difficult to identify due to greater variability compared to those detected in able-bodied subjects.

For each of the Motion Intervals, cadence is determined as follows: "Cadence" = "Number of steps" / "Interval duration".
Cadence values are estimated during the calibration phase of the MCCV and subsequently used to estimate the total number of steps based on the cumulative duration of the recognized activities.

The algorithm is validated by comparing its outputs with a reference, obtained through video synchronization. The results show: activity classification accuracy of about 98%; relative step counting error less than 1% across all types of activities. These results demonstrate the methodological robustness of the proposed approach in an offline, controlled experimental environment.

2.2.5 Concluding Remarks on the Software State of the Art

The reference document [9] demonstrates that by combining low-frequency accelerometry with time-frequency processing techniques and specific calibration techniques, highly accurate monitoring of activity and prosthesis usage can be achieved. However, the entire architecture of the software system has been developed to operate in an offline processing mode. Therefore, issues related to energy requirements, latency, and computational resource constraints are not of concern. These aspects become central when considering an embedded implementation. The activity classification algorithm, in fact, while guaranteeing 98% accuracy, has a computational complexity equal to O(NlogN) operations, where N = number of samples of the analyzed signal, thanks to the use of FFT made possible by a special library. This computational cost, which is well suited to the use of a common PC in an offline environment, is not compatible with the limited resources of the nRF52840 SoC and its pre-set energy autonomy objectives. It is necessary to identify new descriptors in the time domain that are less costly but equally significant.

In order to integrate each of the proposals presented in an offline environment, in a wearable, queryable, low-energy device, it is necessary to adapt everything that has been described, with a careful focus on optimization.

2.3 Transition Toward Embedded Systems

This chapter has provided a summary of state-of-the-art research demonstrating how inertial signals, or movement data recorded by sensors, can be used to determine how lower-limb prostheses behave, as well as how Paralympic athletes move. The results obtained so far in the Olympia Project provide evidence that the algorithms used to analyze signal data are methodologically sound and yield accurate results when applied in an offline experimental setting.

The use of these methods in a wearable system integrated into a prosthesis, results in a new set of challenges not encountered in an offline computing environment. Specifically, a wearable system that operates on battery power and provides long-term usability must be designed with consideration for issues such as power consumption, size/shape, manufacturing costs, and processing capabilities.

These factors become imperative to the overall design of the prosthetic system. Based upon the above observations, this work represents a design evolution from experimental data analysis to the creation of an optimized embedded system that can acquire and analyze inertial data locally, while providing an optimally balanced relationship between algorithm complexity, quality of the information provided, and length of operational autonomy.

Chapter 3

System Requirements and Design Constraints

To design a wearable device integrated into a lower-limb prosthesis, it is necessary to clearly define the project requirements, which must comply with the conditions set out in the Olympia project specification and reflect the limitations identified through analysis of the existing technology.

The embedded device must be capable of reliable operation in real-world conditions, while maintaining appropriate trade-offs in functionality, energy autonomy, physical size, and overall cost. The goal of this chapter is to clarify the requirements (both functional and non-functional) for the embedded device. These specified requirements, along with the integration constraints of embedded platforms, have directed the choices made regarding hardware components and firmware architecture in this study.

3.1 Functional Requirements

The wearable system developed in this work is not conceived as a simple data logger, but as a device capable of acquiring, processing, and providing concise information on the performed activity, operating autonomously over time. Functional requirements formally describe the functionalities that a system must provide, specifying what the system is expected to do in order to meet the application objectives.

First, the device must support the acquisition of inertial signals. These include tri-axial linear and angular accelerations, within the frequency range defined during the offline algorithm validation phase. The system must be able to operate at sampling rates on the order of several tens of Hz, ensuring adequate signal quality for the extraction of features required for on-board activity classification and step counting.

This requirement directly leads to the need for sufficient on-board data processing capabilities. Activity classification and step detection algorithms are executed locally on the microcontroller, ensuring independence from external processing. This approach enables continuous and autonomous monitoring and ensures the extraction of high-level, synthetic information rather than raw data.

A further key requirement concerns system interrogability, defined as the ability to access both the results of the processing and the system status through a wireless communication interface. Data transmission, event notifications, and configuration of operating parameters must therefore be supported without relying on wired connections or manual data download procedures.

The system must also allow for configurable behavior. The firmware is required to support different operating scenarios, enabling the analysis across multiple contexts, including laboratory testing and real-world usage. Modification of operating parameters, such as sampling frequency and algorithm thresholds, must be possible depending on the application context. This flexibility is particularly relevant for long-term monitoring studies.

Another important functional requirement is the management of inactivity periods. This feature is essential to preserve energy and requires the ability to detect the absence of movement and dynamically adapt system behavior, without compromising reliable reactivation when new activity is detected.

Finally, functional requirements also include the availability of a suitable development environment, enabling efficient, modular, and maintainable firmware development. The environment must provide appropriate debugging and profiling tools and facilitate the integration of libraries required for sensor management and communication. This aspect is crucial to ensure reproducibility of the work, extensibility of the firmware, and overall reliability of the development process.

3.2 Non-Functional Requirements

Non-functional requirements of the system were defined during the preliminary phases of the project, following discussions with the INAIL Prosthetic Center and in alignment with the clinical and application-oriented objectives of the Olympia Project. These requirements represent initial constraints imposed on the design, independent of the hardware and firmware solutions subsequently adopted, and they provided the quantitative reference framework for all the design choices discussed in the following chapters.

The first non-functional requirement concerns the operational lifetime of the system. The device is conceived for long-term monitoring, with the ambition of ensuring reliable operation

over a period on the order of 36 months without frequent user intervention. This objective is particularly challenging for a battery-powered system and required, from the earliest stages of the project, a strong focus on energy sustainability and long-term power consumption management.

The expected usage profile is characterized by a marked temporal asymmetry, with limited periods of effective activity (associated with training sessions or tests) and extended periods of inactivity, during which the device remains installed on the prosthesis but is not subjected to significant mechanical excitation. In this context, the average energy consumption of the system must be compatible with multi-year autonomy, making the minimization of power drawn during inactivity phases a critical design constraint.

Another fundamental non-functional requirement is the form factor. The device must exhibit a geometry compatible with integration within the internal space of a lower-limb prosthesis, without interfering with mechanical functionality or user comfort. In particular, the following upper bounds were defined: a maximum width of approximately 2 cm; a maximum thickness of approximately 1 cm. Conversely, the length of the device does not represent a strict constraint, as it is more easily adaptable to the available internal spaces of the prosthetic structure. This requirement excludes solutions with compact yet bulky form factors and favors devices with a longitudinal layout, significantly restricting the range of commercially available platforms.

The unit cost of the system represents another relevant design constraint. A maximum threshold of 40 euros per unit was defined, with an ideal target range between 30 and 40 euros. This constraint stems from the objective of making the device potentially applicable not only in experimental sports contexts, but also, in the future, on non-sport prostheses, thus enabling broader and more sustainable deployment scenarios. As a result, proprietary or high-cost platforms were excluded, and preference was given to solutions based on widely available commercial components.

With regard to system interrogability, which is introduced among the functional requirements, important non-functional constraints also arise. In particular, the requirement to allow the device to be interrogated via a smartphone implies: limited energy consumption associated with wireless communication; high reliability of the communication stack; compatibility with widely used consumer mobile devices. The specific choice of wireless communication technology was therefore not fixed at this stage, but deferred to the hardware selection phase, where different options were evaluated in light of these constraints.

These aspects have a direct impact on the overall energy autonomy of the system and must be considered as non-functional design constraints, despite originating from a primary functional requirement.

Finally, an additional non-functional requirement concerns the operational reliability and robustness of the system. The device must be capable of continuous and unsupervised operation,

tolerating long periods of inactivity, repeated cycles of subsystem activation and deactivation, and variations in operating conditions, while maintaining predictable and consistent behavior over time.

3.3 Requirements Considerations

Taken together, the functional and non-functional requirements defined in this chapter outline a clear and stringent set of constraints for the design of the wearable system. The need to support local processing of inertial signals, ensure wireless interrogability, comply with strict power consumption limits, and maintain a form factor compatible with integration into a lower-limb prosthesis significantly narrows the range of feasible technological solutions. In light of these considerations, the selection of the hardware platform cannot be addressed as a purely performance-driven choice, but must instead result from a trade-off among computational capabilities, energy autonomy, physical dimensions, cost, and firmware development support. The following chapter is therefore dedicated to a comparative analysis of the main candidate platforms and to the justification of the hardware solution adopted in this work.

Chapter 4

Hardware Design and Platform Selection

4.1 Hardware Platform Selection

The process of selecting the hardware platform considered both functional and non-functional requirements defined in Chapter 3. The goal was not a platform that offered the best overall performance, but rather to find the platform that would best fit the actual application constraints of the Olympia Project in a real-world, long-term operating environment. Selection of the platform included a comparative analysis of commonly used development platforms and microcontrollers, considering a representative sample of the different classes of technologies available for a given architecture, power requirements, sensor integration and firmware development. The platforms evaluated included those based on ESP32 microcontrollers, compact devices in the Seeed XIAO family, Arduino Nano microcontrollers and other low-power microcontrollers.

Evaluation criteria were based directly on project requirements and included: ability to acquire inertial data and integrate all necessary sensors; energy consumption during both active and idle operation; wireless communication capabilities; form factor and compatibility with mechanical integration; unit cost and commercial availability; support of development environments, debugging tools and maturity of the software ecosystem. The comparison presented in this chapter will take into account quantitative considerations, based on the technical specifications reported in datasheets and in the literature, as well as qualitative considerations, related to the flexibility of the platform and the documentation available for the development environment. Solutions that did not meet the most stringent constraints were progressively excluded, narrowing down the pool of alternatives available for the choice of the hardware platform to be adopted in the system. The following paragraphs describe in detail the platforms and the comparison between them, highlighting the reasons that led to the final project choice.

4.2 Candidate Platforms

4.2.1 CodeCell C3

The first platform considered was the CodeCell C3 [13], [14], identified at the INAIL Prosthetics Center as a potential candidate for the device's implementation, as it is designed for embedded applications and oriented towards IMU signal analysis.

The board is based on an ESP32-C3 controller [15], a low-power SoC featuring a single core RISC-V architecture, and providing Wi-Fi and BLE 5.0 connectivity. The ESP32-C3 can operate at clock frequencies of up to 160 MHz, is equipped with 400 kB of SRAM and 4 MB of external Flash memory, ensuring good flexibility in firmware programming and data acquisition.

An interesting feature of CodeCell C3 is the set of sensors on board. These include a 9-degree-of-freedom IMU (BNO085) [16], offering the possibility of future developments that can integrate the use of a magnetometer, accelerometer, and gyroscope into classification algorithms. There are also sensors for environmental quantities, like proximity and light sensors.

As mentioned, wireless communication is guaranteed by Wi-Fi and Bluetooth Low Energy support, making the platform flexible for direct interactions with mobile devices as well as more complex network infrastructures.

Another point of strenght is the card's form factor. Its dimensions of 18.5 x 18.5 x 9.4 mm and weight of 3.4 g make the card easy to integrate within the limits of the pre-established dimensional constraints.

One of the most important aspects, however, concerns the CodeCell C3 energy management. The platform integrates a charging circuit for lithium polymer (Li-Po) batteries and power management, however specifications report relatively high average sleep currents, ranging from 476 μA to 860 μA. Considering the asymmetric and long-term application requirements, these figures raise doubts about energy efficiency in sleep mode that plays a primary role in the device's autonomy.

From a firmware development perspective, the CodeCell C3 is supported by the ESP-IDF ecosystem and Arduino-compatible development environments, offering libraries and application examples that facilitate code development.

4.2.2 Arduino Nano 33 BLE Sense

Another platform that has been considered as a candidate for this work is the Arduino Nano 33 BLE Sense [17]. Widely used in research contexts for rapid prototyping, it is one of the most popular solutions based on the nRF52840 microcontroller and offers an excellent compromise between ease of use and performance in sensing applications.

From a computational point of view, the Nordic nRF52840 microcontroller [18] is an ARM Cortex-M4F [19] architecture operating at 64 MHz, equipped with 256 kB of RAM and 1 MB of Flash memory. The literature demonstrates the adequacy of these resources for the applications under consideration, making this architecture particularly interesting.

The Arduino Nano 33 BLE Sense also stands out for its many onboard sensors. These include a 9-degree-of-freedom IMU (LSM9DS1)[20] and various environmental sensors, such as temperature, humidity, pressure, brightness, and a microphone, which allow for a wide range of applications. This is another reason why this platform represents a standard for algorithmic validation in prototyping contexts.

The board supports BLE, provided directly by the nRF52840 microcontroller. BLE communication is well suited for short-range interaction with mobile devices, making additional Wi-Fi connectivity unnecessary. The development of a wearable device for long-term personal monitoring is consistent with this design choice.

The mechanical profile measures 45 x 18 x 1.3 mm and the device weighs 5 g. Although the form factor meets the initial project constraints, potential risks due to space limitations for future expansion must be considered if additional components need to be integrated.

The most critical aspect of this solution concerns power management. Since there is no charging circuit for Li-Po batteries, external components must be used. Furthermore, consumption values reported in the documentation refer to individual components and do not provide consumption values for the board equipped with sensors, making it difficult to estimate consumption in sleep mode.

From a firmware development perspective, the available ecosystem is extremely mature, thanks to native support for the Arduino IDE and the availability of libraries. Things become more complex in the case of ultra-low-power development phases, where the focus is solely on power efficiency.

4.2.3 ESP32-based platforms

One family of platforms that was considered during the analysis of characteristics is the ESP32, widely used in academic and experimental contexts thanks to its integrated resources and enormous potential. These microcontrollers are typically based on Xtensa LX6/LX7 cores or, in the most recent versions, on a RISC-V architecture [15]. The clock frequency generally varies in a range between 160 MHz and 240 MHz, and configurations can be either single-core or dual-core. The SRAM, in the order of hundreds of kB, and the Flash memory, typically between 2 MB and 16 MB, show very high capacity compared to common low-power platforms.

Integrated wireless connectivity includes native BLE and Wi-Fi, which are often available simultaneously. The resulting flexibility allows for applications that interact with both mobile

devices and integrated network infrastructures. However, the presence of Wi-Fi entails higher energy consumption, even when this interface is not in use.

Most ESP32 platforms do not integrate onboard sensors. The need to acquire inertial signals therefore requires the use of external sensors, connected via I^2C or SPI bus. This is not necessarily a negative aspect, considering the resulting flexibility in sensor choice, but it is important to bear in mind the increased hardware complexity and power consumption due to the additional components and communication interfaces that need to be managed.

Furthermore, ESP32 microcontroller datasheets report power consumption under ideal conditions and with most peripherals disabled. It is necessary to pay attention to actual power consumption, considering all the components involved.

Power consumption in sleep mode for ESP32 boards is in the order of hundreds of μA or higher, making the goal of achieving multi-year device autonomy without significant hardware interventions utopian.

ESP32s are available in different form factors, and do not introduce limitations in terms of the physical integration of the platform.

The ecosystem supporting firmware development is a strong point of the ESP32, which benefits from numerous open-source libraries, Arduino support, and includes the ESP-IDF, facilitating project development and prototyping.

4.2.4 Seeed Studio XIAO family

The Seeed Studio XIAO family of platforms is an extremely interesting class of solutions, thanks to the coexistence of factors such as extreme compactness, low energy consumption, and the availability of different functional variants with the same form factor, suitable for the application context of this work.

The different versions are based on ARM Cortex-M0+/M4/RISC-V [19] (and others) and nRF52, and the design is aimed at low-power wearable and IoT applications. All XIAO boards have a form factor of 21 x 17.5 mm and the thickness of the PCB alone is less than 1.2 mm, making them perfect candidates for integration into lower limb prostheses.

In addition, the "Sense" variants of XIAO integrate onboard 6-axis IMU sensors, simplifying the system architecture and eliminating the need for external sensors. The difference from a 9-axis IMU lies in the absence of a magnetometer, which is not used for our purposes. The integration not only ensures system reliability but also has a positive impact on power consumption. In fact, the energy consumption profile is a central aspect of this work. XIAO platforms exploit the microcontroller's deep sleep mode, delivering very low currents, in the order of a few microamperes, under ideal conditions and with peripherals disabled. Furthermore, even in active mode, when used at low inertial sampling rates, they guarantee excellent performance,

more in line with the objectives defined in Chapter 3, compared to the platforms analyzed previously. To support power consumption reduction, variants of the family without Wi-Fi are used. In these cases, communication takes place via BLE, which offers native support for device interrogation without the need for external modules.

Firmware development benefits from the Arduino framework and, in some cases, the microcontroller manufacturer's dedicated SDK. This allows flexibility between rapid development and the possibility of low-level optimization.

Finally, the compactness of the board and the availability of standard interfaces (I^2C, SPI, UART) offer interesting potential for future developments. These include the creation of custom PCBs and the integration of external memory to store large amounts of inertial data for sporadic analysis.

Seeed Studio XIAO nRF52840 Sense Plus

The Seeed Studio XIAO family platform best suited to the application context under consideration is the "nRF52840 Sense Plus" [21]. In addition to its small form factor and low cost, it combines low power consumption, integrated inertial sensors, and native support for Bluetooth Low Energy. The architecture is based on the Nordic Semiconductor nRF52840, an ARM Cortex-M4F microcontroller equipped with FPU, with a clock frequency of up to 64 MHz [19]. It features 256 kB of SRAM and 1 MB of Flash memory. These features enable the development and execution of activity classification algorithms, step detection, effective BLE communication, and intelligent resource management based on the circumstantial needs of the project.

The "Sense Plus" also integrates an onboard 6-degree-of-freedom IMU sensor (LSM6DS3TR-C) [22], including an accelerometer and gyroscope, eliminating the need for external components thanks to the internal connection via I^2C bus (interfaces also include UART, SPI, and I^2S). This reduces the area occupied on the device, the overall energy consumption of the system, the complexity of the hardware and, with it, potential mechanical and electrical failures. The IMU supports frequencies from 12.5 Hz to 1.66 kHz, ensuring compatibility with those used in offline algorithmic validation. Built-in sensors also include a PDM microphone, which is of interest for possible project developments in the sports context, particularly in athletics. In addition to these, there are physical buttons and status LEDs which, although not a relevant factor for the final operational use, provide valuable support during debugging, firmware validation, and experimental verification of the various operating scenarios.

The nRF52840 is designed for ultra-low-power applications. Sleep currents are typically less than 5 μA (1-2 μA when the system is OFF) and active currents are low (on the order of mA), making it particularly suitable for applications with a reduced duty cycle. The platform integrates Bluetooth Low Energy 5.4 [23], with support for the Nordic SoftDevice stack, which is a

precompiled, certified, and tested binary file that resides in the lower part of the microcontroller's flash memory. It acts as an interface between the radio hardware and the user application, allowing the device to be queried via smartphone and the transmission of events and parameters of interest, as well as the ability to remotely configure the firmware. This provides advantages in terms of system reliability and limits power consumption compared to Wi-Fi-based solutions.

The development environment benefits from the Arduino framework and the Nordic ecosystem (nRF SDK / Zephyr) and allows for both rapid development during prototyping and low-level power consumption optimization. This flexibility is a fundamental requirement for the development of complex projects that require continuous analysis and improvement over time, as is typically the case in research.

Finally, the XIAO nRF52840 Sense Plus stands out for its commercial availability, the short lead time for the platform, and the unit cost of 15 dollars, allowing it to fall within the project target, even with the additional expense of a customized battery. The battery is easy to solder to the platform thanks to the pin layout, and the battery management circuit is integrated on the board, which allows recharge via a dedicated USB-C port.

Table 4.1: Comprehensive comparison of candidate hardware platforms against the functional and non-functional requirements of the Olympia Project.

Parameter / Requirement	ESP32-based platform	Arduino Nano 33 BLE Sense	CodeCell C3	Seeed XIAO nRF52840 Sense Plus
A. MCU architecture and computational resources				
MCU architecture	Xtensa LX6/LX7 or RISC-V	ARM Cortex-M4F	ESP32-C3 RISC-V	ARM Cortex-M4F
Max clock frequency	160–240 MHz	64 MHz	160 MHz	64 MHz
Flash memory	2–16 MB	1 MB	4 MB	1 MB
SRAM	Hundreds kB	256 kB	400kB	256 kB
Floating Point Unit (FPU)	Yes (many variants)	Yes	Yes	Yes
B. Inertial sensing				
Onboard IMU	No (external)	Yes	Yes	Yes
IMU model	External required	LSM9DS1 (typical)	BNO085	LSM6DS3TR-C
Degrees of freedom	Depends on sensor	9 DoF	9 DoF	6 DoF
Accel range	Sensor dependent	Up to ±16 g	Up to ±8 g	Up to ±16 g
Gyro range	Sensor dependent	Up to ±2000 dps	Up to ±2000 dps	Up to ±2000 dps
Supported ODR	Sensor dependent	From 14.9 to 952 Hz	From 10 to 400 Hz	From 12.5 Hz to 1.66 kHz
Integration complexity	High	Low	Low	Low
C. Wireless communication				

Continued on next page

Parameter / Requirement	ESP32-based platform	Arduino Nano 33 BLE Sense	CodeCell C3	Seeed XIAO nRF52840 Sense Plus
Wireless technologies	Wi-Fi + BLE	BLE + NFC	Wi-Fi + BLE	BLE + NFC
BLE version	4.x / 5.x	5	5.x	5.4
Typical radio current	High (Wi-Fi)	Low (BLE)	High (Wi-Fi)	Low (BLE)
Smartphone compatibility	Yes	Yes	Yes	Yes
D. Power consumption characteristics (indicative)				
Active current (MCU + radio)	High	Moderate	Low	Low
Idle / light sleep	Moderate–low	Low	Moderate	Very low
Deep sleep / system off	Low (chip-level); board may dominate	Very low (chip-level); board may dominate	Low	Ultra-low
E. Power management and battery support				
Native Li-Po support	Board dependent	No	Yes	Yes
Onboard battery charger	Board dependent	No	Yes	Yes
USB charging	Board dependent	External	Yes	Yes
External PMIC required	Often yes	Often yes	No	No
F. Form factor and mechanical integration				

Continued on next page

Parameter / Requirement	ESP32-based platform	Arduino Nano 33 BLE Sense	CodeCell C3	Seeed XIAO nRF52840 Sense Plus
Board dimensions (L×W)	Variable	~45×18 mm	~18.5×18.5 mm (+ antenna)	~21×17.5 mm
Board thickness (PCB only)	Typically 1.0–1.6 mm[1]	Typically 1.0–1.6 mm[1]	9.4 mm (with USB port) mm[1]	Typically 1.0–1.6 mm[1]
Prosthesis integration suitability	Often limited	Medium	Good	Excellent
G. Cost and availability				
Typical unit cost	Low–medium	Medium–high	Medium	Excellent
Compliance with 30–40 target	Variable	Often no	Yes	Yes
Commercial availability	High	High	Medium	High
H. Firmware development ecosystem				
Frameworks	ESP-IDF, Arduino	Arduino	Arduino	Arduino, Nordic ecosystem
Docs/community	Very large	Very large	Small	Large
I. Overall evaluation				
Functional requirements	Partial	Good	Good	Very good

Continued on next page

Parameter / Requirement	**ESP32-based platform**	**Arduino Nano 33 BLE Sense**	**CodeCell C3**	**Seeed XIAO nRF52840 Sense Plus**
Non-functional requirements	Low	Medium	High	Very high
Overall suitability (Olympia)	Low–medium	Medium	Medium–High	**Very high**

[1]PCB-only thickness is rarely specified in board datasheets; values shown are typical FR-4 PCB thickness ranges. If required, thickness should be measured or obtained from manufacturer mechanical drawings.

4.3 Final Hardware Selection

Comparing the platforms described in the previous sections, it can be concluded that there is no single best platform in absolute terms. The choice must be guided by requirements and constraints defined by the project.

ESP32-based solutions offer the best computational resources and the most extensive connectivity, but at the expense of power consumption that is incompatible with the goal of multi-year autonomy. The integration of Wi-Fi, in this specific application context, is penalizing from an energy point of view. A similar argument applies to the CodeCell C3 which, despite its excellent form factor and good sensor integration, has significantly higher power consumption in sleep mode than nRF52-based solutions, making it less suitable for this specific case.

The Arduino Nano 33 BLE Sense guarantees a low-power architecture and benefits from a very mature development ecosystem. However, its less compact form factor, and above all its cost and the absence of an integrated battery management circuit, make other solutions based on nRF52 architecture more attractive.

In fact, the "Seeed Studio XIAO nRF52840 Sense Plus" offers the best compromise between form factor, power consumption, sensor integration, interrogability, and unit cost. In addition, the mature development ecosystem, the easy availability of the device, and the on-board battery management circuit allow it not only to simultaneously meet the energy, mechanical, and economic constraints of the project, but also to do so while ensuring simplicity of development and design flexibility with a view to future developments.

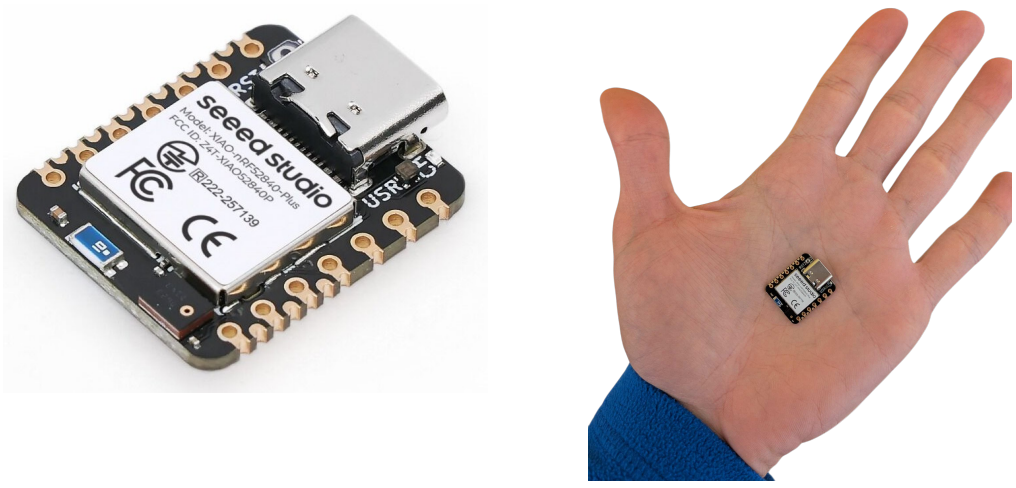

Figure 4.1: Seeed XIAO nRF52840 Sense Plus microcontroller. Left: device view; right: size comparison when held in hand.

4.4 Battery Selection and Power Supply Considerations

The choice of battery is a design phase closely linked to that of the selected hardware platform. Given the ambitious goal of multi-year autonomy for the device, the idea was, once compatibility with the Seeed XIAO nRF52840 Sense Plus was guaranteed, to select the battery with the highest capacity that would fit within the device's shape constraints, as established in Chapter 3.

Considering the negligible thickness of the platform, it was therefore possible to use the entire available space to integrate a 1100 mAh Li-Po 102050 battery, while remaining within the 2 cm width and 1 cm thickness limits.

The choice of a rechargeable lithium-ion battery, in particular, is motivated by:

- high energy density compared to alternative technologies;
- compatibility with the integrated charging circuit on the XIAO;
- commercial availability in thin and integrable formats.

The theoretical target of 36 months of autonomy, given the nominal capacity of the selected battery, would require an average current consumption of less than 42 μA.

Although this imposes demanding requirements on duty cycling strategies, aggressive optimization, and excellent firmware, the compromise identified represents the best balance between operational autonomy, dimensional constraints, safety, and cost, consistent with the objectives of the Olympia Project.

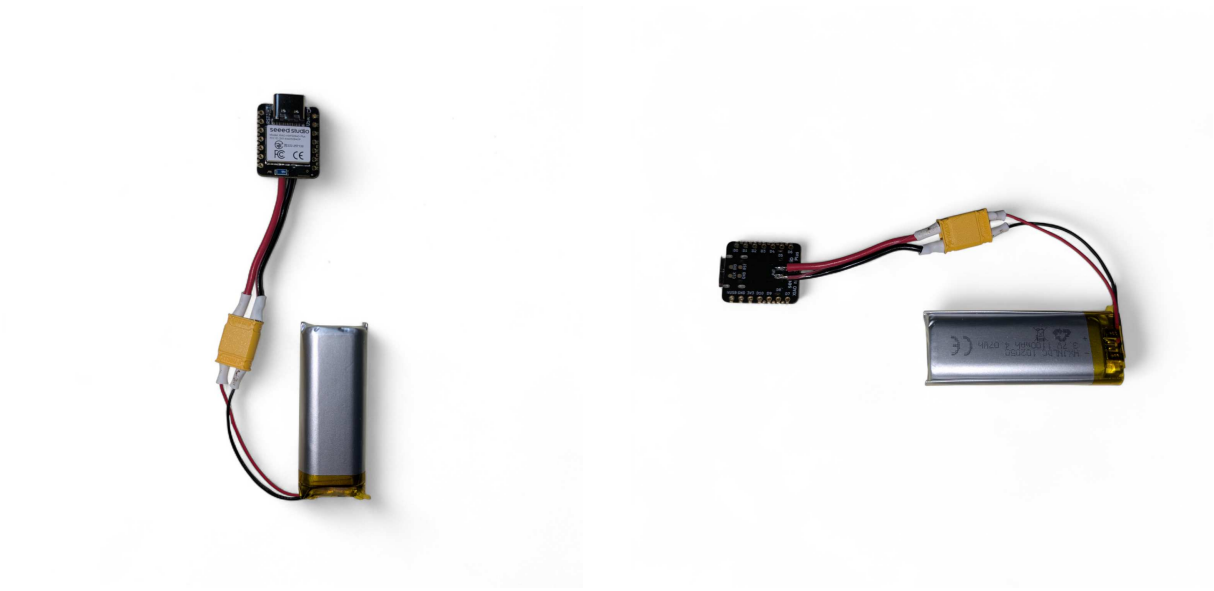

Figure 4.2: Device hardware: Seeed XIAO nRF52840 Sense Plus and 1100 mAh Li-Po battery connected via connector (two views).

Chapter 5

Firmware Architecture and Embedded Implementation

In Chapter 4 a careful selection of the device hardware components was made, establishing the best starting point for the development of the wearable device.

However, it is with the design of the firmware that the functional and non-functional requirements are met, transforming the platform into a true embedded system, ready for the next phase of experimentation of the Olympia Project.

In this regard, the work was not a simple transposition of the code developed in an offline environment onto the microcontroller, requiring instead adaptation to the operating conditions of edge computing, where data acquisition and processing are performed locally on the embedded device rather than on an external computing platform [24].

In particular, the algorithms for activity classification and step counting, which until now had no constraints in terms of memory, latency, and energy, were revised in order to offer better management of these factors, which now had to be limited, while preserving the reliability of the system.

At the same time, a modular firmware architecture was designed to support different operating scenarios. The primary goal was to identify the consumption associated with each component of the system under different operating conditions in order to obtain a realistic quantitative assessment that could be associated with each individual block.

The integration of BLE communication was a central element of the work, not only as a data transmission channel but also as a tool for remote system querying and configuration [24].

This chapter describes the firmware architecture in terms of its experimental and implementation purposes, focusing on movement analysis, wireless communication, and device energy management [25], in accordance with the established application requirements.

5.1 Data Processing and Management Pipeline

Before going into the technical details of the implemented firmware structure, it is important to visualize the logical flow of activity from sampling acceleration signals to the delivery of biomechanical features to the user. Throughout this chapter, several scenarios will be proposed, developed for different operating contexts, which may differ or partially adhere to some of the stages described below. The flow described covers the most relevant aspects of the firmware from a functionality point of view. The process can be read in five consecutive stages, each structured within the limits of the developed "edge" context.

1) Sensing and Normalization: the first stage involves interfacing with the LSM6DS3 IMU. The accelerometer data are sampled and immediately processed in order to extract its magnitude. This choice demonstrates robustness with respect to the physical placement of the sensor on the prosthesis, which, in calculating the magnitude, maintains algorithmic invariance with respect to how it is oriented.

2) Circular Buffering and Windowing: before being processed, each sampled data point is placed in a circular buffer. In this way, the RAM maintains an observation window of 48 samples, 24 of which are subject to overlapping, so as to ensure both stability and continuous updating of the information.

3) Feature extraction: once the time window has been structured, the calculation phase takes place. Features are extracted by working only in the time domain, avoiding complex processing in the frequency domain, and are based on variance calculation. At this stage, the raw data is associated, through light processing, with information representative from a locomotion point of view.

4) Classification: the locomotive descriptors extracted from the previous stage determine the athlete's activity status and detect the steps taken, using dynamic thresholds. Adaptive refractoriness-based logic is used to ensure the extraction of accurate and causal outputs.

5) Energy management and communication: based on the results of the previous stage, the firmware decides whether to keep the system in active mode or to start the transition to a throttling or sleep state. At the same time, the BLE module transmits the results obtained by the XIAO to the mobile device, completing the data flow logic.

5.2 Firmware Architecture Overview

The firmware architecture was structured by separating the logic that implements the algorithms from the blocks that provide functionality to the system. The goal was to achieve a good level of modularity, facilitate debugging and enable accurate consumption analysis.

In this regard, the structure is divided into independent modules:

- an acquisition module, responsible for configuring and reading IMU data. It must ensure temporal consistency and correct data buffering.
- A processing module, which implements the activity classification algorithms and the step counter. This module operates on time windows of acquired data.
- A communication module, which manages the Bluetooth Low Energy stack. It handles both the transmission of information and the remote configuration or modification of operating parameters.
- An energy management module, which manages the selective activation and deactivation of subsystems, optimizing consumption according to specific application requirements.

The behavior of the firmware can be described according to a finite state model, which includes the conditions of:

1. **active acquisition**;
2. **low-power inactivity (throttling)**;
3. **deep idle**.

Transitions between system states are determined by motion detection, the passage of time without any activity being detected, or requests transmitted by the BLE channel. These are presented in the diagram of Fig. 5.1.

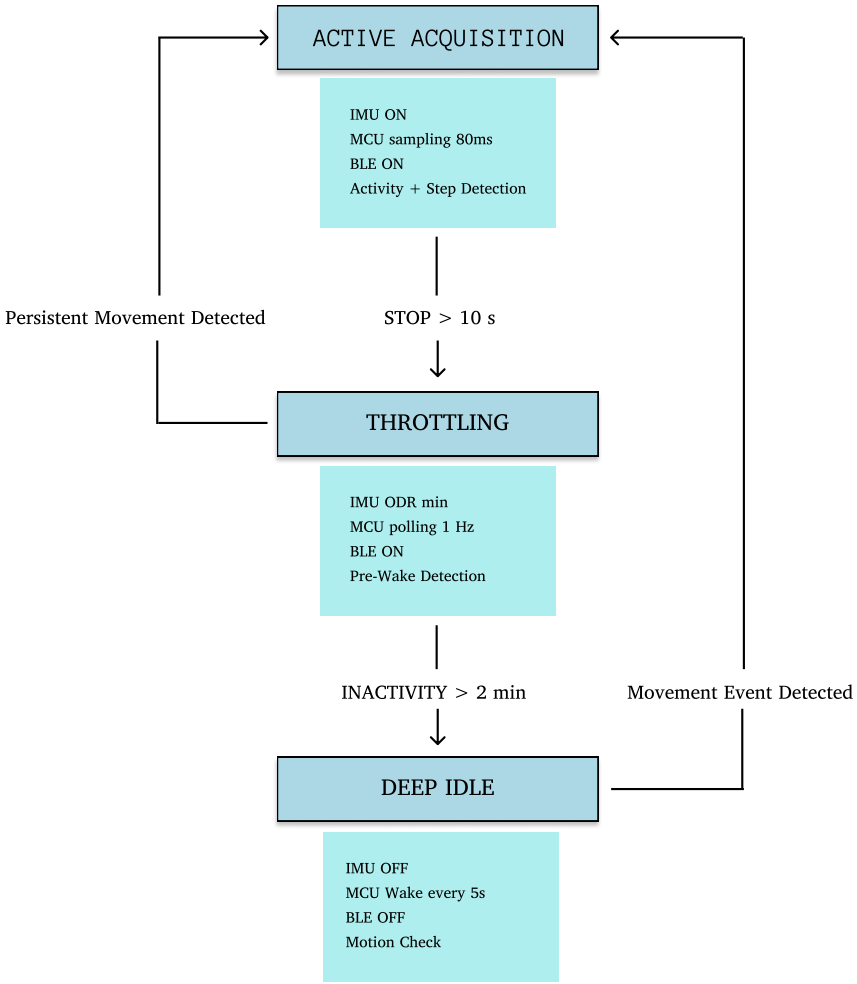

Figure 5.1: Finite state machine of the firmware: the system cycles between active acquisition, throttling, and deep idle states. Transitions are triggered by motion detection or elapsed inactivity.

This organization facilitates the analysis of the energy contributions associated with each module and state, allowing for effective analysis of the system, which will be discussed in more detail in the following paragraphs.

5.3 Transposition of Algorithms into an Embedded Environment

The work of transposing the algorithms from an offline context to operation on a microcontroller was not a simple transfer of code, but rather a readjustment of the code to the new requirements and the new environment under consideration. The availability of limited computational resources and the requirement for real-time processing necessitated specific modifications that took into account, among other factors, the available energy budget.

These modifications resulted in a complete conceptual overhaul of the algorithm, which ensured that the basic logic was retained while adapting the operating modes.

5.3.1 Activity Classification

As described in Chapter 2, the classification of activities in an offline context used tools based on complex and energy-intensive operations (CWT) and the ability to analyze complete datasets. The computational complexity and dependence on extended time windows that characterize this approach guarantee excellent performance in terms of accuracy, but are unsustainable on a microcontroller without significant compromises in terms of power consumption and device responsiveness.

The idea from a firmware-embedded perspective was to replace the more costly tools with "smart" features, which are lighter to implement. In this case, compatible with fixed-size circular buffer structures.

The accelerometer signals are managed by the circular buffers apBuf and ccBuf, in time windows of 48 samples (WINDOW_SIZE) and update steps of 24 samples (WINDOW_STEP). In this case too, 50% overlapping guarantees classification updates every 1.92 s (ODR = 12.5 Hz), offering a good compromise between system responsiveness and statistical stability of the extracted information.

By analyzing the variance of the AP signal and measuring the intensity of the vertical CC component, a new decision tree was proposed that was deterministic, computationally lightweight, compatible with periodic execution on a sliding window, and minimized energy consumption:

```
if (ccMax < T1_CC_STOP) return STOP;
else if (apMax < AP_MIN_LOCOMOTION) return NON_LOCOMOTION;
float varAP = variance(apBuf, WINDOW_SIZE);
if (varAP < T2_WALK_JOG)   return WALK;
else if (varAP < T3_JOG_SPRINT) return JOG;
else return SPRINT;
```

In this case too, the decision structure is hierarchical, with logic based on physically interpretable thresholds that allow the distinction between STOP, WALK, JOG, and SPRINT activities.

Here, the SWING activity has been classified as NON_LOCOMOTION and falls within the set of movements that do not show significant acceleration in the direction of motion, therefore classified as "not useful" for work purposes.

The most significant change, compared to the approach adopted in the offline context, is due to the fact that classification is now based on variance calculation (variance()). The decision tree first evaluates the vertical component of motion (ccMax) to identify the STOP state in cases where a minimum threshold is not detected on the respective axis. At this point, once the application cases below a certain front-to-back acceleration threshold have been eliminated, the entire classification is based precisely on the variance of the component detected on this axis (varAP). There is no longer any need for complex transforms, as required by the implementation of CWT, but only sum and square operations, which are well compatible and optimized for the device's ARM Cortex-M4F core.

It should be noted that the proposed implementation choice is not based on a quantitative comparison between the classification accuracy achieved using the reference approach and the method adopted in this work. The main objective is, in fact, the development of a system capable of operating under constraints that are significantly different from those of the AX6-based offline framework, particularly in terms of energy consumption, computational complexity, and real-time execution. Within this context, the adoption of a lightweight algorithm represents a reasonable first step toward ensuring the coexistence of functional and energy requirements. Furthermore, the system's ultimate goal is to estimate the prosthesis's wear state. At the current stage, despite the adoption of appropriate signal analysis techniques, the relationship between the measured signals and their impact on prosthesis durability has not yet been fully quantified. For this reason, a strict comparison between different signal processing approaches would be premature and potentially misaligned with the overall objectives of the project.

The accuracy of this classification is closely linked to the definition of the decision tree thresholds. These, determined through MCCV in an offline context, depend on the individual athlete and can vary depending on various factors, from stride style to the different mechanical characteristics of the prosthesis. The firmware is designed to allow these thresholds to be set remotely, without the need for firmware reprogramming and without having to physically access the device.

The elimination of the manual labeling procedure, based on video sessions, and of MCCV, requires new strategies for calibrating locomotor thresholds, which are calculated based on the specifications of the individual reference athlete [26].

A simple guided protocol, that athletes can follow during the initial warm-up phase, is proposed:

1. 10 seconds of STOP
2. 30 seconds of WALK
3. 30 seconds of JOG
4. 15 seconds of SPRINT

This allows the following values to be determined:

- var_AP_walk
- var_AP_jog
- var_AP_sprint
- CC_max_stop
- CC_walk_min

The decision tree thresholds are determined accordingly:

- T1= (CC_max_stop + CC_walk_min)/2
- T2= (var_AP_walk + var_AP_jog)/2
- T3= (var_AP_jog + var_AP_sprint)/2

This approach replaces the static threshold definition derived from offline analysis with a semi-adaptive configuration, enabling a basic level of personalization while maintaining negligible computational overhead. As a result, the classifier can better account for inter-subject variability without compromising the energy and resource constraints of the embedded system.

5.3.2 Step Counting

The original approach is based on estimating the average cadence over complete movement intervals and requires access to the entire time series for the interval considered. This introduces a non-causal behavior into the algorithm, making it suitable for offline analysis but not directly implementable in a real-time embedded context, where decisions must be made incrementally, sample by sample, with limited computational and memory resources.

The algorithm designed to identify the correct step count has been redesigned, no longer with a view to estimating it, but rather to count individual steps. The count is based on the acceleration values at a given moment, compared to the average of the current window, then compared with a predefined threshold. This change represents a trade-off between statistical robustness achievable through batch processing of full windows and the need to ensure causality and real-time responsiveness. In particular, the event-based approach allows for continuous operation, at the cost of greater sensitivity to noise and signal variability.

On the other hand, the greater variability in acceleration patterns associated with locomotion with prostheses reduces the reliability of estimates based on average cadence values. This factor supports the decision to adopt an approach based on local events, which is less dependent on assumptions regarding the periodicity and regularity of the signal. The code below shows the logic of the step detection algorithm:

```
if((totalAcc - meanTotalAcc) > STEP_THRESHOLD &&
   (sampleCounter - lastStepSample) >= minS) {
    lastStepSample = sampleCounter;
    steps_total++;
    }
```

Here, the parameter `minS` corresponds to the refractory period, i.e., the minimum number of samples that must elapse between two consecutive valid step detections. The variable `sampleCounter` represents the current time reference expressed in samples, i.e., the instant at which a potential new step is being evaluated. Finally, `LastStepSample` stores the time (in samples) at which the last valid step was detected. Based on these definitions, a new step is validated only if the difference between `sampleCounter` and `LastStepSample` exceeds `minS`, ensuring temporal consistency and preventing multiple detections of the same physical step.

There are two aspects to pay particular attention to:

1. The count is taken into consideration if and only if the current activity is locomotor. Once the step is detected, both the total step counter and the current activity counter are incremented.

```
if(!isLocomotion(currentActivity)) return;
```

2. A dynamic Refractory Period is defined, i.e., depending on the type of activity detected:

```
uint8_t refractorySamples(Activity a) {
  if(a==WALK) return 5;
  if(a==JOG)  return 3;
  if(a==SPRINT) return 2;
  return 255;
}
```

The system adjusts the waiting time between two steps based on the detected activity:

- WALK: 5 samples (minS ~400 ms)
- JOG: 3 samples (minS ~240 ms)
- SPRINT: 2 samples (minS ~160 ms)

This avoids high-frequency double counting or missing significant detections due to excessively long periods. The refractory period is scalable with the set sampling frequency, therefore to the Output Data Rate (ODR) value of the IMU.

In summary, the logic according to which step counting follows activity classification has been preserved, using the same overlap of time windows. The need to make everything work in real-time precludes the manual removal of outliers and the analysis of entire time series, which was previously adopted. Nevertheless, greater causal consistency of the step detection algorithm with respect to the system behavior was enforced and, as the following paragraphs will show, the possibility of retroactive calibration of thresholds was maintained through the implementation of the BLE communication module.

5.4 Operational Scenarios and Energy Consumption Modeling

During the definition of objectives, the Prosthetic Center expressed a desire to understand the energy behavior of the device's various sensor configurations.

In particular, a quantitative comparison of consumption attributable to accelerometers and gyroscopes set at different frequencies would make it possible to guide algorithmic implementation choices retrospectively, favoring algorithms based on sensors associated with lower energy consumption, while maintaining the same level of accuracy guaranteed by the sensors themselves.

The Prosthetic Center, in fact, uses algorithms based either solely on accelerometer data or solely on gyroscope data. Assuming that these are equally accurate, it is possible to prioritize the use of the algorithm based on the sensor system that requires less energy.

In addition to this energy comparison, there is also a need to compare raw data transmission and consumption due to on-board processing, which requires a quantitative comparison between different functional blocks that are mutually exclusive.

5.4.1 Definition of Scenarios

The questions of greatest interest concern the energy impact of introducing the gyroscope, that is not employed at present, the reduction in consumption achievable by reducing BLE traffic, and the variation in the sampling frequency of the IMU blocks.

In this regard, four comparative operating scenarios were implemented in the firmware to acquire useful information, plus one designed as an optimized scenario for the daily operation of the device.

A runtime parameterization was proposed, through enumeration, which would allow a single firmware base to be maintained but, above all, a set-up that was not subject to variations during measurement, ensuring comparability thanks to the possibility of varying the scenario via BLE.

Specifically, the comparison scenarios reported are:

1. **Accelerometer only** (raw data): the device acquires and transmits raw data from the triaxial accelerometer, configured with a full scale of ±16 g. This configuration provides a baseline description of motion based only on linear acceleration, with limited computational and transmission requirements.

2. **Accelerometer and gyroscope** (raw data): the device acquires and transmits raw data from both the triaxial accelerometer and the gyroscope. While accelerometer data are collected on all three axes, only a single gyroscope axis is considered. This axis is selected based on its biomechanical relevance, as it captures the dominant rotational component during locomotion. The gyroscope operates with a full scale of ±2000 °/s, in order to avoid saturation during more dynamic activities such as jogging and sprinting. Although this choice reduces measurement resolution, it ensures reliable acquisition across different movement intensities. The combined use of accelerometer and gyroscope signals provides complementary information on linear and rotational motion. At the same time, limiting the gyroscope to a single axis reduces both computational requirements and BLE transmission load.

3. **Step counting** (accelerometer only): accelerometer data is processed on-board to estimate the step count for each activity. In this sub-scenario the gyroscope is not used and BLE

transmission is limited to processed data. This configuration limits data transmission and reflects an embedded processing approach, where only the estimated step count is communicated.

4. **Single-axis gyroscope** (raw data): the device acquires and transmits raw data from a single axis of the gyroscope, without using the accelerometer. This sub-scenario is motivated by the availability of an already validated algorithm, which operates effectively on a single gyroscopic signal and is particularly suitable for acquisition sessions of limited duration.

```
enum ScenarioMode : uint8_t {
  S2_ACCEL_RAW   = 1,
  S2_IMU_RAW     = 2,
  S2_STEPCNT     = 3,
  S2_GYRO_MONO   = 4,
  S5_RUN_DEFAULT = 5
};
```

As mentioned above, there will be two types of comparisons between scenarios:

- between sensor blocks
- between continuous streaming and event-driven communication

Scenarios 1), 2), and 4) involve a BLE notification for each sample acquired. In these cases of continuous streaming, $f_{\text{BLE}} = f_{\text{sampling}}$.

Scenario 3), on the other hand, involves event-driven communication, transmitting only when there is a change of activity or step detection.

This methodology exploits the comparison between a demanding condition from a radio point of view and embedded processing to analyze the energy consumption of both and understand how they impact battery life.

5.4.2 Scenario 5: Demonstration Firmware

Scenario 5 was implemented in order to provide an example of code that simulates realistic use of the device within the Olympia Project.

This required customized energy optimization to adapt to the temporal asymmetry of device use and the different requirements that a real-world context may demand. The underlying philosophy was to overcome the logic of continuous streaming and fixed-frequency sampling, taking care in the management of system resources.

The adaptation of the IMU sampling frequency, the reduction of the CPU duty cycle, event-driven BLE communication, and the introduction of sleep states are some of the functions implemented, aimed at achieving the best possible level of autonomy without compromising performance.

Before going into detail about the architecture of this scenario, it is important to highlight the central point of the application context. The sports prosthesis is used during daily training sessions lasting no more than 2 hours. Therefore, the responsiveness required of the device during this time period is greater than during the remaining 22 hours of the day. The obvious goal is to drastically reduce consumption during prolonged periods of inactivity, thanks to appropriate measures.

Architecture

Scenario 5 integrates:

- accelerometer acquisition;
- activity classification;
- step counting;
- dynamic BLE management;
- time throttling;
- sleep state management.

We can see scenario 5 as a state machine with three states having different energy levels:

1. Active Mode
2. Throttling Mode
3. Deep Idle

1) Active Mode

When motion is detected, the IMU operates at nominal frequency (defined in a range between 12.5 Hz and 416 Hz). The MCU performs a sampling cycle every 80 ms and, between one sample and the next, enters light sleep mode via wait-for-event. In this mode, the MCU executes a WFE instruction, which puts the processor into a low-power state until an event occurs (such as new data from the IMU), thereby reducing the CPU duty cycle.

```
static inline void mcuWaitForEvent() { __WFE(); }
```

During this phase, the MCU collects data from the IMU at each sampling instant and transmits it over BLE according to the selected scenario. The MCU transmits notifications related to the detection of a step or a change of energy state.

2) Throttling Mode

If no significant movement is detected for 10 consecutive seconds, the system enters a STOP throttling phase: the MCU further reduces its activity by switching to 1 Hz polling, while the IMU remains active at the minimum practical ODR (12.5 Hz), allowing the detection of a possible restart without the need to revert to active mode. In this phase, consumption is reduced mainly by adjusting the wake-up frequency of the MCU, while maintaining the ability to detect movement.

```
#define S5_STOP_ENTER_MS          10000UL
#define S5_SAMPLE_ACTIVE_MS       80
#define S5_SAMPLE_STOP_IDLE_MS    1000
```

The timing parameters governing this transition are defined in the firmware through a set of constants. In particular, `S5_STOP_ENTER_MS` sets the inactivity duration required to enter throttling mode, while `S5_SAMPLE_ACTIVE_MS` and `S5_SAMPLE_STOP_IDLE_MS` define the sampling period in active and throttling conditions, respectively. This results in a transition from a nominal sampling period of 80 ms (12.5 Hz) to 1000 ms (1 Hz) when the system enters the low-power throttled state.

3) Deep Idle

If the idle state persists for a longer period of time (configurable value), the system enters deep idle: BLE communication is completely disabled and the IMU is turned off, eliminating the main energy contributions during long periods of non-use. In this mode, the MCU wakes up periodically, every `S5_DEEP_CHECK_PERIOD_MS` (5 s), and temporarily reactivates the IMU for a control window of approximately 200 ms to check for movement. The system enters deep idle after a continuous inactivity period defined by `S5_DEEP_IDLE_ENTER_MS` (2 min). The firmware evaluates the change in acceleration (“jerk”) to decide whether to reset the system.

```
#define S5_DEEP_IDLE_ENTER_MS     (2UL*60UL*1000UL)
#define S5_DEEP_CHECK_PERIOD_MS   5000
```

If persistent movement is detected during one of these windows, the system exits deep idle and automatically restores active mode, reactivating the IMU, MCU at 12.5 Hz, and BLE communication. During this transition, internal states such as data buffers and counters are reset to prevent residual or outdated information from causing incorrect behavior or false re-entries. This strategy minimizes energy consumption during long periods of inactivity while ensuring reliable and deterministic system reactivation.

5.5 Wireless Communication

As defined in the requirements set out in Chapter 3, the device must guarantee wireless communication, allowing the system to be queried without the need to transfer data via cables.

This requirement not only ensures greater convenience in using the prosthesis during analysis but also allows the device to be integrated where this would not be possible if it were necessary to physically access it.

The BLE module is implemented through a dedicated stepService BLE service, a software entity that organizes the data exchanged over Bluetooth. The MCU acts as a BLE peripheral, exposing services and characteristics that can be accessed by a remote client (smartphone or tablet), which functions as the central device. This architecture allows both data acquisition and dynamic configuration without requiring physical connection: on one hand, the device continuously collects and transmits measurement data to the client; on the other hand, it allows the client to modify system parameters at runtime through dedicated configuration commands. The stepService is responsible for handling all step and activity related data, and it provides two fundamental features that make it a key element of the wearable system:

1. Read | Notify: this property allows data to be transmitted to the remote device, ensuring that the device can be queried. It is an output channel that transmits raw IMU data, activity events, and step counts.

2. Write: this property allows runtime configuration. It is a remote control channel that allows to change the sampling frequency, threshold, operating scenarios, and BLE enablement.

Specifically, the BLE service includes:

- `dataCharacteristic`: used to send measurement data such as acceleration, steps, and activity events. Configured with `BLERead | BLENotify`, it allows the client to read values on demand or receive automatic notifications.

- `configCharacteristic`: used for runtime configuration. Configured with `BLEWrite`, it allows the client to send configuration packets to the MCU, changing parameters like sampling frequency, thresholds, operating scenarios, or enabling/disabling BLE.

The MCU writes new measurement data to `dataCharacteristic`, while the client reads it. Conversely, the client writes configuration packets to `configCharacteristic`, which the MCU reads and applies immediately.

The behavior of the device depends on the selected operating scenario. These scenarios determine whether data are transmitted continuously or only when relevant events occur, and they also define how IMU measurements are processed. In raw scenarios (S2_ACCEL_RAW, S2_IMU_RAW, S2_GYRO_MONO), data are transmitted continuously, with a transmission rate equal to the IMU sampling frequency. In event-driven scenarios (S2_STEPCNT and S5_RUN_DEFAULT), instead, data are sent only when a change in activity or a step is detected.

In both cases, `dataCharacteristic` is used to transmit the information generated by the system. Depending on the active scenario, this may correspond either to a continuous stream of IMU samples or to discrete updates associated with detected events.

The operating scenario, together with other parameters such as sampling frequency, detection thresholds, and BLE enablement, is configured through `configCharacteristic`. This allows the client to modify the system behavior at runtime without requiring firmware updates.

Configuration is performed by sending packets from the client to the MCU. Each packet contains a parameter identifier and an associated value, so that specific parameters can be updated independently with minimal communication overhead.

The configuration packets sent through `configCharacteristic` reference specific system parameters using predefined identifiers. These identifiers are defined in the firmware as an enumeration:

```
enum ConfigParam : uint8_t {
  CFG_SAMPLE_PERIOD = 1,
  CFG_AP_MIN        = 2,
  CFG_T2_WJ         = 3,
  CFG_T3_JS         = 4,
  CFG_STEP_THR      = 5,
  CFG_SCENARIO      = 6,
  CFG_GYRO_AXIS     = 7,
  CFG_BLE_ENABLE    = 8
};
```

Each value corresponds to a configurable parameter: for example, `CFG_SAMPLE_PERIOD` controls the sampling period, `CFG_STEP_THR` the step detection threshold, and `CFG_BLE_ENABLE` allows enabling or disabling BLE communication. Using this approach, the client can selectively update individual parameters with minimal communication overhead.

The BLE module also contributes to energy management. During prolonged inactivity, communication can be disabled to reduce power consumption, depending on the system energy state. The firmware periodically checks for activity and re-enables BLE communication when movement is detected.

5.6 Battery level

The power management and monitoring subsystem is based on the hardware design of the Seeed XIAO nRF52840 Sense Plus. The system integrates a charging circuit and a voltage-sensing network designed for ultra-low-power operation.

Hardware Configuration

The battery charging is handled by the Texas Instruments BQ25101, a linear charger optimized for small Li-Po cells. To monitor the state of charge (SoC), the board implements a voltage divider connected to the analog pin `P0.31` (ADC input).

The divider consists of a $1\,M\Omega$ high-side resistor ($R16$) and a $510\,k\Omega$ low-side resistor ($R17$). This configuration reduces the battery voltage (up to $4.2\,V$) to a range compatible with the internal $3.6\,V$ reference of the nRF52840 ADC.
The attenuation factor is defined by:

$$V_{ADC} = V_{BAT} \times \frac{R_{bottom}}{R_{top} + R_{bottom}} = V_{BAT} \times \frac{510\,k\Omega}{1\,M\Omega + 510\,k\Omega} \approx \frac{V_{BAT}}{2.9608} \tag{5.1}$$

To prevent constant current leakage through the divider, the hardware employs a power gating strategy. The bottom of the divider is connected to GPIO `P0.14` (`PIN_VBAT_ENABLE`) rather than directly to ground.

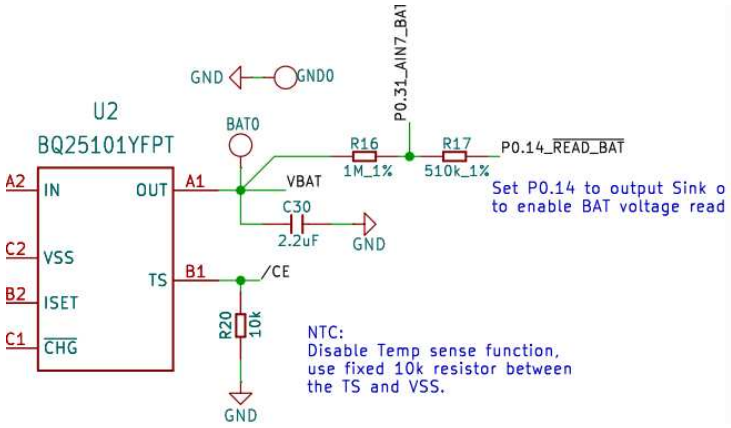

Figure 5.2: Battery charging circuit schematic: BQ25101 linear charger with voltage divider 1 MΩ / 510 kΩ for SoC monitoring, featuring GPIO-controlled power gating.

Firmware Implementation

The firmware was developed to actively manage the hardware sensing network, ensuring high accuracy while maintaining energy efficiency. The sampling logic is encapsulated in the `readVbat()` function, which follows a specific execution sequence:

1. Dynamic Activation: the `PIN_VBAT_ENABLE` is configured as an output and set to `LOW`. This acts as a current sink, effectively connecting the divider to ground and activating the sensing circuit.

2. Settling Time: a software delay of `3 ms` is introduced. This interval allows the voltage at the ADC node to stabilize, accounting for the RC time constant introduced by the 2.2 μF filter capacitor ($C30$).

3. Oversampling and Filtering: to mitigate thermal noise and electromagnetic interference from the BLE radio, the `adcReadAvg()` function performs a mean of 16 consecutive samples with 12-bit resolution.

4. Voltage Reconstitution: the raw ADC value is converted back to the actual battery voltage by applying the `VBAT_GAIN` constant ($2.9608 f$), which represents the inverse of the hardware divider ratio.

5. Power-Down: once the measurement is complete, the `PIN_VBAT_ENABLE` is reverted to `INPUT` (High-Impedance) mode, physically breaking the path to ground and eliminating quiescent current draw.

Data Processing and BLE Transmission

The calculated voltage is further processed by the `lipoPercentFromV()` function, which utilizes a piecewise linear approximation of the Li-Po discharge curve to estimate the remaining capacity.

The information is then exposed via Bluetooth Low Energy (BLE) using the standard *Battery Service* (UUID `0x180F`). In addition to the standard percentage characteristic (`battLvlChr`), a custom characteristic (`vbatMvChr`) was implemented to provide the precise battery potential in millivolts, facilitating advanced diagnostic monitoring from the central device.

Chapter 6

Experimental Validation

6.1 Experimental Setup

Following the design and development of the device, experimental measurements were carried out to characterize the actual current consumption and make more accurate estimates of battery life. The instrumentation needed to take the measurements was chosen based on the theoretical estimate of the quantities involved. Specifically, the instruments used in this phase of the work were:

- a Keithley 6485 picoammeter;
- an Agilent E3631A bench power supply;
- BNC coaxial cables;
- connections to the DUT.

To perform the current measurements in question, it was deemed functional to disconnect the battery in order to replace it with the Agilent DC power supply [27]. This ensures a perfectly stable voltage, allowing the measurements of interest to be taken. The picoammeter, on the other hand, is connected in series on the positive side of the power supply and its output goes to the positive terminal of the DUT. Current flows in the DUT and the loop is closed on a shared common ground node. Since the currents involved are in the μA range, the BNC connection is insulated with tape. The Keithley 6485 [28] introduces a very low burden voltage and therefore does not significantly alter the DUT's supply voltage. This setup avoids parasitic currents, capacitive coupling, and noise.

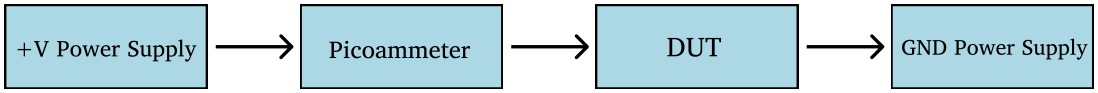

Figure 6.1: Experimental setup for current measurement of the DUT.

It is important to clarify that for the purposes of our project, the focus is on average current measurements and characterization deals with long-term behavior. This type of setup would not be suitable for characterizing instantaneous dynamics, detecting BLE current peaks or analyzing fast transients.

6.2 Operating scenarios

After setting up the measurement setup, we proceeded to test the scenarios presented in chapter 5. In order to obtain comparable and repeatable measurements without physical changes to the system configuration, all the functions that guaranteed the change of scenario and relevant parameters during comparison via BLE were implemented by firmware programming. Using the "nRF Connect" application, it was possible to use the WRITE function to select the payload identifying the scenario and sampling frequency of interest. In particular, once the device was scanned and connected, it was sufficient to follow these sequential steps on the mobile device:

1. select WRITE;
2. select format BYTE ARRAY;
3. enter the specific payload (5 bytes);
4. press SEND.

Below are the tables with the payloads for the selected scenarios and sampling frequencies:

Frequency (Hz)	**Period (ms)**	**Final payload (Byte)**				
		B1	B2	B3	B4	B5
12.5	80	01	00	00	A0	42
25	40	01	00	00	20	42
50	20	01	00	00	A0	41
104	10	01	00	00	20	41
208	5	01	00	00	A0	40
416	2	01	00	00	00	40

Table 6.1: Relationship between sampling frequency, period, and BLE payload (little-endian float)

ID	**Modalità**	**Payload (Byte)**				
		B1	B2	B3	B4	B5
1	S2-ACCEL-RAW	06	00	00	80	3F
2	S2-IMU-RAW	06	00	00	00	40
3	S2-STEP-INT	06	00	00	40	40
4	S2-GYRO-NORM	06	00	00	80	40
5	S5-RUN-DEFAULT	06	00	00	A0	40

Table 6.2: Scenario 1 - Operating configurations and related BLE payload (5 bytes)

6.3 Average current measurements

Once the setup was complete, measurements were taken for each scenario-frequency configuration, varying the scenarios according to ascending order and, within each scenario, varying the frequency in descending order from 416 Hz to 12.5 Hz. During the transition from one scenario to another, the system was allowed to stabilize for a certain period of time in order to obtain the average steady-state current value, which was displayed on the picoammeter.

Scenario	**12.5Hz**	**25 Hz**	**50 Hz**	**104 Hz**	**208 Hz**	**416 Hz**
S2_ACCEL_RAW	690 μA	744 μA	837 μA	1.01 mA	1.39 mA	2.52 mA
S2_IMU_RAW	725 μA	795 μA	950 μA	1.25 mA	1.87 mA	3.42 mA
S2_STEPCNT	681 μA	715 μA	768 μA	872 μA	1.08 mA	1.76 mA
S2_GYRO_MONO	691 μA	730 μA	800 μA	949 μA	1.25 mA	2.15 mA

Table 6.4: Average values of the current absorbed in scenarios S2 as the sampling frequency varies

Analysis of the measured currents shows a close correlation between the increase in sampling frequency and the increase in current values in each of the scenarios considered, as was to be expected. This increase is more pronounced at high frequencies, especially in the S2_IMU_RAW scenario, where the average current settles at 3.42 mA. This increase in high-frequency consumption is more pronounced in scenarios characterized by continuous BLE transmission than in the event-driven S2_STEPCNT scenario. The latter, characterized by embedded processing, shows a moderate increase and, even at a sampling frequency of 416 Hz, limits consumption to 1.76 mA. In relative terms, the increase between 12.5 Hz and 416 Hz is approximately:

- ~3.6× for ACCEL_RAW
- ~4.7× for IMU_RAW
- ~2.6× for STEPCNT
- ~3.1× for GYRO_MONO

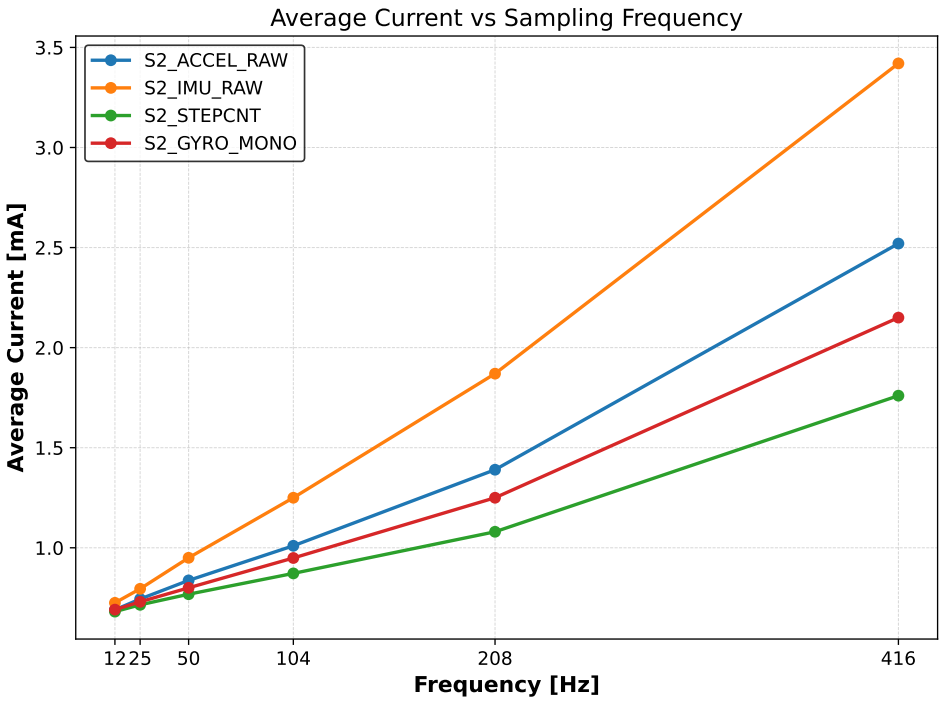

Figure 6.2: Average current consumption in S2 scenarios vs. sampling frequency.

These results show:

- higher energy consumption for continuous transmission configurations, accentuated by the condition imposed in the firmware whereby: $f_{\text{BLE}} = f_{\text{sampling}}$;
- a comparable increase in consumption due to the activation of the accelerometer and gyroscope blocks;
- the interesting prospect of edge processing compared to raw transmission, which shows good prospects for applicability, from an energy point of view, at high frequencies.

6.4 Long-term autonomy

6.4.1 Theoretical estimates

Based on the specifications of the battery used in the device and taking into account the average current measurements taken using a picoammeter, it is possible to estimate the theoretical duration of the device in various "non-optimized" scenarios, i.e., assuming continuous and periodic transmission and sampling at the set nominal frequency. To do this, we start from the reported capacity of the 102050 battery, i.e., 1100 mAh. This means that, if we assume an average current delivery of 1 mA, the total life would be around 1100 h, or approximately 45

days. Following this logic, it is possible to identify the theoretical duration of a complete battery discharge cycle for each of the proposed configurations. If we assume continuous transmission at a nominal frequency of 416 Hz (i.e., $f_{\text{BLE}} = f_{\text{sampling}} = 416Hz$), in the most demanding scenario, represented by S2_IMU_RAW, the battery life should not exceed 14 days. Meanwhile, the same scenario configured at 12.5 Hz should theoretically exceed 63 days. Similar considerations can be made for any scenario, obtaining, in the case of continuous sampling, results that can guarantee durations of up to 67 days in the S2_STEPCNT scenario at 12.5 Hz.

6.4.2 Comparison with empirical measurements

High frequency

In order to verify the reliability of the measurements taken in a real context, a device durability test was carried out. To this end, it was decided to test the system in the "worst-case scenario," i.e., S2_IMU_RAW (theoretical average current: 3.42 mA). Among the reasons for choosing this configuration were the desire to set a lower limit on autonomy and the need to test a scenario that was compatible with the time available for the candidate to present this thesis. First, a complete battery recharge cycle was performed. Thanks to the battery management circuit integrated in the nRF and the presence of a green status LED, it is possible to ensure that the battery was fully charged. At this point, after loading the firmware, the device was inserted into a running "cell phone belt" so that it could be carried continuously and simulate accelerations similar to those encountered during use. Logically, the accelerations of interest in the context of the Olympia project will be very different, but for the purposes of energy profile analysis, keeping the device in motion is sufficient to obtain plausible results.

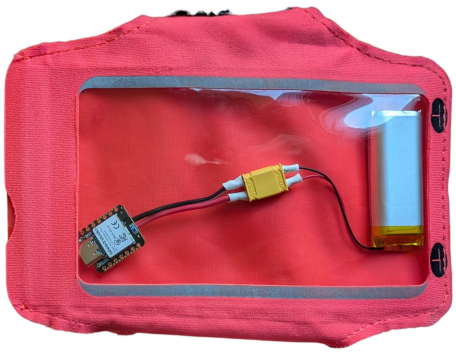

Figure 6.3: Experimental setup showing the wearable system integrated into a running band. The configuration allows the collection of inertial data and battery-powered operation during empirical tests.

Under these operating conditions, the device remained functional for 23 days. This duration would be justified by an average current consumption of about 2 mA, apparently at odds with the 3.42 mA measured in the laboratory. The difference with the theoretical hypothesis of 14 days is to be found in the temporary disconnections between the nRF and the smartphone that occurred during the test. The main causes that can lead to disconnection are:

- radio interference: during the day, the RF environment changes continuously. If the link quality deteriorates for a few seconds, a supervision timeout may occur.
- Aggressive management on BLE: the smartphone (in this case Android OS) can put apps in the background, activate energy saving or reduce BLE priority when the screen turns off.
- Micro power interruptions: an unstable battery, radio peaks or slight voltage drops can cause sporadic resets, which are perceived as disconnections.

These events, combined with specific smartphone user requirements, which have made it difficult to monitor the BLE pairing between the two devices in specific circumstances, justify a hybrid behavior of the system, which sees the device operating in "continuous streaming" mode for an average time interval of between 10 and 12 hours per day, and entering a "disconnected" state for the remaining 12-14 hours of the day. Assuming an average current consumption of 973 μA in the disconnected state for 14 hours per day and the theoretical consumption obtained with the picoammeter in the "connected" state, an average consumption of 1.99 mA would be obtained, consistent with 23 days of operation. The energy limitation linked to BLE data traffic, which reduces consumption to around 1 mA, is reasonable given the assumptions made. In this state, the IMU continues to operate at high frequency but significantly limits the consumption linked to wireless data transmission.

Low Frequency

In order to validate the device's behavior at low frequencies, a second test was performed. In this case, the S2_IMU_RAW was tested at its nominal frequency of 12.5Hz during a reduced portion of the device's discharge cycle. Using the code dedicated to reading the battery charge status, the discharge time of 3-4% of the device's total charge was analyzed. Considering the average consumption of 720 μA for this configuration, the remaining battery charge should have lasted between 45 and 61 hours. The result was consistent with the assumptions made: the device exhausted its remaining charge after 54 hours. Given the short duration of this test, a BLE connection and transmission status was verified throughout the experiment, ensuring accurate results that were not subject to assumptions related to system monitoring.

6.5 Discussion of Results

The comparison between theoretical estimates of battery life based on average current measurements taken with the picoammeter, and inductive tests carried out, shows consistent results, with a clear correlation between an increase in nominal frequency (sampling and transmission) and the system's energy consumption. It has been shown that, depending on the firmware operating mode, continuous BLE transmission operating periods of up to 67 days can be achieved with a single recharge cycle. It is therefore possible to rely on the measurements taken for each scenario in order to get a quantitative idea of the device's battery life. This allows to choose the algorithms and mode that are most appropriate for the use case, with an awareness of the resulting energy profile. It should also be noted that, compared to initial predictions, the use of the low-frequency gyroscope does not have a significant energy impact. This awareness opens the door to the use of gyroscope-based algorithms, already developed in the context of the "Olympia Project," which may lead to interesting implementation developments for biomechanical monitoring purposes.

It is interesting to compare the results obtained for the XIAO with those obtained by the AX6 under similar operating conditions. According to the datasheet, the AX6's battery life, in continuous acquisition mode at 100 Hz and set to 6-axis, is 7 days. Considering that the battery used is 250 mAh, we can deduce that the AX6's average power consumption under these conditions is 1.49 mA. The Seeed XIAO nRF52840 Sense Plus, under the same operating conditions in terms of inertial signal acquisition (IMU_RAW scenario, 100 Hz frequency), shows an average current consumption of 1.25 mA, while simultaneously maintaining BLE transmission active. This comparison not only demonstrates better energy efficiency of the XIAO compared to the AX6, but does so despite the implementation of additional features, such as BLE interrogation capability, which enables real-time usage.

The comparison takes on a different perspective when applied to a 3-axis measurement based solely on the accelerometer. In this case, the AX6 demonstrates a battery life of 17 days, associated with an average current consumption of 613 μA. The XIAO, despite benefiting from the gyroscope block being turned off, shows an average current close to 1 mA, still significantly influenced by the contribution from BLE transmission. To ensure a fair comparison under identical operating conditions, it would be advisable to repeat the measurements in the same scenario and at the same frequency, with BLE disabled on the nRF.

Another major advantage of the XIAO over the AX6 is its cost. The unit price of the AX6 is approximately 140€, which is considered to be relatively inexpensive when compared against devices used in other experimental studies that involve multiple subjects. As a result of having many similar studies conducted, the overall cost to deploy the AX6s will be prohibitively high, especially if they are used on a large-scale or systematic basis. A distinguishing element of the AX6 is that it has no means of access: AX6s will acquire data and store measurements, but

there are no wireless or wired communication interfaces to allow for access to the AX6 for either configuration or retrieval of data during the acquisition phase.

In contrast, the cost of the Seeed XIAO nRF52840 Sense Plus microcontroller alone is approximately 15€, and even after adding a dedicated battery, it remains significantly lower than that of the Axivity. The excellent energy efficiency, optimized form factor, and affordability of this device make it a viable option for the development and expansion of the Olympia Project, which, thanks to additional implementations and cost reductions, could be applied to the monitoring of all types of prosthetics.

The tests and considerations carried out so far are based on continuous use cases, which do not yet assume customized optimizations on the temporal asymmetry of use in the context of Paralympic training. The next chapter proposes an operating model aimed at maximizing system optimization, as a natural development of scenario 5 described above, with a view to achieving multi-year autonomy, under strict energy restrictions on the system.

Chapter 7

Conclusions and Future Developments

7.1 Summary of Contributions

This work has shown the preliminary analysis, design, development, and validation of an embedded system that can be integrated into lower limb sports prostheses for the biomechanical monitoring of Paralympic athletes. The objective of the system is to acquire, process, and transmit inertial data under stringent energy autonomy constraints.

The main contribution of the thesis lies in the definition of the system architecture, both hardware and firmware, customized to physical and operational specifications dictated by the Olympia Project context. The latter led to choices oriented towards energy efficiency, the most stringent constraint among those to be respected.

From a hardware point of view, the selection of the Seeed XIAO nRF52840 Sense Plus, based on Nordic architecture, coupled with a multi-axis IMU sensor, ensured mechanical compatibility, while at the same time providing a high-potential environment for achieving functional and non-functional requirements, optimally limiting costs per unit. From a firmware perspective, on the other hand, all possible scenarios for using the device have been considered, thanks to the development of a modular architecture based on functional blocks for each sensor and a state machine to manage energy. The frequency of acquisition and transmission of inertial data can be selected according to specific requirements, and the algorithms for activity classification and step detection are based on a set of computationally light features. System resource management, through the reduction of the microcontroller's duty cycle, ensures high functional autonomy for the device in realistic usage scenarios.

The measurements taken with a picoammeter and the empirical tests carried out the capability of the proposed system to operate for several months. This proves to be a technically sound and scalable solution in the context of biomechanical monitoring on sports prostheses.

7.2 Limitations of the Proposed System

Although the developed system meets the design requirements defined in Chapter 3, the architectural analysis shows some structural limitations that should be outlined.

- **Memory**: the device was developed on the assumption that local storage for long-term acquisitions was not possible and an external device for collecting data transmitted via BLE would be constantly available. The system is therefore dependent on connectivity and has extremely limited autonomy in offline scenarios.
- **Robustness**: the classification algorithm is based on signal variance thresholds along the anterior-posterior axis. This approach, while benefiting from computational lightness, can be sensitive to impulsive disturbances and inter-subject variability. The absence of a calibration phase makes the system operate in a deterministic but not adaptive manner.
- **Signal processing**: although the integrated IMU has an internal digital filtering system, the proposed system does not implement a dedicated filtering stage at the firmware level. Consequently, the spectral content of the signal is not explicitly controlled, and the separation between the quasi-static gravitational component and the dynamic acceleration associated with locomotion is only partial. This can lead to the propagation of high-frequency noise that affects the stability of the extracted features. A more structured filtering approach could improve the robustness of the system.
- **Energy optimization**: the logic implemented in the firmware allows for energy optimization of the system by reducing the MCU duty cycle and taking precautions with regard to the management of the on-board sensors. However, the system depends on MCU polling and continuous IMU operation at low frequency, and does not achieve completely event-driven behavior. The asymmetric nature of the application, characterized by limited daily usage periods of approximately two hours, can be more effectively exploited with appropriate design considerations.

7.3 Future Developments

So far, the work has focused on developing a system capable of ensuring continuous communication, so that the device can be queried at any time, in order to monitor every relevant aspect, with a priority on ease of use in a real-time context.

In this regard, analyzed scenarios share a mode of use that always provides for the possibility of transmitting BLE data, regardless of the selected rate. For this reason, until now, no design

constraints follow the limited device storage space, which had as its only prerequisite the ability to store firmware and small data packets of negligible size.

Assuming we want to store quaternions of signals (3 accelerations + 1 gyroscope axis) at a frequency of 12.5 Hz and considering 4 bytes per axis (float 32), we can deduce a flow of 200 B/s. This means that a single day's data would occupy 17 MB of data and exceed the microcontroller's internal flash memory.

It is possible to envisage some intermediate stages between a system operating entirely offline and the newly developed always-connected system, by equipping the nRF with local memory that is compatible with long-term storage. This would allow for autonomy in terms of data logging or complement BLE streaming.

One of the reasons for implementing a solution that is not completely dependent on continuous BLE transmission is the associated energy consumption. However, microSD cards have higher write currents than BLE acquisition or transmission. The impact that implementing an SD card has on battery life must be carefully evaluated.

A second critical aspect is related to time management. Writing to SD can introduce latency and generate jitter or sample loss if buffering is not sized correctly. In addition, writing occurs in asynchronous mode, which can compromise temporal determinism and therefore requires changes to the code structure that implements scenario 5.

A further critical issue concerns the risk of corruption of the microSD file system and therefore the possible compromise of the integrity of the recorded data. Controlled flush and file closure mechanisms must be implemented.

Finally, the integration of microSD increases the vulnerability of the system to vibrations and disconnections, which can complicate normal operation in biomechanical monitoring applications.

7.4 Future Firmware and Algorithmic Improvements

Quartiles

As described in the previous chapters, the classification algorithm in the embedded environment uses the variance of the AP signal, based on an estimate applied to a moving window, which reduces the computational complexity of the CWT, applied in an offline context.

However, the variance is potentially sensitive to impulsive outliers, such as accidental shocks or non-periodic micro-movements.

For this reason, a quartile-based logic is proposed, more specifically defining the interquartile range (IQR) as: IQR = Q3 - Q1.

Q1 and Q3 represent the first and third quartiles of the distribution in the sampled time window, respectively.

In this way, the system would be more robust with respect to the possibility of exceeding the classification threshold due to outliers and therefore more stable in uncontrolled operating conditions, increasing reliability in distinguishing between activities in real environments.

Digital filters

In order to obtain useful data in various real-world application contexts, it was deemed appropriate to select a filter library suitable for the developed system. In particular, the nRF52 integrates a Cortex-M4F core and can benefit from compatibility with the CMSIS-DSP library, developed for ARM Cortex-M cores.

This adoption would allow the implementation of FIR/IIR filters and therefore the separation of low-frequency components (gravity) from high-frequency components (noise), isolating contributions not relevant to the dynamics of the step that affect the features of interest.

This would result in better control over the useful band, reducing noise and strengthening the link between biomechanics and DSP.

Adaptive classifier

Although the warm-up based calibration provides an initial level of subject-specific adaptation, it is inherently based on a simplified and controlled representation of locomotor dynamics. In real-world conditions, variability across athletes, differences in prosthetic configurations, and context-dependent movement patterns introduce complexities that are not fully captured by such predefined segments. For this reason, future work should explore more flexible personalization strategies, moving toward adaptive and data-driven approaches capable of updating decision thresholds over time, based on the actual usage of the device rather than relying exclusively on an initial calibration phase.

Wake Up (via IMU, via internal RTC)

Scenario S5 shows how solutions can be adopted at the firmware level that reduce the MCU duty cycle, thereby limiting the system's energy consumption.

However, the asymmetric nature of the system's application and therefore the need for continuous use limited to 2 hours per day suggests that periodic MCU polling could be replaced by a hardware interrupt from the IMU sensor following motion detection. This event-driven solution would allow the microcontroller to go into deep sleep until the event occurs. In this scenario, the IMU could remain powered in low-power mode, resulting in constant consumption.

An even more energy-efficient solution, which could fully exploit a context in which the sports prosthesis is used at scheduled times, is a type of activation based on internal RTC, which assumes a programmed activation strategy. In this mode, both the MCU and IMU can be in complete power-down mode, and the system can be awakened by the internal RTC timer, which limits consumption to close to the theoretical minimum for extended periods of the day. Giving up continuous monitoring can extend battery life.

Considering an operating duty cycle of less than 10%, the theoretical autonomy can increase by an order of magnitude, which, in an active phase operating scenario at 12.5 Hz, would make the system compatible with the multi-year autonomy objective required by the prosthetics center (22 hours off out of 24 hours: duty cycle = 2/24 = 8% -> 1/0.08 = 12.5x).

Despite the advantages in terms of autonomy, it must be keep in mind that waking up involves the reinitialization of peripherals, which introduces a latency time at restart that must be managed correctly.

The choice between the two architectures depends on the continuous monitoring requirement: the event-driven solution maintains responsiveness at any time of day, while scheduled activation favors energy efficiency over functional time limitations.

7.5 Final Remarks

In conclusion, this work proposes a device that responds excellently to stringent design requirements. The choice of hardware provides a solid basis for development, with real potential for scalability. The firmware demonstrates the operation of each functional block involved in biomechanical monitoring and structures efficient management of system resources. The measurements taken with a picoammeter and the empirical tests relating to the device's autonomy serve as experimental validation of the electronic performance of the selected hardware components.

In the context of the Olympia Project, the reduction in the cost of the implemented device and the addition of new features, in particular the interrogability of the device, lay the foundations for new developments in academic and commercial research, opening the door to the spread of integrated microelectronics in prosthetic, sporting, and other contexts.

Bibliography

[1] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 21, 2012, Accessed: 25/03/2026. [Online]. Available: https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-9-21.

[2] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. Havinga, "Complex human activity recognition using smartphone and wrist-worn motion sensors," *Sensors*, vol. 16, no. 4, p. 426, 2016, Accessed: 25/03/2026. [Online]. Available: https://www.mdpi.com/1424-8220/16/4/426.

[3] M. Mellema and T. Gjøvaag, "Reported outcome measures in studies of real-world ambulation in people with a lower limb amputation: A scoping review," *Sensors*, vol. 22, no. 6, p. 2243, 2022, Accessed: 25/03/2026. DOI: 10.3390/s22062243. [Online]. Available: https://www.mdpi.com/1424-8220/22/6/2243.

[4] G. Bastas, J. J. Fleck, R. A. Peters, and K. E. Zelik, "IMU-based gait analysis in lower limb prosthesis users: Comparison of step demarcation algorithms," *Gait & Posture*, vol. 64, pp. 30–37, 2018, Accessed: 25/03/2026. DOI: 10.1016/j.gaitpost.2018.05.025. [Online]. Available: https://doi.org/10.1016/j.gaitpost.2018.05.025.

[5] D. Marcos Mazon, M. Groefsema, L. R. B. Schomaker, and R. Carloni, "IMU-based classification of locomotion modes, transitions, and gait phases with convolutional recurrent neural networks," *Sensors*, vol. 22, no. 22, p. 8871, 2022, Accessed: 25/03/2026. DOI: 10.3390/s22228871. [Online]. Available: https://www.mdpi.com/1424-8220/22/22/8871.

[6] N. Petrone, F. Giubilato, S. Low-Choy, M. Schrevens, and V. Quaggiotti, "Development of instrumented running prosthetic feet for the collection of track loads on elite athletes," *Sensors*, vol. 20, no. 20, p. 5758, 2020, Accessed: 25/03/2026. DOI: 10.3390/s20205758. [Online]. Available: https://www.mdpi.com/1424-8220/20/20/5758.

[7] F. Gariboldi, M. Scapinello, G. L. Migliore, A. G. Cutti, and N. Petrone, "Structural evaluation of lower-limb prosthetic sockets for running– part 1: Design, implementation and first assessment of an innovative test bench," *Results in Engineering*, vol. 26, p. 105 224, 2025, Accessed: 25/03/2026. DOI: 10.1016/j.rineng.2025.105224. [Online]. Available: https://doi.org/10.1016/j.rineng.2025.105224.

[8] Axivity Ltd, *Ax6 6-axis logging accelerometer & gyroscope - user manual*, Accessed: 25/03/2026, 2024. [Online]. Available: https://axivity.com/product/ax6.

[9] M. Tioli, I. Bernardoni, M. G. Santi, R. Di Marco, N. Petrone, and A. G. Cutti, "Activity detection of paralympic athletes with lower limb running-specific prosthesis during extended periods of time: Software development and preliminary validation," *Gait & Posture*, vol. 122, 2025, Accessed: 25/03/2026. DOI: 10.1016/j.gaitpost.2025.08.016.

[10] A. M. Sabatini, "Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing," *Sensors*, vol. 11, no. 2, pp. 1489–1525, 2011, Accessed: 25/03/2026. [Online]. Available: https://www.mdpi.com/1424-8220/11/2/1489.

[11] S. Zhang et al., "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors*, vol. 22, no. 4, p. 1476, 2022, Accessed: 25/03/2026. DOI: 10.3390/s22041476.

[12] V. R. M.Susi and G.Lachapelle, "Motion mode recognition and step detection algorithms for mobile phone users," *Sensors*, vol. 13, pp. 1539–1562, 2013, Accessed: 25/03/2026. DOI: 10.3390/s130201539. [Online]. Available: https://www.mdpi.com/1424-8220/13/2/1539.

[13] Microbots. "CodeCell: A tiny ESP32-C3 module for wearables and robotics." Accessed: 25/03/2026. [Online]. Available: https://microbots.io/products/codecell.

[14] Electronics-Lab. "CodeCell – a tiny ESP32-C3 module with Arduino compatibility." Accessed: 25/03/2026. [Online]. Available: https://www.electronics-lab.com/codecell-a-tiny-esp32-c3-module-with-arduino-compatibility-for-wearables-robotics-and-iot-projects/.

[15] Espressif Systems, *ESP32-C3 series datasheet v1.5*, Accessed: 25/03/2026, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf.

[16] L. Ada and B. Siepert, *Adafruit 9-DOF orientation IMU fusion breakout - BNO085: User guide*, Accessed: 25/03/2026, Adafruit Industries, 2024. [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-9-dof-orientation-imu-fusion-breakout-bno085.pdf.

[17] Arduino, *Arduino Nano 33 BLE sense (original sku: Abx00031)*, Accessed: 25/03/2026, 2026. [Online]. Available: https://store.arduino.cc/products/arduino-nano-33-ble-sense.

[18] Nordic Semiconductor, *nRF52840 product specification v1.9*, Accessed: 25/03/2026, 2023. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.9.pdf.

[19] ARM Ltd., *Arm® cortex®-M4 processor technical reference manual*, Accessed: 25/03/2026, 2020. [Online]. Available: https://developer.arm.com/documentation/100166/latest/.

[20] STMicroelectronics, *Lsm9ds1: iNEMO inertial module - 3d accelerometer, 3d gyroscope, 3d magnetometer*, Accessed: 25/03/2026, 2015. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm9ds1.pdf.

[21] Seeed Studio. "Seeed studio XIAO nRF52840 sense (nRF52840 & 6-axis IMU)." Accessed: 25/03/2026. [Online]. Available: https://www.seeedstudio.com/Seeed-Studio-XIAO-nRF52840-Sense-p-5253.html.

[22] STMicroelectronics, *Lsm6ds3tr-c iNEMO inertial module*, Accessed: 25/03/2026, 2017. [Online]. Available: https://www.st.com/en/mems-and-sensors/lsm6ds3tr-c.html.

[23] Bluetooth SIG, *Bluetooth core specification version 5.4*, Accessed: 25/03/2026, Jan. 2023. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification-5-4/.

[24] M. Zhang et al., "Integrated sensing and computing for wearable human activity recognition with MEMS IMU and BLE network," *Measurement Science Review*, vol. 22, no. 4, pp. 193–201, 2022, Accessed: 25/03/2026. DOI: 10.2478/msr-2022-0024.

[25] Unknown, "Energy-aware human activity recognition for wearable devices: A comprehensive review," *Pervasive and Mobile Computing*, 2024, Accessed: 25/03/2026. DOI: 10.1016/j.pmcj.2024.101976.

[26] E. S. Arch, J. M. Sions, J. Horne, and B. A. Bodt, "Step count accuracy of StepWatch and FitBit One among individuals with a unilateral transtibial amputation," *Prosthetics and Orthotics International*, vol. 42, no. 5, pp. 518–526, 2018, Accessed: 25/03/2026. DOI: 10.1177/0309364618767138. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0309364618767138.

[27] Agilent Technologies, *E3631a triple output dc power supply user's guide*, Accessed: 25/03/2026, 2013. [Online]. Available: https://www.jays.co.kr/12_Files/E3631A_DC%20Power_Supplies_UG.pdf.

[28] Keithley Instruments, *Model 6485 picoammeter instruction manual*, Accessed: 25/03/2026, 2023. [Online]. Available: https://pf18b.neocities.org/docu/Keithley_6485_Instruction_Manual_(part).pdf.

Ringraziamenti

Desidero ringraziare il Prof. Narduzzi per la disponibilità e l'attenzione con cui ha seguito e revisionato questo lavoro, accettando di affiancarmi in questo progetto in qualità di relatore. Un sincero ringraziamento al Prof. Morato per il supporto e la guida nello sviluppo della parte tecnica; al Prof. Petrone per avermi dato l'opportunità di entrare a far parte del progetto Olympia; e all'Ing. Cutti, referente INAIL, per il confronto e il contributo fornito nella fase iniziale del progetto.

A chi ha condiviso con me una corsa lungo il Piovego, una passeggiata in montagna o un'alba dal terzo piano.
A chi ha stonato insieme a me sulle note di una canzone Disney, dei FASK o di "Sembro F. Totti".
A chi mi ha accolto a casa sua, offrendomi una torta salata o un divano su cui dormire.
A chi, in silenzio, ha prenotato un tavolo al St. John, mi ha versato un bicchiere di cumino o ha filosofeggiato con me fino a tarda notte.
A chi mi ha ricordato cosa sia l'amicizia e a chi non ha mai smesso di dimostrarmelo.
A chi è stato per me come un fratello, sapendo ascoltare e consigliare. A chi, con sensibilità, è riuscito a essere allo stesso tempo leggerezza e punto di riferimento quando più ne ho avuto bisogno.
A tutti voi, vanno i miei sentiti ringraziamenti.

Un grazie particolare alla mia famiglia, ai miei nonni, ai miei zii e alla mia cuggi Vane.

Ale, grazie di essere il calore che ritrovo in un abbraccio e la sicurezza di un rifugio, anche quando sono lontano da casa. Grazie per custodire sempre uno spazio di confronto e condivisione, e per aver irradiato di gioia le mie giornate, rendendomi lo zio di Emma.

Cri, grazie di condividere con me lo sguardo che hai sul mondo, in cui riesco a ritrovare anche una parte di me. Sei la persona grazie alla quale non mi sento mai solo. Grazie per gli sguardi di intesa, per le battute dette all'unisono e per quelle che non hanno bisogno di essere dette (perché gls).

Mamma, grazie di essere la forte donna che sei. È grazie alla tua tenacia se ho potuto vivere questa esperienza con la leggerezza che cercavo. Grazie per avermi accompagnato fin qui, facendo sì che questo mio percorso diventasse una tua priorità, ancora prima che lo diventasse per me, e per avermi sostenuto con piccoli gesti costanti e fondamentali.

Grazie alla persona a cui devo tutto. A chi mi ha insegnato a non avere paura di fissarmi un obiettivo e, con il suo esempio, mi ha mostrato come raggiungerlo. A chi mi ha trasmesso cosa conta davvero, definendo la persona che sono oggi. A chi mi ha dimostrato quanto intensa e piena possa essere la vita, insegnandomi ad apprezzare ogni secondo che vive in un minuto.

A te, oggi, dedico questo traguardo, papà.