UNIVERSITY OF TURKU

Design and Implementation of Low Power Embedded System for OSA Monitoring

Digital Health, Department of Computing

Author(s):
KEJIA JIN

07.04.2026
Turku

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

Master of Science in Technology thesis

Subject: Master's Degree Programme in Information and Communication Technology

Author(s): Kejia Jin

Title: Design and implementation of a Low-Power Embedded system for OSA Monitoring

Supervisor(s): Tuukka Panula and Katri Karhinoja

Number of pages: 59

Date: 07.04.2026

Sleep disorders, and more specifically Obstructive Sleep Apnea (OSA), are becoming a major population health problem. Original overnight polysomnography is expensive, complicated and inapplicable to long term home monitoring. OSA is associated with hypertension, arrhythmia and cognitive decline. Therefore, there is a need for low-power wearable systems to reliably screen for OSA in the daily living environment as well as in a situation of eldercare.

This thesis presents the design and implementation of a low power embedded system for OSA monitoring. The system incorporates 2 key components: an NRF52840 system on chip (SoC) for data processing and wireless data transmission and an LSM6DSOTR inertial measurement unit (IMU) for measuring the respiration amplitude. Together, these modules comprise a small wearable platform to collect physiological signals and transmit it via Bluetooth Low Energy (BLE) to other platforms for further analysis.

The sleep is then converted into an Apnea-Hypopnea Index (AHI) which gets categorized into normal, mild, moderate or severe risk levels based on the number of events detected during sleep.

The system is able to detect OSA and generate AHI values that agree with the values generated by commercial sleep monitoring devices. A power characterization test revealed that the system can go all night and process consistent respiratory signals.

Keywords:

Obstructive Sleep Apnea (OSA); Wearable Health Monitoring; Inertial Measurement Unit (IMU); Embedded System; Low-Power Design; Respiratory Signal Detection

Table of contents

1 Introduction **5**
1.1 Problem Statement 5
1.2 Research Objective 8
1.3 Scope and Limitations 9
1.4 Research Report Structure 10

2 Background **12**
2.1 Sleep apnea physiology and screening methods 12
2.2 Review of Hardware Technologies for Wearable Sleep Monitoring 15
2.3 Future smart platform integration opportunities 18

3 Proposed System and Implementation **21**
3.1 Hardware Architecture and Hardware Module Breakdown 21
3.1.1 Respiratory signal Acquisition Module (LSM6DSOTR) 23
3.1.2 Main Control and Communication Module (NRF52480 + SDK) 24
3.1.3 BLE Transmission and Power consumption 26
3.1.4 System Block Diagram and PCB design 27
3.2 Firmware Design overview 31
3.2.1 Data Acquisition and Timing Control 32
3.2.2 Band-Pass Filtering and Sliding-Window Buffering 32
3.2.3 PCA-Based Fusion to a One-Dimensional Respiration Signal 34
3.2.4 Respiration Feature Extraction and Apnea Event Detection 35
3.3 System Integration 36
3.3.1 End-to-End System Workflow 36
3.3.2 Mobile Application Integration 37
3.3.3 Cloud-Based Elderly Care Platform 38
3.3.4 Application Scenarios in Elderly Care 38
3.3.5 System Output and Detection Logic 39
3.3.6 Challenges and Considerations 40

4 Experimental Results and Evaluation **41**
4.1 Experimental Objectives and Evaluation Strategy 41
4.1.1 Hardware and Firmware Configuration 41
4.1.2 Human Study Conditions and Scenarios 43
4.1.3 Test Dataset and Evaluation Framing 48
4.1.4 Progression from Raw Motion to a Breathing Trace 48

4.1.5 Time-Domain Respiratory Rate Behaviour 50
4.1.6 Frequency-Domain Validation 52
4.1.7 Motion Robustness and Quality Interpretation 53
4.1.8 Real-Time Operation and Power Consideration 54
4.1.9 Summary of Experimental Findings 55
4.2 System Performance 56
4.2.1 Power Consumption Test 56
4.2.2 BLE Transmission Speed and Reliability 57
4.2.3 Summary of System-Level Performance 57

5 Conclusion 58
5.1 Limitations 58
5.2 Future Work 59

References 60

Appendix A 64

1 Introduction

Obstructive Sleep Apnea (OSA) is one of the disorders that is characterized by sleep being often disrupted by breathing forfeiture that leads to a lack of oxygen and heart stress [1]. When OSA is not detected on time, it could result in higher blood pressure, heart rhythm disruptions, and cognitive impairment, and subsequently, the elderly would no longer enjoy the quality of life [2]. The conventional method of diagnosing the problem is overnight polysomnography (PSG) that is carried out in sleep laboratories within hospitals. The approach is both precise and highly expensive, time-consuming and labor intensive, and unsuitable for population level screening. With recent advances in sensor technology and embedded systems, it is possible to combine various electronic elements in the screening of sleep apnea into a single compact device, as shown in Fig 1.1. Inertial measurement unit (IMU) such as the STMicroelectronics LSM6DSOTR, can be used to detect respiratory patterns and changes in body posture.

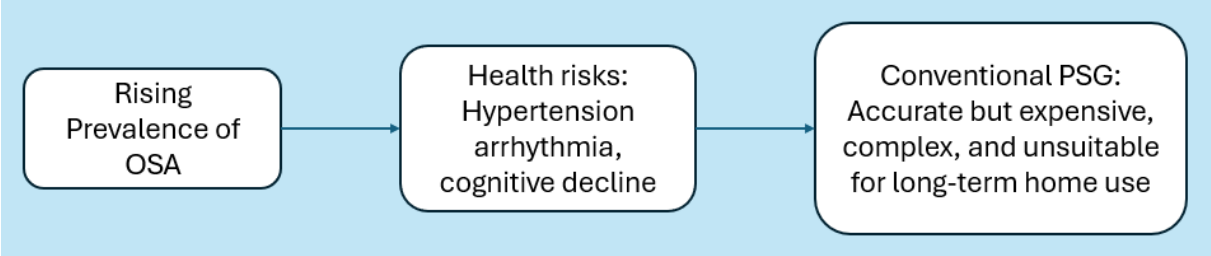

Fig 1.1: Research background of sleep apnea monitoring

Bluetooth Low Energy (BLE) microcontrollers offer continuous data acquisition and wireless transmission with minimal power consumption [2]. By combining these elements, we can create a system that is supported by very low power technologies, and that uses them to capture signals of respiration all night long, to detect events of disorganization such as apnea and accidental falls and will send a signal to a caregiver platform or to a family member. The goal is to develop a reliable privacy-preserving health-monitoring solution that runs autonomously for a few days to extend the clinical insight [3].

1.1 Problem Statement

Despite the years scientists have been studying OSA and the methods available for its clinical diagnosis, we have not yet found effective and accessible screening methods.

The current industrial standard, overnight PSG gives detailed information in multiple physiological channels but is limited by cost, complexity and heaviness [1]. PSG requires

specialized clinical facilities, well trained technicians and expensive equipment which are not appropriate for large-scale, nor for multiple home screening simultaneously. Patients also need to sleep while hooked up to many electrodes and sensors, which often disrupts natural sleep patterns.

Because PSG is usually performed for only one night, the variation of the severe level of OSA might not be well reflected by it, resulting in an underdiagnosis or late diagnosis in many cases. To address these shortcomings, scientists and businesses have invested in creating continuous and non-invasive monitoring devices like wearable and home-based monitoring devices, which are always-on to measure vital signs (e.g. heart rate, oxygen saturation, respiratory effort) [2]. However, most consumer wearables are still lacking clinical accuracy. Photoplethysmography (PPG) - based devices, commonly used in smartwatches and fitness trackers, are sensitive to artifacts in motion and skin contact, which differ in individuals. Actigraphy specifying apnea events can only be determined with certainty. More advanced home monitors with chest belts, nasal sensors or fingertip oximeters provide better coverage of physiological information at the cost of an uncomfortable fit for long term use, and in some cases draw too much current for an overnight home monitoring solution.

From a technical perspective, the main restriction is high power consumption when many sensors operate simultaneously, which is not very practical for overnight-operational battery-based systems. Many devices are based on a single method of sensing, meaning they lack the reliability of detecting apnea and proprietary algorithms rarely provide interpretation that relates to clinical. e.g. Apnea Hypopnea Index (AHI). Bulky sensors or cables limit compliance, among those groups that benefit most from long-term monitoring: elderly users. In the end, there is a clear gap between current clinical-grade diagnostic devices and the consumer wearable market.

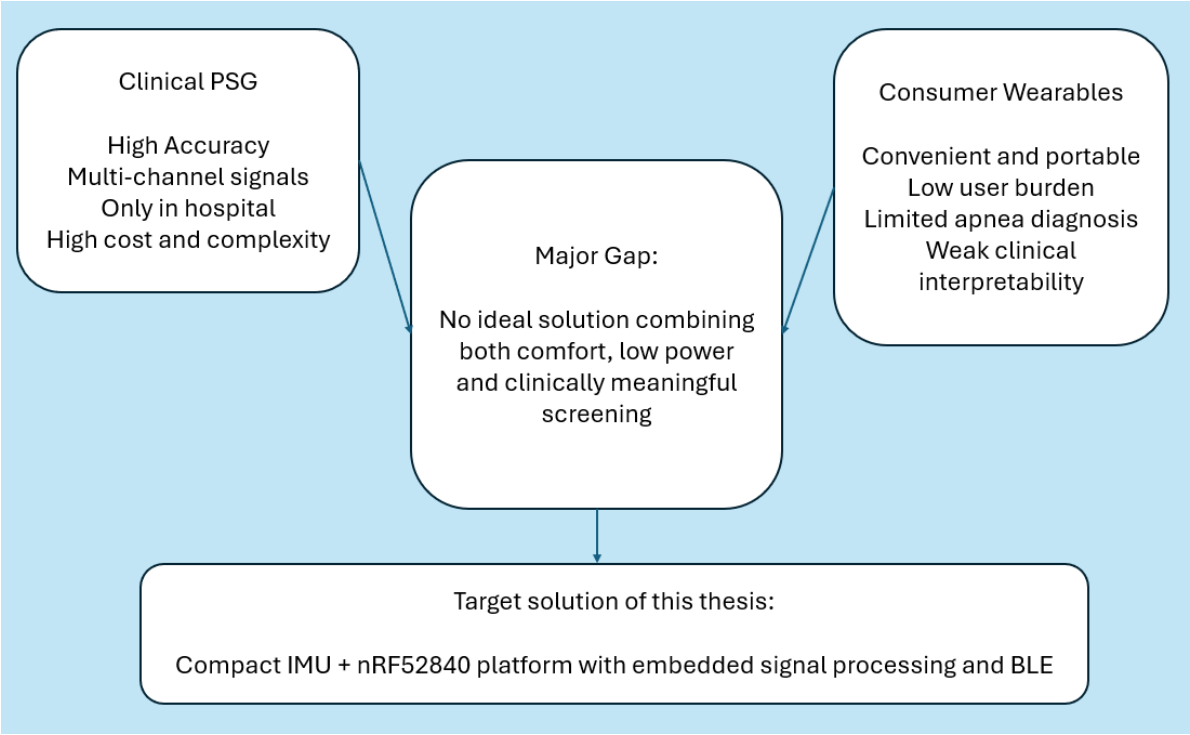

Fig 1.2: Technology gap with proposed solution

This thesis introduces a novel class of low-power wearable devices that integrate clinically level signal processing algorithms into a compact and easy-to-operate device in order to calculate respiratory disorders, as shown in Fig 1.2.

Specifically, the system is a combination of an IMU, the LSM6DSOTR assembled by STMicroelectronics, to identify the respiration. The LSM6DSOTR is a low-power six-axis inertial sensor, that combines three-axis accelerometers and three-axis gyroscopes in a small package. The device has programmable sampling rates and ranges making it able to detect movements of small body parts accurately. In this project, an accelerometer sensor is used to record minor movements of the chest during respiration, which can be used with embedded processing algorithms to approximate breathing patterns and identify respiratory abnormalities. A Nordic Semiconductor nRF52840 microprocessor with efficient processing and BLE communication [3].

The thesis is based on two research questions:

1. Whether the design of a power-efficient embedded system to acquire high-quality respiratory signals for overnight use is possible or not, this is indeed challenging task to scale the quality of signals up to the desired level and at the same time meet the very low power demands of battery-operated wearable devices.

2. Build a lightweight, interpretable algorithm that approximates clinical apnea standards. This involves developing processing algorithms that are computationally efficient enough to run on resource-limited microcontrollers while still providing reliable outputs.

Success in overcoming these challenges is necessary for building a digital health monitoring device with desirable outcomes. The research device will work on formulating a usable and scalable treatment for sleep-apnea screening and aim to integrate into the home health monitoring solution for elders in the future.

1.2 Research Objective

The main objective of this thesis is to design and implement a low-power embedded system for OSA monitoring, which can be integrated later into other smart digital healthcare platforms.

The research aims to bridge the gap between diagnostic systems at the clinical grade solutions used by professionals and smart wearable solutions used by consumers through developing a solution that is capable of high-quality biomedical signal acquisition and efficient signal processing. To accomplish this goal, the work is managed with clear objectives. The first objective is to build a wearable system based on dedicated biomedical and embedded hardware. Potentially the solution combines two major parts: the LSM6DSOTR IMUs to detect thoracic or abdominal movement as a key parameter for respiration measurement, and the nRF52840 microcontroller as the main processing and communication unit that can carry out signal preprocessing and BLE transmission [1]. This hardware design targets a tradeoff between signal fidelity, power consumption and wearability and can track users through the night.

The second objective is to establish an effective signal processing and event generating algorithm. The embedded system carries out a number of functions. They are signal filtering and preprocessing, respiration of IMU signals using band-pass filtering to remove high-frequency noise and separate low-frequency breathing motion, and detection of apnea events by detection of periods during which the amplitude of respiration decreases below a predefined threshold for at least 10 seconds. Besides, the system has the potential to integrate heart rate variability analysis to further verify and determine the AHI to determine the severity of sleep apnea as normal, mild, moderate, and severe based on the current clinical guidelines [3]. Computational resources should be optimized to fit into the other hardware.

The third objective is to evaluate the performance of the system in a real-life situation. The analysis contains several system performance factors. First, the quality of the signal is determined through the determination of the respiration stability involving IMU-based

respiration measurements. Second, the accuracy of the apnea detection is also tested to determine how well the system can reliably detect apnea-like events and compare the results with the AHI estimates. The average power consumption is also checked in terms of power efficiency to ensure the possibility of overnight operation. Moreover, the upcoming work will involve the assessment of the stability and integrity of BLE transmission between the device and a mobile application.

The fourth objective is to determine the possibilities of the system in the future as a solution to elderly care. In addition to tracking the presence of OSA, the device would be tested in terms of its ability to support other tasks such as fall detection, abnormal respiratory rate (RR) detection, and health reporting via mobile platforms. To find out the possible integration with other sensors and provide a contribution to the digital healthcare ecosystems, as shown in Fig 1.3.

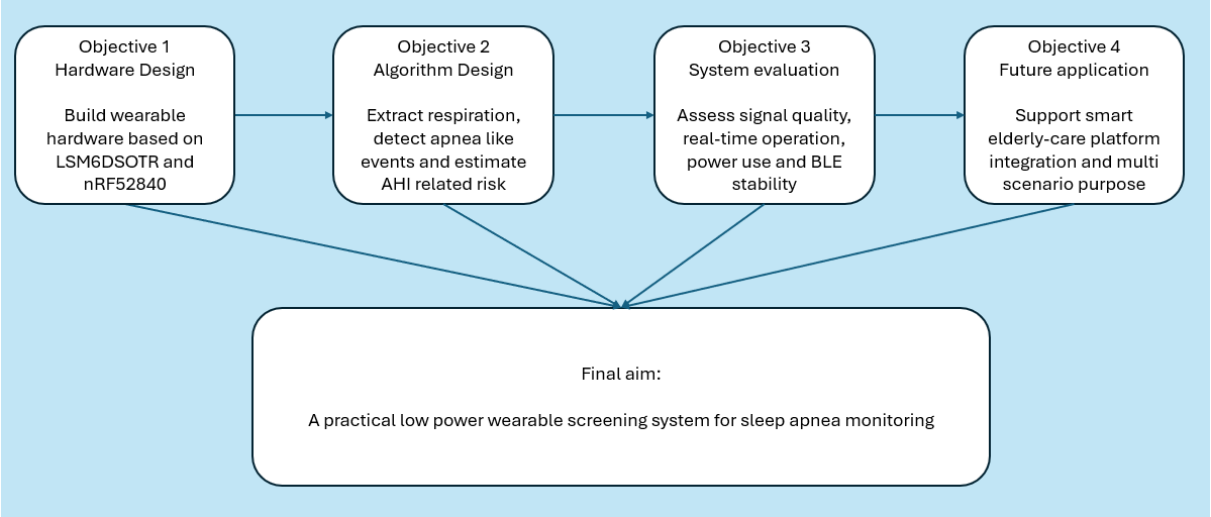

Fig 1.3: Research objectives of the thesis

1.3 Scope and Limitations

The thesis scope includes the implementation and validation of a low-power embedded system to detect sleep-apnea and may be incorporated in other intelligent healthcare systems. The research notes that there might be a necessity to strike some compromise between engineering feasibility and clinical relevance to demonstrate that small and low-power consumption devices can generate clinical level insights [2]. The thesis study addresses a broad array of typical industrial practices, but there are several specific technical limitations, nevertheless.

The proposed scope of this thesis is the design and validation of an embedded system with low power consumption to detect sleep apnea and potentially integrate it into intelligent healthcare systems. The study aims to show that small wearable systems can deliver useful respiratory data with limited power and hardware requirements.

The proposed system will be constructed on the two core hardware components: the LSM6DSOTR IMU to sense respiration and the nRF52840 microcontroller to process the data, and support BLE communication. These involve the design of the electronic design that will connect these parts and the programming of the firmware that will preprocess the signal, extract the respiration waveform, and detect an apnea event.

Three-axis inertial measurements recorded at the chest are used to monitor respiration. These signals are processed in real time by the system using lightweight embedded algorithms, which include cascaded second-order section (SOS) filtering to remove noise, principal component analysis (PCA) to fuse signals, and adaptive hysteresis zero-crossing to estimate the respiratory rate.

Even though the system illustrates the practicality of IMU-based respiration monitoring, there are several factors that are beyond the scope of the work. Specifically, modern machine learning methods to predict apnea are not considered because of the processing power of the embedded platform. In addition, long-term system validation, large-scale user studies, and regulatory considerations such as medical certification are not addressed in this study.

1.4 Research Report Structure and Use of AI

Chapter 1 provides the context of the entire research. It starts with the introduction of the existing social issues with the OSA especially the increasing impact of sleep-related diseases over the years. The scope and constraints of the existing diagnostic methods and potential clinical solutions are mentioned.

Chapter 2 gives a review of the background and relevant work being done on it. It offers an overview of the medical aspects of OSA, and the clinical indicators used to diagnose OSA for patients. The chapter then provides an overview of the hardware technologies used in wearable monitoring devices with inertial sensors, and low-power microcontrollers. Finally, it discusses some existing applications of smart healthcare systems, to put this research in the larger context.

Chapter 3 describes the design and implementation of the proposed system. It describes the hardware architecture, circuit schematic including the integration of LSM6DSOTR and

NRF52840 on the printed circuit board (PCB). The chapter also describes the firmware algorithm with special emphasis on signal acquisition, preprocessing, event detection and risk assessment. The future potential integration of the system with the healthcare platform is also introduced in a nutshell.

Chapter 4 contains the account of the experimental evaluation of the system. This includes the design of test scenarios, the analysis of the quality of respiration signals, and the analysis of the apnea detection. The chapter also discusses system aspects such as circuit schematics to ease power consumption, how to stabilize wireless communication, and a few future elder care application user interface mockups.

Chapter 5 is the concluding chapter for the thesis. It also gives a summary of the main contributions, the limitations of the present work and mentions the potential future research directions which include clinical verification, multi-sensor integration and algorithmic enhancement.

This structure provides a logical guide on the way to learn more about the development of the suggested system, starting with motivation and ending with validation and further use.

Regarding the use of artificial intelligence (AI), Gemini AI tool was used in a limited supporting role for language polishing, grammar correction, and improving clarity of expression, whereas the core ideas, methodology, experiments, analysis, and main written content were developed independently.

2 Background

This chapter provides background information from both a physiological and technical perspective of the proposed system. At the beginning of this chapter, OSA physiology and existing available screening methods are reviewed to provide a clinical context. We will also discuss the latest relevant sensor technologies, e.g., IMUs, which are made up of accelerometers and gyroscopes that detect and measure motion and orientation, respectively; thus, the IMUs can also be used to find the respiratory movements of the chest or identify changes in body position. In the end, we will explore the criteria for hardware platform and component selection, followed by a discussion of smart healthcare applications where such monitoring systems could be applied.

2.1 Sleep apnea physiology and screening methods

OSA is one of the most common sleep disorders, characterized by repeated episodes of partial or complete upper-airway obstruction during sleep. These events cause a decrease or a lack of airflow despite continued attempts at respiration and will cause oxygen destruction and sleep fragmentation [1]. As a result, individuals suffer from excessive daytime sleepiness, impaired concentration and have a higher risk of long term cardiovascular and metabolic problems. OSA is known to be a major public health concern, affecting an estimated 400 million people worldwide, with its presence significantly increasing in the elderly as well as in those with obesity or hypertension [3].

Diagnosis and severity assessment of OSA is based on a primary metric called the AHI, which is the average number of apnea and hypopnea events in one hour of sleep [4]. Apnea refers to discontinued airflow for at least ten seconds while hypopnea is defined as a 30% or more reduction of airflow associated with oxygen destruction or cortical arousal. The AHI helps doctors identify the severity of OSA and provide appropriate treatment to patients.

According to the clinical diagnosis, 5 or fewer apnea events per hour is considered normal; 5-15 is mild OSA, 15-30 is moderate OSA, and more than 30 events per hour is severe OSA [5]. In practice, there are also other symptoms that doctors consider (e.g., fatigue, hypertension, or arrhythmia); however, AHI remains the primary metric for the diagnosis, shown in Fig 2.1.

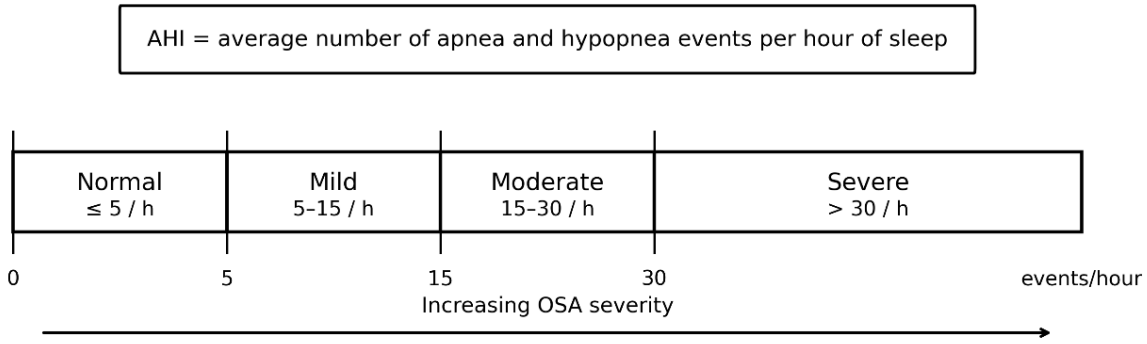

Fig 2.1: AHI-based severity classification of OSA

PSG is still considered the standard in diagnosing AHI [6]. It is an overnight sleep study in some specialized clinical labs, and many physiologic signals are collected concurrently, including the Electroencephalogram (EEG), Electrooculogram (EOG), Electromyogram (EMG), Electrocardiogram (ECG), nasal airflow, thorax and abdomen movements, and blood oxygen saturation [7]. This multi- sensor technology provides complete data on sleep stages, breathing patterns and cardiopulmonary responses that guarantee proper detection of apnea and hypopnea episodes. Although it is highly accurate in diagnosis, PSG has several drawbacks that make it inapplicable in all clinical cases. The PSG test is costly, lengthy and demands complicated clinical facilities. Patients are likely to find the dozens of electrodes and sensors quite uncomfortable, and they are certainly being disrupted in quality sleep, making the quality of the obtained data lower [8]. Secondly, since PSG tests are usually conducted in the context of a single night, they may not be exposed to the variability of OSA severity over multiple nights, which would cause underdiagnosis in the population with low healthcare access [9].

To overcome these limitations, Home OSA Testing (HSAT) systems have been developed, that record fewer physiological signals and can be analyzed without supervision by a health care professional [10]. Typically, cardiopulmonary test devices monitor airflow, respiratory effort and oxygen saturation to estimate AHI. Thus, while HSAT makes the system more accessible and less expensive, due to the lack of EEG products, HSAT cannot ascertain sleep stages with the same accuracy, so the resulting AHI estimation will be less accurate, especially in borderline conditions [11].

Wearable sleep monitoring devices have granted accessibility with the embedded sensors into products such as wristbands, chest straps or smart rings [12]. These devices are based on physiological surrogates such as actigraphy and PPG and can analyze the quality of sleep and respiration conditions. Although they are comfortable to carry compared to those clinical

facilities, most consumer wearables cannot process signals with reliable apnea results, and this makes it very difficult to produce accurate AHI scores.

From recent research, we have also identified promising solutions between diagnostic platforms and consumer wearable devices, by integrating specialized biomedical sensors and intelligent signal processing algorithms [13]. Recently, it has been investigated how IMUs can be used to measure respiration in wearable health systems. One of the methods is to use accelerator-based sensors to detect minute thoracic motions that occur during breathing to obtain resonance patterns based on motion signals. Several studies have revealed that IMU sensors on the chest or abdomen can yield credible readings of respiratory activity in both home-monitoring and clinical settings [14].

Embedded physiological signals processing. Low-power embedded platforms have also been used extensively to compute physiological signals in wearable monitoring systems. Energy efficient microcontrollers allow continuous data capture and signal processing and high battery life, which is important in overnight monitoring systems. They are frequently combined with wireless communication devices like BLE to relay physiological information to mobile devices or cloud services to analyze it further [15].

Whereas wearable sensing systems cannot substitute the clinical gold standard of diagnosing sleep apnea, PSG offers a promising answer to high-scale screening and long-term monitoring without the need for clinical procedures. This can be useful, especially in home-based monitoring and digital health, where ease of use, low cost and continuous data collection are important, as shown in Fig 2.2.

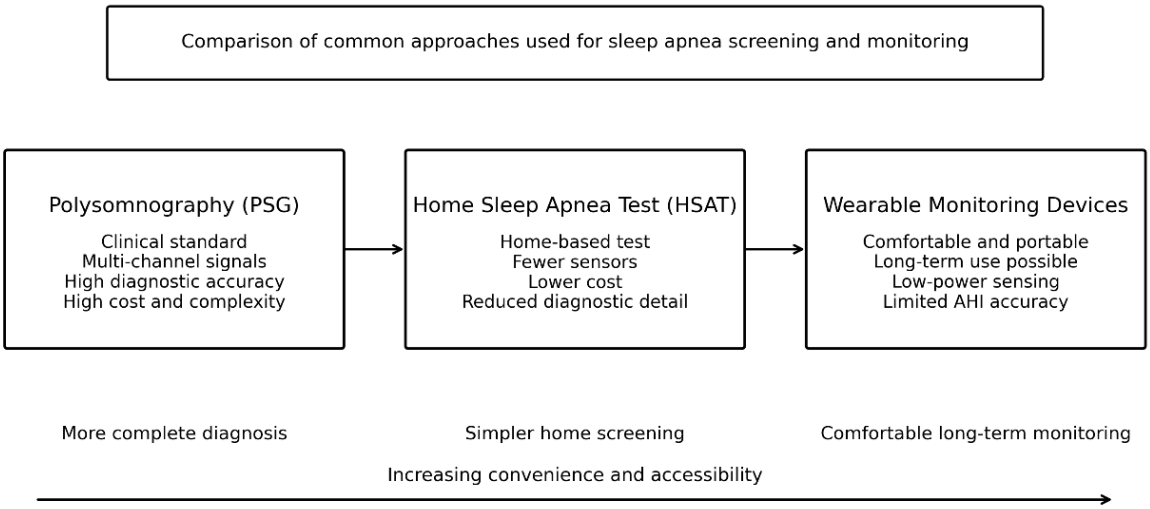

Fig 2.2: Common OSA screening and monitoring methods.

2.2 Review of Hardware Technologies for Wearable Sleep Monitoring

Wearable devices are largely dependent on sensor technologies to ensure signal quality and usability, with accuracy of signal acquisition and power consumption remaining the most important factors in selecting the sensor technologies. In this section, the review of representative devices in two categories is made to detect OSA: microcontroller platforms, and IMUs. There have been numerous new solutions over the past few years to improve the detection, prevention, and long-term monitoring of sleep disorders. It reflects a current trend toward the implementation of digital technologies in healthcare, which opens new opportunities to further enhance accessibility, accuracy, and patient comfort when implemented in everyday monitoring uses.

Since sensing accuracy power allocation are key requirements for signal quality and usability, selecting sensor components for wearable devices remains very important. From a system hardware perspective, there are two key components, which are the contributors for OSA detection, and they are the micro-controller unit, and IMU.

At the center of every wearable system, there is the microcontroller unit (MCU), which controls the operation of various sensors, microscopes, data processing units and wireless communication technologies on board. The choice of MCU has a direct impact on computing power consumption and integration compatibility with other components.

The nRF52840 is a very popular Low-power SoC that was optimized for wearable health monitoring applications. It has an ARM Cortex-M4 processor with a floating-point hardware accelerator operating at 64 MHz with enough computing resources for embedded biomedical algorithms. With 1MB of flash memory and 256 KB of RAM, it has good allocation space for the firmware and signal buffers [16].

One of the key features of the nRF52840 is the capability of supporting various wireless protocols, including BLE 5, Thread, Zigbee and ANT. BLE 5 also has a longer range and higher throughput than previous versions, which is crucial for reliable communication between healthcare devices at night. The microcontroller unit is additionally integrated with a DC/DC converter into a stand-alone chip with a supply voltage range between 1.7 V and 5.5 V. Integrating protocol stacks along with the nRF5 SDK, this enables us to develop firmware to effectively monitor respiratory apnea activities [17], as shown in Table 2.1.

Table 2.2.1 Comparison of Microcontroller platforms.

Feature	nRF52840 (Chosen)	STM32L4 Series	ESP32	nRF52832
CPU Architecture	ARM Cortex-M4F	ARM Cortex-M4	Xtensa LX6	ARM Cortex-M4
Max Clock Frequency	64 MHz	80 MHz	240 MHz	64 MHz
Flash / RAM	1 MB / 256 KB	up to 1 MB / 128 KB	4 MB ext. / 520 KB	512 KB / 64 KB
Wireless Support	BLE 5, Thread, Zigbee, ANT	External RF required	Wi-Fi + BLE	BLE 4.2
Power Consumption (Sleep)	<1 µA	~1–2 µA	>10 µA	~1 µA
Power Consumption (Active)	Low (optimized for BLE)	Low	High	Low
BLE Throughput & Range	High (BLE 5, long range)	N/A	Moderate	Limited
SDK & Ecosystem	Mature (nRF5 SDK)	Mature	Moderate	Mature
Suitability for Wearables	Excellent	Good	Poor	Limited

The main reason why nRF52840 was selected was that it has the following key features: lower power consumption, integrated BLE5, and sufficient computational capability for real-time respiratory signal processing. If we compare it with ESP32, for example, which offers significantly higher power consumption, and it is a very important feature to have lower power consumption for overnight monitoring. STM32 would be an excellent alternative but it requires an external module specifically for wireless connectivity. nRF52840 has a built-in BLE module and it has enough memory on the chip to support extensive firmware expansions.

Respiratory monitoring is one of the most important aspects of the OSA diagnosis. Although from a clinical diagnosis perspective, PSG with airflow sensors and respiratory belts can provide more accurate results, they cannot be used in regular practice due to comfort and compliance concerns. IMUs are a promising alternative solution because they can detect minor chest movements or the abdomen with respiration cycles. There are many alternative IMUs for this project to use, as shown in Table 2.2[18].

Table 2.2.2 Comparision of IMUs.

Feature	**LSM6DSOTR (Chosen)**	**MPU6050**	**BMI160**	**ADXL345**
Sensor Type	6-axis (Accel + Gyro)	6-axis	6-axis	3-axis Accel
Accelerometer Resolution	16-bit	16-bit	16-bit	13-bit
Noise Density (Accel)	Low	Moderate	Low	Moderate
Power Consumption	Ultra-low	Moderate	Low	Very low
Embedded Functions	Activity, step, low-power modes	None	Limited	None
Sampling Flexibility	High	Moderate	High	Limited
Size & Integration	Compact	Larger	Compact	Compact
Motion Artifact Handling	Good	Moderate	Good	Poor
Suitability for Respiration	Excellent	Acceptable	Good	Limited

We selected LSM6DSOTR sensor as it can provide us with high-resolution acceleration, ultra low power operation and stable performance at low frequencies, all these aspects are essential for detecting minor chest movements associated with respiratory activitiese during sleep. ADXL345 would be a great alternative solution but lacking robustness against posture changes. The LSM6DSOTR sensor offers very low power consumption, making it suitable for battery-powered wearable devices. Its compact size and integrated six-axis sensing capability allow it to be easily embedded into small wearable systems, improving user comfort during long-term monitoring.

2.3 Future smart platform integration opportunities

During the recent couple of decades, numerous smart health monitoring wearable solutions have been developed, with the help of more mature wireless communication technology and cloud technology, these solutions can perform health monitoring continuously for days and sometimes even weeks. With the continuous digital healthcare technology evolution, remote monitoring as a diagnosis solution plays a vital role in enhancing the quality of life for elderly populations.

Approximately 15% of the world's population is aged 65 or older and this number will easily reach 22% by 2050. Healthcare for the older generation presents one of the biggest healthcare challenges of the 21st century. Elderly people are the majority group to have chronic diseases such as cardiovascular disease, diabetes, brain impairment and sleep disorders. All of these require constant monitoring and long-term medical treatment [19]. Traditional healthcare systems that are built around hospital-based care and periodic visits make it very difficult for many elderly people who are living alone at a remote distance and especially for those who lack mobility. This has led to the development of smart healthcare solutions with wearable sensors, wireless communication and cloud-based data analytic platforms. This would allow individuals to monitor their self-health data in real-time and acquire remote medical support when needed. The remote health monitoring solution is designed to capture physiological signals outside of traditional healthcare centers and send regular diagnosis reports to healthcare professionals when needed. Remote healthcare solutions have two main purposes: to decrease the need for supervision from hospitals and decreasing elder people who lives alone the need for supervision from hospital care at the same time as ensuring safety due to constant monitoring of vital signs and activities. Most monitoring metrics are heart rate, blood pressure, oxygen saturation, respiration, temperature and body movement. Detecting early signs of deterioration and initiating treatment on time is made possible by using remote monitoring solutions.

Sleep apnea monitoring is especially relevant to elderly care as the quality of sleep directly affects physical and cognitive health, and OSA is very common among elderly people [20].

Advancements in sensor technologies and wireless communication technologies have made it feasible to integrate wearable solutions with other healthcare systems. Devices are usually developed to be comfortable for long term continuous monitoring. Common examples are wristbands, chest patches or systems made of textile with embedded sensors [21]. For sleep-apnea monitoring purposes, a wearable device with an IMU sensor can continuously capture the respiratory activity without the complication of involving other medical equipment.

Through BLE connectivity, physiological information for example, can be sent to a smartphone and then to a cloud platform for comprehensive diagnosis analysis [22]. With the help of such a wearable solution, it enables caregivers and family members to access the health data in real time and increase awareness and early engagement when needed [23].

A typical smart healthcare solution architecture includes two layers:

1. Sensing Layer - Sensors that can be worn on the body to gather physiological and behavioral data. Examples of these types of sensors include ECG sensors, accelerometers (detecting falls), and radar-based motion sensors.
2. Application Layer – Data is aggregated, analyzed and visualized using mobile applications or dashboards on a cloud platform, and digital alerts are being sent out when abnormal events are detected.

Although smart healthcare systems have a very promising future, there are still several challenges. Maintaining the signal's reliability is challenging because the elderly users might lose contact with the system periodically or remove it while sleeping. Energy efficiency is also crucial since devices need to run overnight or for several days without being recharged [24]. Ensuring data privacy and data security is also important, as sensitive health related information must be protected from being used for wrong purposes and they are normally very personal and confidential.

Some smart wearable solutions for elderly care include applications for detecting falls with accelerators or radar-based devices, these devices automatically notify caregivers if a person has fallen and is in a dangerous situation. Remote cardiac monitors can read the arrhythmia and send the data from the ECG for professional review [25]. Sleep-monitoring solutions are added to the elderly care platforms to recognize apnea and overall sleeping quality.

Environmental sensors like motion sensors, temperature probes and appliance-usage sensors add to the context of behavioral and health assessment. Machine learning and artificial intelligence methods are being examined to help analyze multimodal data and to predict health risks, as well as to personalize the delivery of care [26]. Wearable OSA monitoring is one of the most important solutions for the healthcare ecosystem.

In this section, Table 2.3 summarizes representative monitoring functions and sensing schemes that are relevant to elderly-care scenarios.

Table 2.3 Comparison of monitoring functions and sensing schemes

Application Area	Typical Sensing Method	Relevance to Elderly Care
Fall detection	Accelerometer, gyroscope, mm Wave radar	Prevents complications from unattended falls
Cardiac monitoring	ECG front-end, PPG sensor	Detects arrhythmias, heart rate variability
Respiration & sleep	IMU, airflow sensors, ECG, SpO_2	Identifies OSA, tracks breathing patterns
Activity monitoring	Step counters, motion sensors	Assesses daily mobility and independence
Environmental safety	Smart home sensors (motion, appliances)	Provides context, ensures safe living conditions

Smart healthcare and remote health monitoring applications address the critical need for continuous, accessible healthcare in aging societies. Combining wearable sensors, wireless communications, and cloud-based analyses, these systems not only bring medical understanding into the homes of those outside of medical facilities but also identify conditions like OSA, and track falls and cardiovascular values, showing the possibility of wearable embedded systems to support older adults in a more comprehensive way.

3 Proposed System and Implementation

This chapter presents the design and implementation of the proposed respiration monitoring system. It describes the overall system architecture, including the hardware components and embedded firmware used for respiration signal acquisition and processing. The chapter also explains the signal processing algorithms implemented on the embedded platform for respiration extraction and apnea event detection. Finally, the integration of the hardware and firmware components is discussed to demonstrate the feasibility of the proposed low-power wearable monitoring system.

3.1 Hardware Architecture and Hardware Module Breakdown

The proposed device is organized as a chest-worn embedded node in which motion sensing, computation, power management, and communication are integrated on a compact board. The implementation path used in this thesis prioritizes local acquisition and real-time transfer (RTT) based validation, while keeping BLE available as a future deployment extension. This separation is deliberate: development measurements are taken from the most transparent data path, whereas wireless telemetry remains reserved for later mobile or cloud integration, as shown in Fig 3.1.

These two components are combined to create an efficient lightweight and low-power system with wireless transmission of data to mobile devices using BLE.

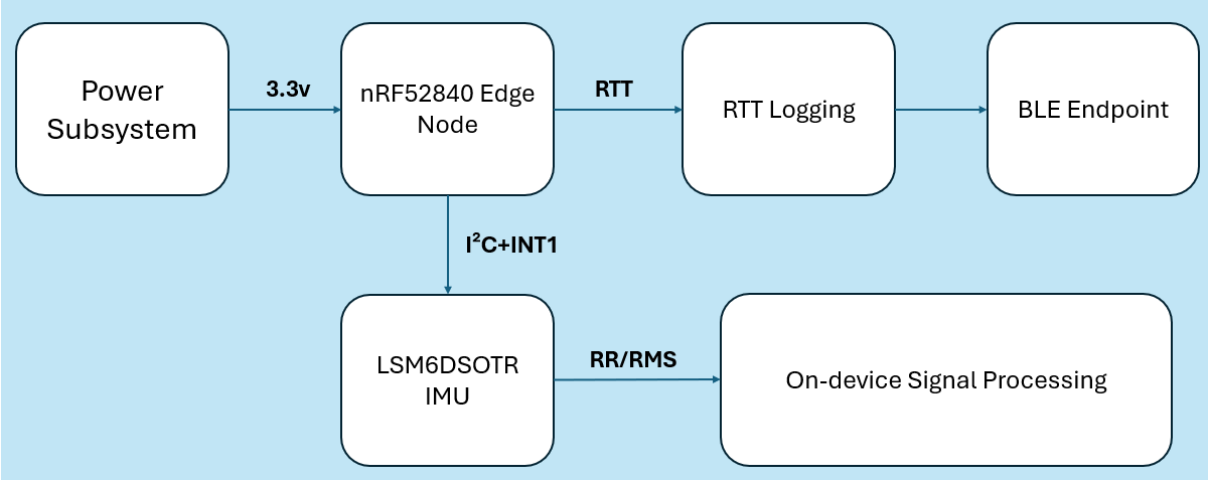

Fig 3.1: System architecture of the proposed respiration-monitoring device

When placed on the chest, low frequency oscillations associated with breathing cycles are captured by the accelerometer. The raw acceleration data is transmitted to the microcontroller

for processing, where respiratory information is extracted through signal processing techniques described in Section 3.2.

Regarding the hardware module shown in Table 3.1, in operational use, the IMU is fixed on the thoracic region so that breathing-induced expansion and contraction appear as low-frequency micro-accelerations. The nRF52840 acquires these samples at a stable rate, updates analysis buffers, and computes the current respiratory features. During the reported proof-of-concept experiments, processed values and status labels are exported over RTT to allow offline verification against the generated plots. This validation-first workflow reduces ambiguity when interpreting firmware behavior.

Table 3.1: Hardware module breakdown

Module	Implementation	Role in the system
Main control and BLE	nRF52840	Schedules acquisition, stores short analysis windows, executes filtering and feature extraction, and supports future BLE telemetry.
Respiratory motion sensing	LSM6DSOTR	Chest-mounted triaxial accelerometer used to capture low-frequency thoracic motion associated with breathing.
Power subsystem	LiPo input, regulator, passives	Provides battery-powered operation with 3.3 V regulation and decoupling suitable for wearable prototyping.
Physical realization	Custom PCB and validation setup	Integrates sensor, controller, connectors, programming/debug support, and practical mechanical routing into a manufacturable prototype.

External interface	Host-side plots	Used as the validation path during development for waveform logging, respiratory-rate review, root mean square (RMS) tracking, and status interpretation.

3.1.1 Respiratory signal Acquisition Module (LSM6DSOTR)

The direct method of obtaining airflow or thoracic effort through clinical polysomnography techniques like belts or nasal cannulas is not appropriate for wearable devices. An alternative method is to use inertial sensors which estimate the respiratory activity in an effective manner, as they can detect tiny motions of the chest and abdomen during inhalation and exhalation [27]. Hence the LSM6DSOTR six-axis inertial measurement unit was chosen for building this prototype, as shown in Fig 3.2.

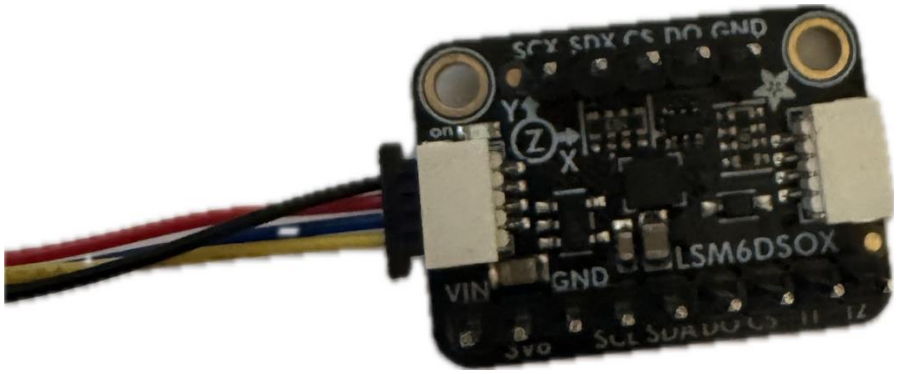

Fig 3.2: LSM6DSOTR IMU sensor.

In this study, three-axis inertial data are used to monitor respiratory motion. The sensor is positioned on the chest with its Z-axis oriented perpendicular to the body surface, allowing thoracic movements during breathing cycles to be captured. The accelerometer has been set with a range of ±2 g and the sampling rate of 25Hz, which is more than enough to capture normal respiratory frequencies (0.1-0.7Hz or 6-42 breaths per minute) [28]. The unprocessed acceleration data is sent to the nRF52840 through I²C for signal processing.

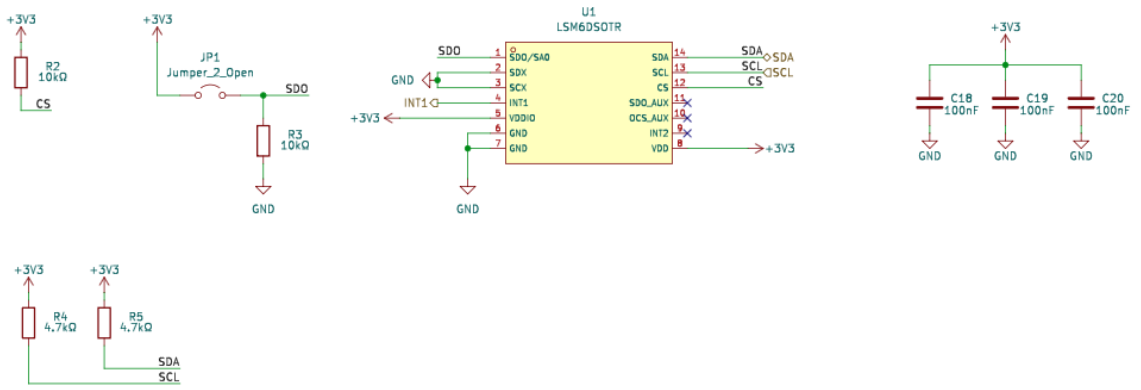

Fig 3.3: Circuit schematic view of LSM6DSOTR IMU sensor

U1 from Fig 3.3 as LSM6DSOTR is the IMU motion sensor which is equipped with an accelerometer and gyroscope. It connects to the main chip using two data and clock lines which are SDA + SCL, this is the common two wire communication method. Where R4 and R5 are assisting in making those two data lines work properly. INT1 is an “attention” line from the sensor to the main chip especially when the new data comes in. C18, C19, and C20 are small power smoothing capacitors for the sensor and there is also a jumper JP1 and resistors R2 and R3 which help define how the sensor is configured and are mainly used during setup.

3.1.2 Main Control and Communication Module (NRF52480 + SDK)

Centralized control and communication are provided by the nRF52840 SoC of Nordic Semiconductor, as shown in Fig 3.4 [29]. This enables a microcontroller that combines a 64 MHz ARM Cortex-M4 processor with floating-point support, which has enough computational power for on-board pre-processing and event detection. It contains 1 MB of flash memory and 256 KB of RAM, which is enough for buffering data and firmware.

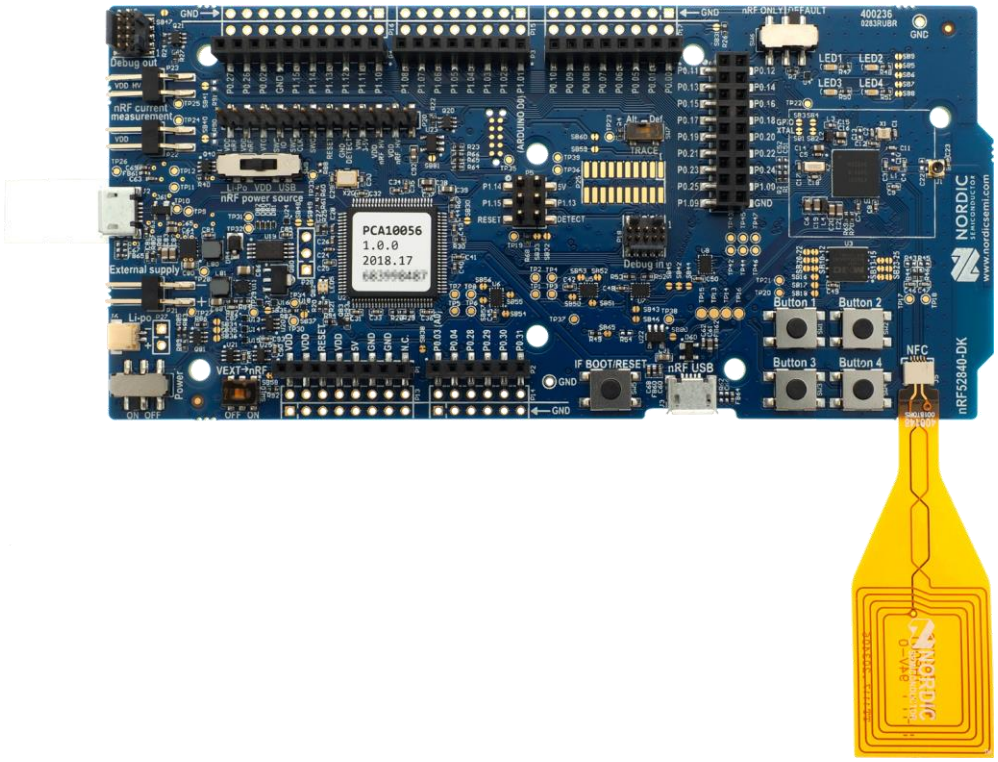

Fig 3.4: nRF52480 microcontroller

The nRF52840 as mentioned earlier is the one which not only provides the required processing power but also has the wireless communication capabilities for the application. In this scenario, the nRF52840 takes the role of the master device to orchestrate the data gathering from the LSM6DSOTR, do the preprocessing and event recognition, and send the results through BLE.

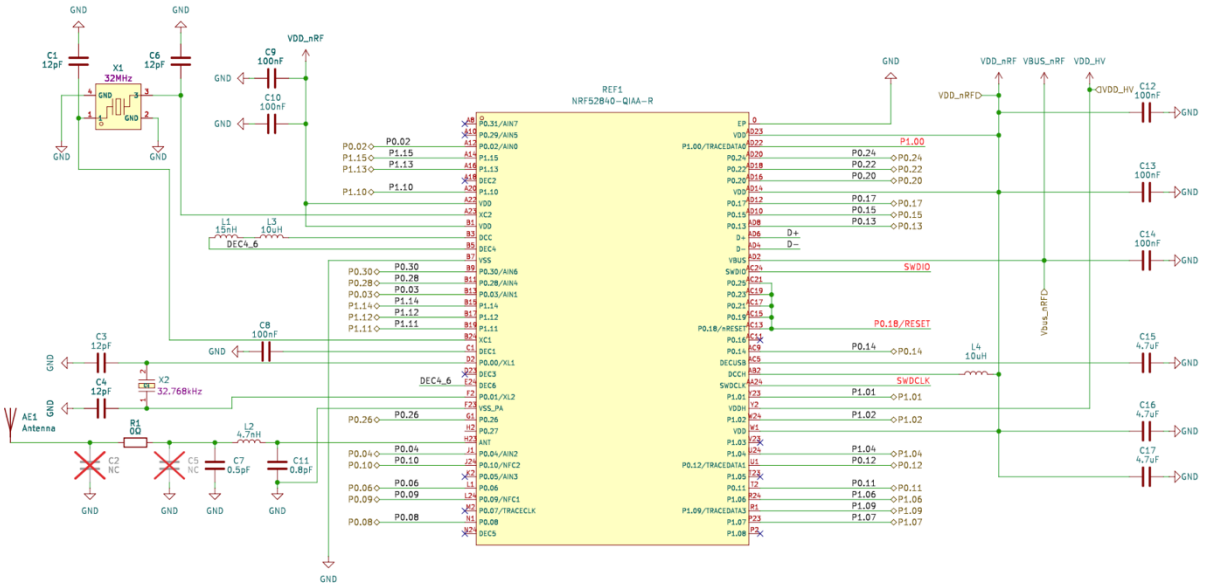

Fig 3.5: Circuit schematic view of nRF52480 microcontroller

The Fig 3.5 nRF52840 microcontroller serves as the main processing unit of the system. It executes the embedded firmware, reads data from the IMU sensor via the I²C interface, and manages BLE communication. The Bluetooth antenna is called AE1 and those small parts near the antenna like L2, L1, C7, C11 etc are radio tuning network for radio communication, X1 with 32 MHz crystal is the main clock for accurate timing this would help the chip run precisely, X2 with 32.768 KHz crystal is a low power clock which is often used for keeping time/low power sleep timing, there are also many 100nF + 4.7uF capacitors like (C9, C10, C12 etc), they are the main power stabilizers placed around the chip so it will stay reliable, D1 LED + R8 is the status light when the power is on or facing errors, RST1 is the reset button and Debug_JN1 is the debug connector use d to load firmware and system debugging.

3.1.3 BLE Transmission and Power consumption

The nRF52840's integrated BLE 5 radio enables wireless data transmission to mobile devices. The BLE communication is configured to transmit processed data packets at 1-second intervals during active monitoring.

The nRF52840 supports multiple wireless BLE protocols such as Bluetooth 5.2 Low Energy (BLE 5), Thread, Zigbee and ANT. BLE 5 comes with long range, high throughput, and increased reliability [33]. From an energy standpoint, the device allows for a supply range of 1.7 - 5.5 V with a high efficiency integrated DC/DC converter. The device supports ultra-low power sleep modes, which can be used for overnight running on batteries. The nRF5 SDK released by Nordic Semiconductor contains pre-compiled protocol stacks and libraries optimized for BLE, streamlining firmware for making the communication between devices stable, as shown in Fig 3.6.

Within this system, the nRF52840 synchronizes data from LSM6DSOTR, performs processing and event detection and sends the results through BLE. The processor's processing power, power consumption and wireless connectivity make it a perfect core for this wearable system.

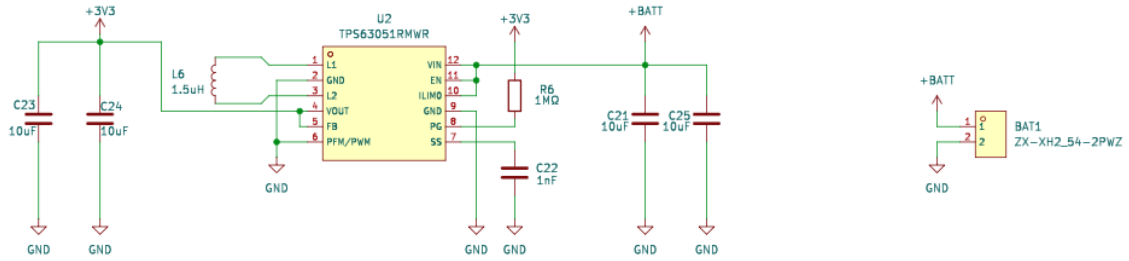

Fig 3.6: Circuit schematic view of nRF52480 battery with 3.3v stable power

The desired system is running with steady 3.3V that the electronics can safely being used, BAT1 is the battery connector where the battery plugs in and U2 (TPS63051) is the power converter chip which works as following, if the battery voltage is higher than needed, it steps it down and if the battery voltage is lower than needed, it boosts it up, its mission is to keep the output at 3.3v, L6 is the coil used by the converter and C21, C23, C24, C25 are capacitors that smooth the power so device won't glitch when the current demand varies.

3.1.4 System Block Diagram and PCB design

However, to emphasize the integration of the different modules, the block diagram of the global hardware is illustrated below. The LSM6DSOTR inertial measurement unit is connected to the nRF52840 microcontroller by an I2C communication bus, as shown in Fig 3.7. The SDA and SCL lines are made of general-purpose I/O pins on the MCU allowing low power and bidirectional data transfer for continuous monitoring of respiratory signals. An interrupt line is used to signal data-ready events, allowing the microcontroller to remain in low-power sleep modes between measurements. Power comes from a regulated 3.3V rail with jumper and test points added which allow for flexible power configuration during development. This hardware configuration allows the good acquisition of the low frequency chest motion signals with low energy consumption which is essential for wearable overnight operation.

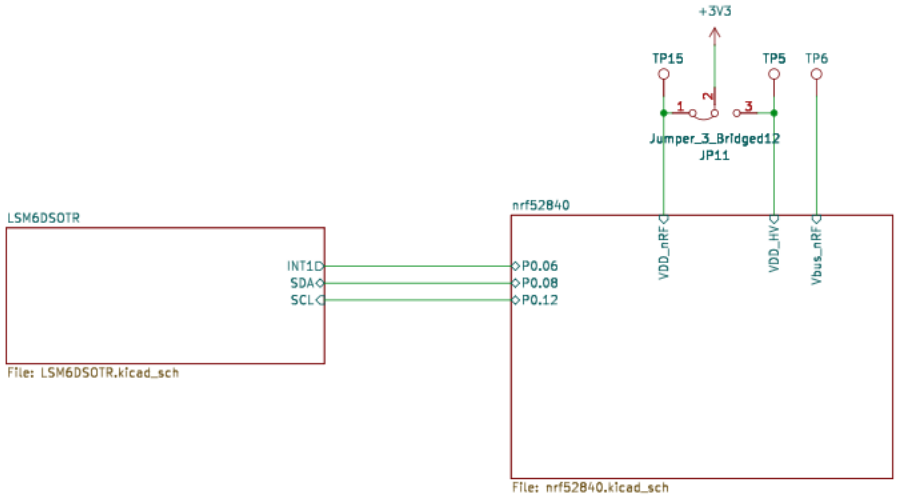

Fig 3.7: Circuit schematic view of electrical connection between the LSM6DSOTR IMU and the nRF52840 microcontroller.

Depending on the system configuration, the PCB of this system has also been designed by us, especially for this research. The PCB contains the primary elements of processing, power control, and wireless communication elements needed to support low-power OSA monitoring in a small size.

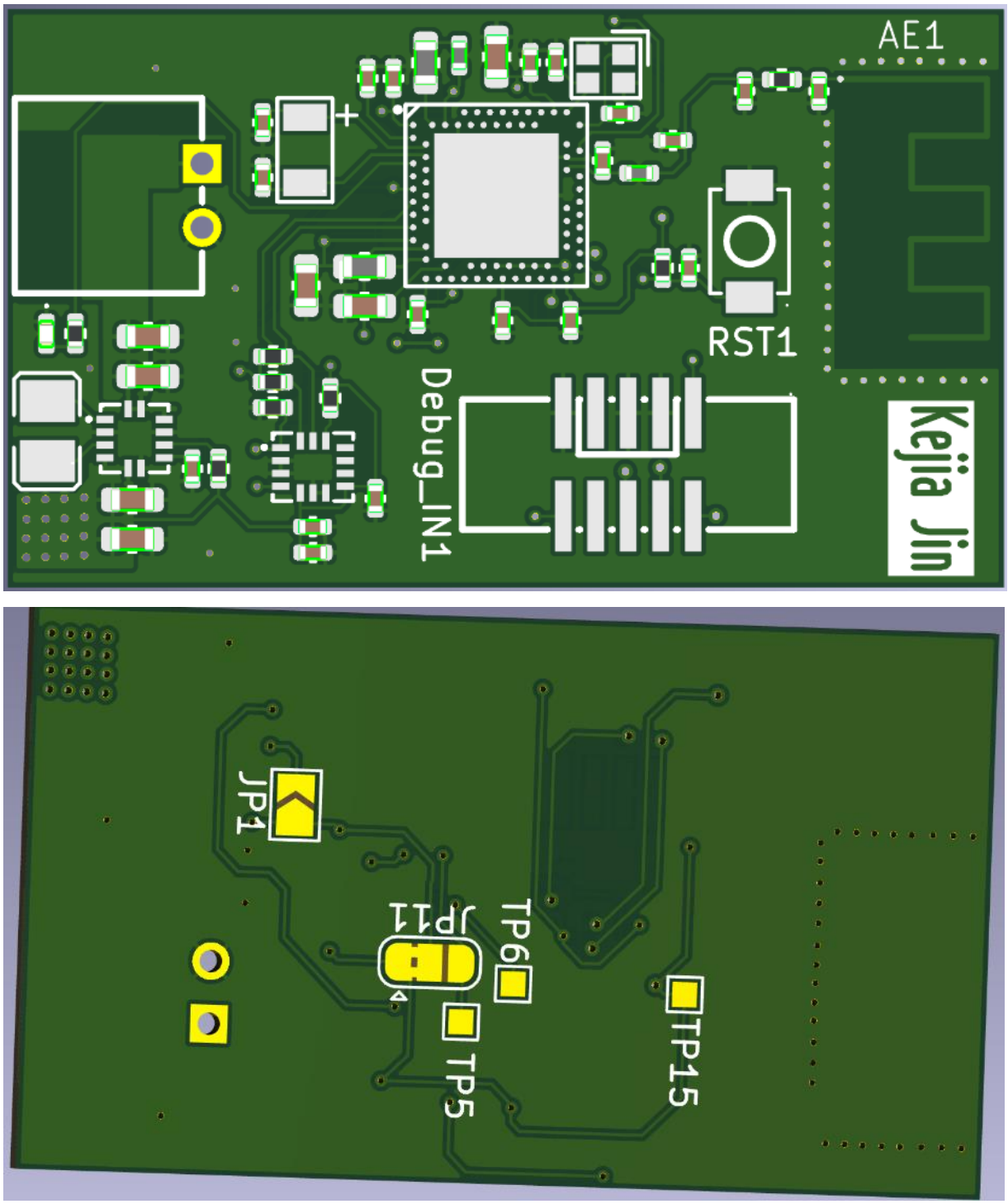

Fig 3.8: Front and back sides of the sleep sense-IMU prototype board.

The nRF52840 system-on-chip is in the middle of the board and it becomes the main processing and communication unit, as shown in Fig 3.8. It contains a BLE radio that allows transmission of data wirelessly without utilizing external RF modules.

The peripheral components are positioned around the microcontroller in order to minimize the length of signal paths and minimize electromagnetic interference. The LSM6DSOTR inertial

unit is placed outside the RF sections to enhance the signal stability and ability to record the low-frequency chest motion related to breathing. Capacitors are installed near the power pins of all active components to provide stability to the voltage in changing dynamic loads.
The board has a power management circuit that assists in operating batteries. It has a rechargeable lithium-polymer battery interface and local voltage regulation to drive a stable 3.3 V rail to the MCU and sensors. They are provided with test points and debug components to enable current measurement, firmware flashing and lack of signal inspection during development and validation.
The PCB design is focused on the low-power and wearable-friendly design concepts. The real dimensions of the PCB are 38.5 mm and 22.5 mm in width, as shown in Fig 3.9.

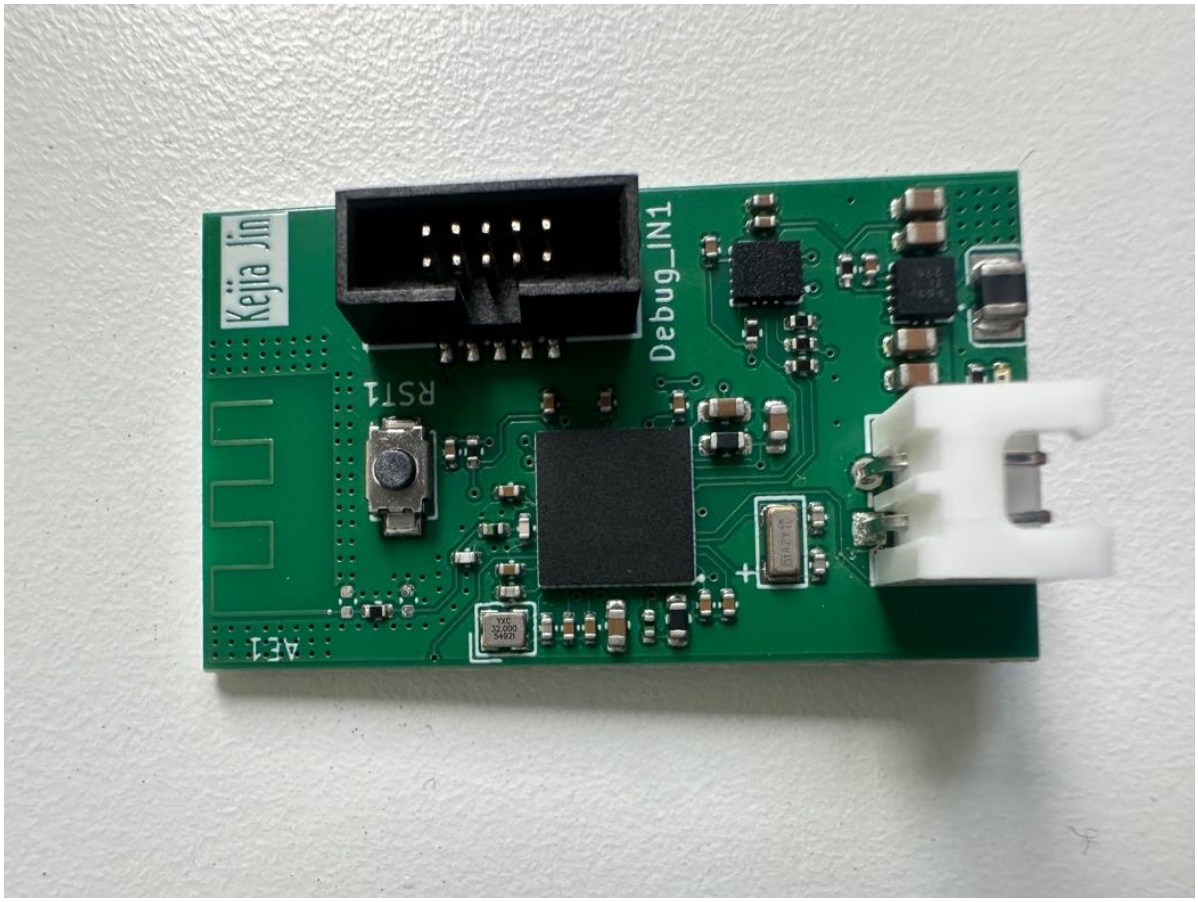

Fig 3.9: Actual PCB for the project work

3.2 Firmware Design overview

The performance of the proposed wearable monitoring system does not only depend on the hardware, but also on the software pipeline through which the raw data of the devices (sensors) is translated into clinically relevant information, as shown in Fig 3.10.

The software performing on the wearable tool is in charge of obtaining inertial sensor data; isolating respiration relative movement; and identifying irregular breathing indicators in real time. The implementation is developed on the Zephyr RTOS platform and is optimized for low-power operation and continuous monitoring. The algorithm's main purpose is to transform raw 3-axis accelerometer data into respiration status through 5 stages: fixed-rate sampling, then with band-pass filtering, adding sliding-window buffering, through PCA fusion to a one-dimensional respiratory waveform, and finally with rule-based feature extraction.

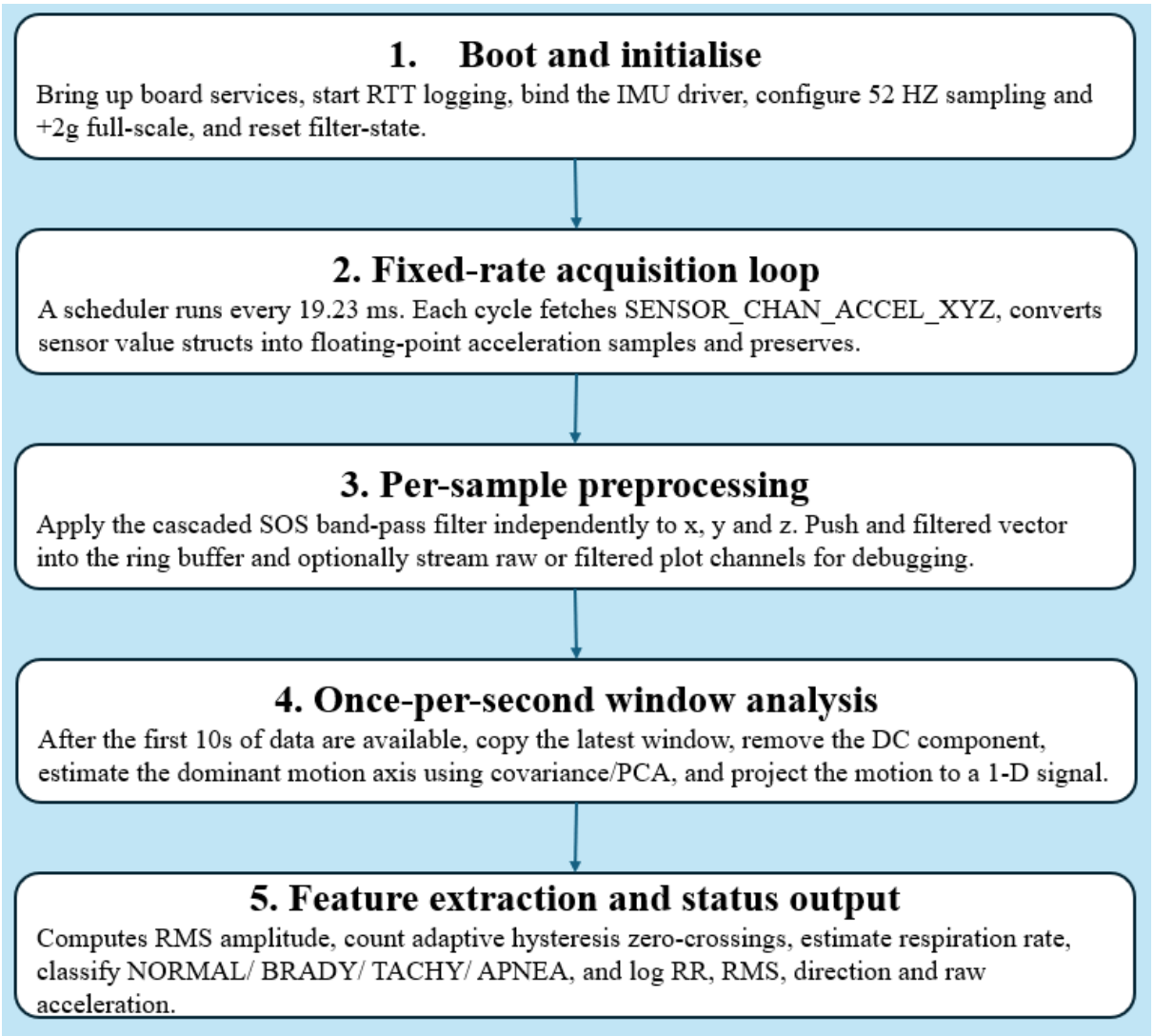

Fig 3.10: Firmware runtime flow used for sample acquisition, processing and output generation.

3.2.1 Data Acquisition and Timing Control

The LSM6DSOTR accelerometer is configured to sample at 52 Hz with a ±2 g measurement range. Configuration is performed using the Zephyr sensor API.

```
#define SAMPLE_RATE_HZ 52.0f
struct sensor_value odr = { .val1 = 52, .val2 = 0 };
sensor_attr_set(dev, SENSOR_CHAN_ACCEL_XYZ,
SENSOR_ATTR_SAMPLING_FREQUENCY, &odr);
struct sensor_value fs = { .val1 = 2, .val2 = 0 }; sensor_attr_set(dev,
SENSOR_CHAN_ACCEL_XYZ, SENSOR_ATTR_FULL_SCALE, &fs);
```

Sampling is timed using a fixed period timing loop that is based on system uptime. The firmware computes the next acquisition deadline and sleeps for the rest of the time. If an overrun occurs the timing reference is resynchronized to avoid drift.

```
int64_t next_us = k_uptime_get() * 1000;
const int64_t period_us = (int64_t)(1000000.0f / SAMPLE_RATE_HZ);
while (1) {
next_us += period_us;
   int64_t now_us = k_uptime_get() * 1000;
   int64_t sleep_us = next_us - now_us;
   if (sleep_us > 0) {
      k_sleep(K_USEC((int32_t)sleep_us));
   } else {
      next_us = now_us;
   }
}
```

Fixed-rate sampling loop and overrun handling.

Inside the loop the collected are obtained from the IMU and transformed to a floating point value which are suitable for the signal processing. To avoid double-precision conversions, the firmware makes use of a lightweight conversion routine.

```
static inline float sensor_val_to_f(const struct sensor_value *v)
{ return (float)v->val1 + (float)v->val2 / 1000000.0f;}
```

Conversion from Zephyr sensor format to floating-point values.

3.2.2 Band-Pass Filtering and Sliding-Window Buffering

To extract respiratory motion from the raw accelerometer measurements, a digital band-pass filter is applied to the three-axis inertial signals. Human respiration typically occurs at low frequencies ranging from approximately 0.1 Hz to 0.7 Hz**,** corresponding **to** 6–42 breaths **per** minute. Frequencies outside this range mainly correspond to motion artifacts, sensor drift, or high-frequency noise. Therefore, a band-pass filter is used to preserve the respiratory components while attenuating irrelevant signal components.

In this system, a 4th-order Butterworth band-pass filter is implemented using cascaded second-order sections (SOS) for improved numerical stability on the embedded platform. The filter operates with a sampling frequency of 52 Hz**,** which is sufficient to capture the low-frequency respiration dynamics.

The respiratory signal was filtered using a fourth-order Butterworth band-pass filter designed at a sampling frequency of 52 Hz. The passband was specified as 0.1-0.8 Hz to retain the dominant respiration-related frequency components. To suppress unwanted low-frequency drift, frequencies below 0.05 Hz were defined as the low-frequency stopband. In addition, frequencies above 1.0 Hz were attenuated as the high-frequency stopband to reduce noise and non-respiratory interference. For improved numerical stability and suitability for embedded implementation, the filter was implemented as cascaded second-order sections (SOS).

The passband preserves the frequency components corresponding to normal respiratory activity, while the stopbands suppress very low-frequency baseline drift caused by posture changes and high-frequency disturbances generated by body motion or environmental vibration.

After filtering, the signals are stored in a sliding**-**window buffer for subsequent processing. The system maintains a window length of 10 seconds, corresponding to approximately 520 samples at 52 Hz sampling rate. This window size provides sufficient temporal information to estimate respiration rate reliably while maintaining low computational latency.

The buffer is implemented as a ring buffer, allowing continuous insertion of new filtered samples while discarding the oldest samples once the window is full. The respiration analysis algorithm is executed at **1-**second intervals, enabling near real-time monitoring while limiting computational overhead on the embedded microcontroller.

This filtering and buffering approach ensures that the respiratory signal is sufficiently clean and stable before further processing steps, such as Principal Component Analysis (PCA) fusion and respiration rate estimation, are applied.

```
f.x = sos_filter_process(&fx, ax);
f.y = sos_filter_process(&fy, ay);
f.z = sos_filter_process(&fz, az);
```

Per-axis SOS band-pass filtering of acceleration samples.

The filtering core uses a Direct Form II Transposed SOS cascade.

```
float sos_filter_process(sos_filter_t *f, float x)
{
   for (uint8_t i = 0; i < f->n_sec; i++) {
      sos_section_t *s = &f->sec[i];
      float y = (s->b0 * x) + s->z1;
      s->z1 = (s->b1 * x) - (s->a1 * y) + s->z2;
      s->z2 = (s->b2 * x) - (s->a2 * y);
      x = y;
   }
   return x;
}
```

Filtered samples are stored in a ring buffer which corresponds to a 10 second window. Respiration analysis is triggered 1 second after the buffer is full. The windowing and triggering logic is illustrated below.

```
ring_push(f);
samples_since_eval++;
if (ring_filled == WINDOW_SIZE && samples_since_eval >= STEP_SAMPLES) {
   samples_since_eval = 0;
   ring_to_window(win_xyz);
}
```

3.2.3 PCA-Based Fusion to a One-Dimensional Respiration Signal

Because the direction of wearable devices might not be the same for all people and the sleeping posture may change, respiration motion might not be always aligned with one accelerometer axis. The firmware therefore uses PCA based fusion to get an orientation independent respiration waveform. the three-axis window data are centered by subtracting the mean. This removes residual offsets before covariance estimation.

The principal motion direction is estimated from the center. This setup would computes the 3×3 covariance matrix and applies a lightweight power iteration method to obtain the dominant eigenvector.

The three-axis data then project onto one direction to product the final respiration waveform.

Below is the PCA workflow.

1. center_vec3f(win_xyz, WINDOW_SIZE, &mean);
2. pca_principal_dir(win_xyz, WINDOW_SIZE, &dir);
3. project_onto(win_xyz, WINDOW_SIZE, &dir, fused);

This fused signal is then used for respiration feature extraction and event classification.

3.2.4 Respiration Feature Extraction and Apnea Event Detection

The respiration analyzer calculates both the amplitude of breathing as well as the respiration rate out of the fused signal. Breathing amplitude is estimated by RMS(Root Mean Square) for the window. RMS is stable to outliers and is an indication of the overall energy of the respiration waveform.

```
float sum_sq = 0.0f;
for (int i = 0; i < len; i++) {
    float s = signal[i];
    sum_sq += s*s;
}
r.amplitude_rms = sqrtf(sum_sq / (float)len);
```

RMS amplitude estimation of the respiration signal.

Apnea-like events are detected based on continous low respiration amplitude. If RMS remains below a defined threshold for at least 10 seconds, the system flags an apnea status.

```
const float apnea_rms_th = 0.0196f;
if (r.amplitude_rms < apnea_rms_th) {
    a->low_energy_seconds += step_sec;
} else {
    a->low_energy_seconds = 0.0f;
}
if (a->low_energy_seconds >= 10.0f) {
    r.status = RESP_STATUS_APNEA;
    r.rate_bpm = 0.0f;
    return r;
}
```

Duration based apnea detection logic using low energy thresholding.

Respiration rate is estimated using adaptive hysteresis zero-crossing, where the hysteresis band scales with the peak-to-peak signal range to reduce spurious crossings from noise. The core crossing count logic is shown below.

```
const float apnea_rms_th = 0.0196f;
if (r.amplitude_rms < apnea_rms_th) {
    a->low_energy_seconds += step_sec;
} else {
    a->low_energy_seconds = 0.0f;
}
if (a->low_energy_seconds >= 10.0f) {
    r.status = RESP_STATUS_APNEA;
    r.rate_bpm = 0.0f;
    return r;
}
```

Respiration rate estimation using adaptive hysteresis zero-crossing.

The crossing count is converted to breaths per minute and used for breathing pattern classification.

When the signal is not catagorized as apnea, breathing status is classified based on respiration rate thresholds. Rates above 25 breaths per minute are labeled as tachypnea, rates below 10 breaths per minute as bradypnea, and values in between as normal respiration[33].

```
if (r.rate_bpm > 25.f) {
   r.status = RESP_STATUS_TACHYPNEA;
} else if (r.rate_bpm < 10.f) {
   r.status = RESP_STATUS_BRADYPNEA;
} else {
   r.status = RESP_STATUS_NORMAL;
}
```

Rule-based respiration classification.

In the end, the full analysis is executed and returned to the main loop using a single function call, make it easy for testing.

```
resp_result_t res = resp_analyze(&analyzer, fused,
                                 WINDOW_SIZE,
                                 SAMPLE_RATE_HZ,
                                 STEP_SEC);
```

Respiration analysis module.

The firmware algorithm is rule based and avoids intensive computational methods, ensuring predictable runtime with low memory usage and explainable behaviour. This make it suitable for battery-powered wearables designed for overnight operations.

3.3 System Integration

This section describes the connection between the hardware layer and the software layer, how data is transported from sensors through the cloud services, and how the system communicates with other potential applications for the future potential development work.

3.3.1 End-to-End System Workflow

The LSM6DSOTR from Fig 3.11 is sampling low-frequency respiratory chest movements.

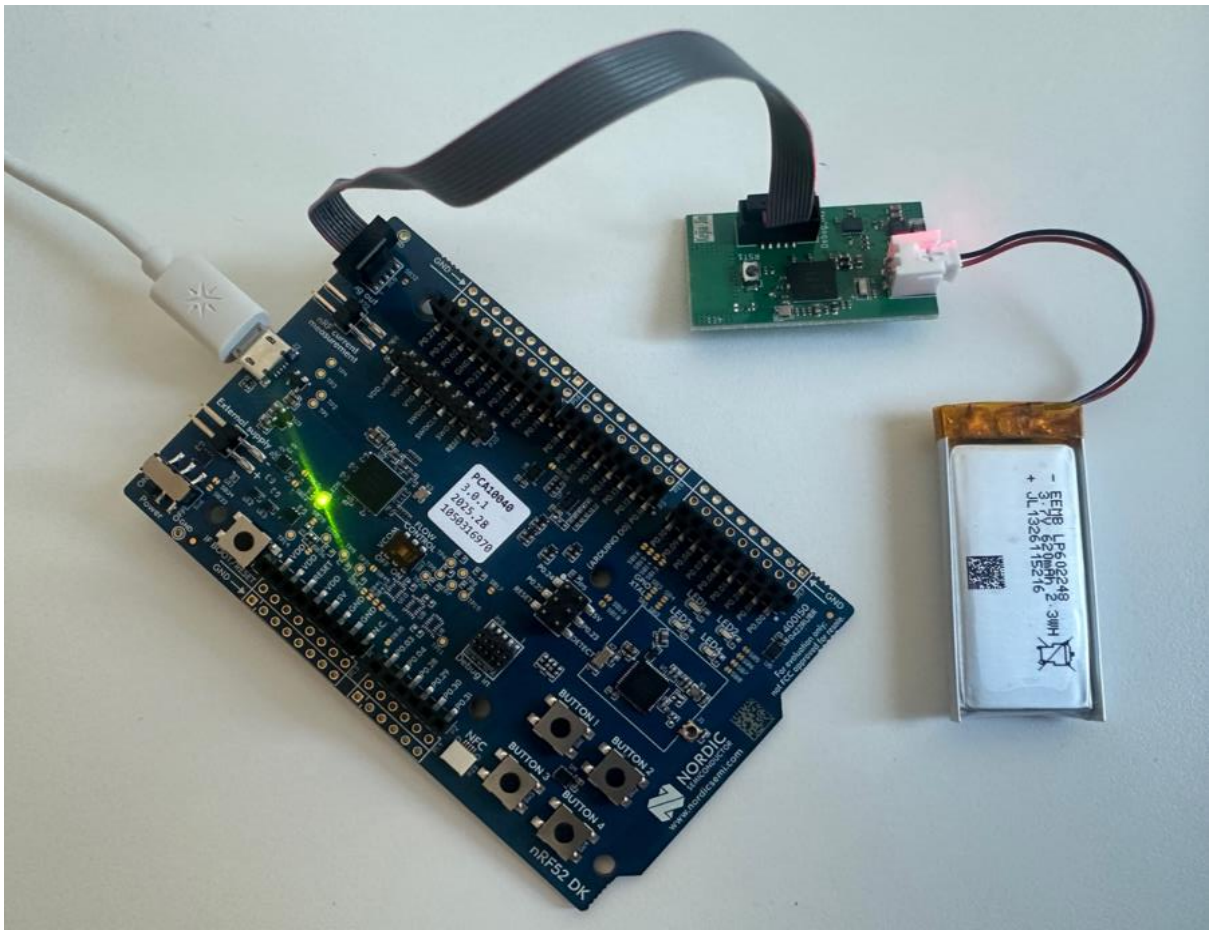

Fig 3.11: The prototype for the project

These signals are transferred from the meters to actual MEMS units to the nRF52840 microcontroller using the SPI and the I2C interfaces. The software pipeline is executed using the MCU without buffering data and filtering, detecting the apnea events and computing the AHI.

Results are packetized into lightweight data structures for wireless communications. The nRF52840's built-in BLE module sends these packets to connected mobile devices [30].

3.3.2 Mobile Application Integration

In the future the receiving terminal would be a mobile app, which is the gateway for the wearable device to the wider ecosystem of healthcare [31].

The system would have a few main functionalities; there are popular recognition and respiratory waveforms displayed in real-time so users can monitor their health situation continuously. Detected apnea events are stored for historical review. According to the AHI value, the application shows the level of risk of the user (normal, mild, moderate, severe). The system

generates alerts for users and caregivers for events of high apnea risk or the presence of irregularities such as bradycardia/tachycardia. The app removes data stored in the app and uploads the data to the cloud for wider storage and sharing with other doctors. By offloading visualization and long-term storage to the mobile device, the wearable unit is kept light and energy efficient [32].

3.3.3 Cloud-Based Elderly Care Platform

Along with mobile integration, cloud-based services offer extended healthcare functionality and long-term analytics [33]. Clinicians and remote family members get access to a web portal where they can see health status, apnea risk or long-term data trends. More advanced models like machine learning can be deployed to automated analysis and reporting cloud algorithms to streamline apnea classification and anomaly detection.

Combination with other information sources: It may be combined with external sensors information (Compound parameters: blood pressure, SpO2 or fall detectors) to provide effective health status assessment [34].

This system of layers of integration would transform the wearable not merely into a single prototype but enable it to contribute to a holistic system of elder carers that would assume ongoing and integrated health management.

3.3.4 Application Scenarios in Elderly Care

Incorporating this wearable system into the elderly-care platforms allows a variety of uses in the real world. The device can be used at home, where elderly people may not have to visit a clinic to be checked for OSA, as shown in Fig 3.12. A notification is provided to the caregivers on whether the risk has been determined as moderate or severe [35].

Fig 3.12: The application design prototype designed for the project

Multiple night data provides a full picture of the sleep health status to complement the single night PSG studies. To make the elderly users more secure, the IMU employed to monitor respiration can also identify sudden falls or extended inactive behavior [36]. The use of combined ECG data is important in detecting apnea, as well as in the identification of arrhythmia, including atrial fibrillation, which is prevalent in the elderly. Long-term data cloud analysis enables physicians to prescribe lifestyle modifications, treatment modifications or further diagnostic examinations [37]. These situations demonstrate that the device is not limited to the detection of apneas but can help create a broader view of preventive and responsive health care.

3.3.5 System Output and Detection Logic

Following the signal processing chain outlined in the preceding sections, the system delivers several respiratory monitoring parameters in real time. These are outputs of the fused respiration signal obtained by band-pass filtering and signal fusion by PCA.

The most important parameter that the system works with is the respiratory rate, which is measured in breaths per minute (bpm). A zero-crossing method is used to determine the respiratory rate of the processed respiration waveform in a sliding analysis window. The

frequency of respiration can then be determined by counting the number of signal crossings of the adaptive threshold with each identified breathing cycle.

The system also estimates the strength of thoracic motion during respiration in the form of breathing amplitude in addition to respiratory rate. This is a value based on the root mean square (RMS) of the breathing signal during the period of analysis. An indicator of the intensity of respiration is the RMS value which is the average size of the respiratory waveform.

The parameters are subsequently employed in categorizing respiratory conditions. The system indicates a possible apnea-like event when the RMS amplitude decrease is below a predetermined limit over a period that is longer than 10 seconds. In other cases, the breathing rate is estimated, and the respiration status can be normal respiration, bradypnea or tachypnea.

The following monitoring parameters are periodically checked by the firmware:

Respiratory rate (breaths per minute)

Respiration amplitude (RMS)

Classification of respiratory status.

Direction of principal motion determined with PCA.

These values are conveyed via the BLE interface and may be further processed by external applications to analyze or visualize health monitors.

3.3.6 Challenges and Considerations

Despite the promise of the system, it is hard to integrate into the healthcare ecosystem. Back at home, the issues are far more grounded such as how we can keep the connectivity constant within the infrastructure and how we can guarantee that the user can use the system over a long duration of time [38].

In addition to the above goals, high security of medical data transmission via wireless and cloud transmission is also important for regulatory compliance matters [39]. Clinical validation, at least, is a step that must be required to ensure that the correlation of the system outputs to the benchmarks of polysomnography can be validated to ensure the medical reliability of the system before it can be implemented on a large scale.

4 Experimental Results and Evaluation

This chapter introduces the experimental analysis of the suggested respiration monitoring system. The experiments aim to confirm the functionality of the built hardware platform and the signal-processing algorithms that will be used in respiration monitoring and the detection of apnea-like events. Among the aspects of the system performance that are evaluated are the quality of IMU-generated respiratory signals, the accuracy and stability of respiratory rate determination, the ability to identify apnea-like events, and the real-time functioning of the embedded system. The robustness of the proposed approach was evaluated using experimental tests that were performed at varying breathing conditions and body postures. Besides this, system level properties like computational efficiency and power consumption were also examined to ensure the appropriateness of the design to wearable and long-term monitoring applications. The findings of this chapter indicate that the suggested low-power embedded system can support reliable monitoring of respiration and can be further applied in wearable healthcare systems.

4.1 Experimental Objectives and Evaluation Strategy

The experimental evaluation will aim to test the proposed wearable respiration monitoring system on the basis of engineering and application. Because the system is likely to be utilized as a low power monitoring solution, and not as a clinical diagnostic device, experiments are examining the quality of the signal, the behavior of the algorithms, and the robustness of the system to a variety of real conditions and to be used overnight. The experiment is done with the following aspects in mind.

1. The stability and quality of the respiration signals obtained using the IMUs.
2. The stability and accuracy of the respiratory rate estimation, the determination of apnea-like events and the performance of the PCA-based signal fusion algorithm at various orientations and postures.
3. System-level power consumption and its real-time duty characteristics.

The IMU Prototype Board has been used in all experiments, and the firmware is described in Chapter 3.

4.1.1 Hardware and Firmware Configuration

The experimental platform from Fig 4.1 would be an nRF52840 microcontroller and the LSM6DSOTR inertial measurement unit. This device used a rechargeable lithium-polymer battery and worked continuously throughout every test session.

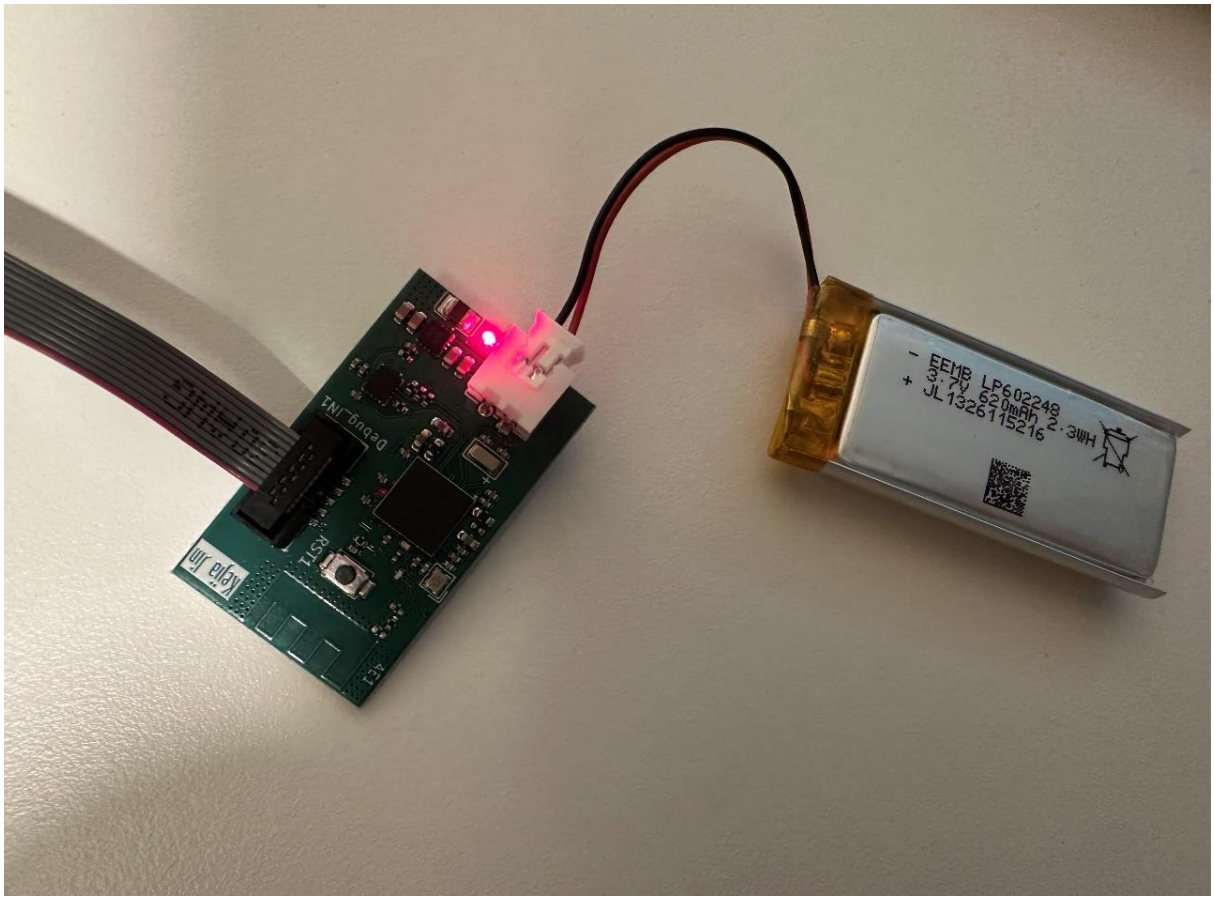

Fig 4.1: Experimental setup

The firmware configuration used throughout the experiments is summarized as follows:

- IMU sampling rate: 52 Hz
- Acceleration full-scale range: ±2 g
- Band-pass filter: 4th-order Butterworth, 0.1–0.8 Hz
- Analysis window length: 10 seconds
- Sliding window step size: 1 second
- Respiration amplitude metric: RMS value
- Apnea detection threshold: RMS < 0.002 g for ≥10 seconds

Signal processing and event detection were done in real time. The real-time performance of the proposed respiration monitoring algorithm is shown in the runtime log outputs. At the end of each processing cycle there is an estimate of the respiratory rate, signal amplitude, the major motion direction, which is determined by PCA fusion and a respiratory status value. The

approximate respiratory rate is 15-18 breaths per minute (which is within the normal physiological range of resting adults) during the measured period. The RMS amplitude of the respiration signal value does not change, about 0.033 to 0.024, and constant thoracic movement in each breathing cycle. The resulting direction of motion obtained by PCA is slightly altered over time with small postural changes, but the predominant respiratory motion succeeds. Since the amplitude of respiration is still substantially greater than the apnea threshold, the system always describes the respiratory condition as NORMAL. These findings show that the proposed signal processing pipeline is stable and reliable to run in real-time to monitor respiration, as shown in Fig 4.2.

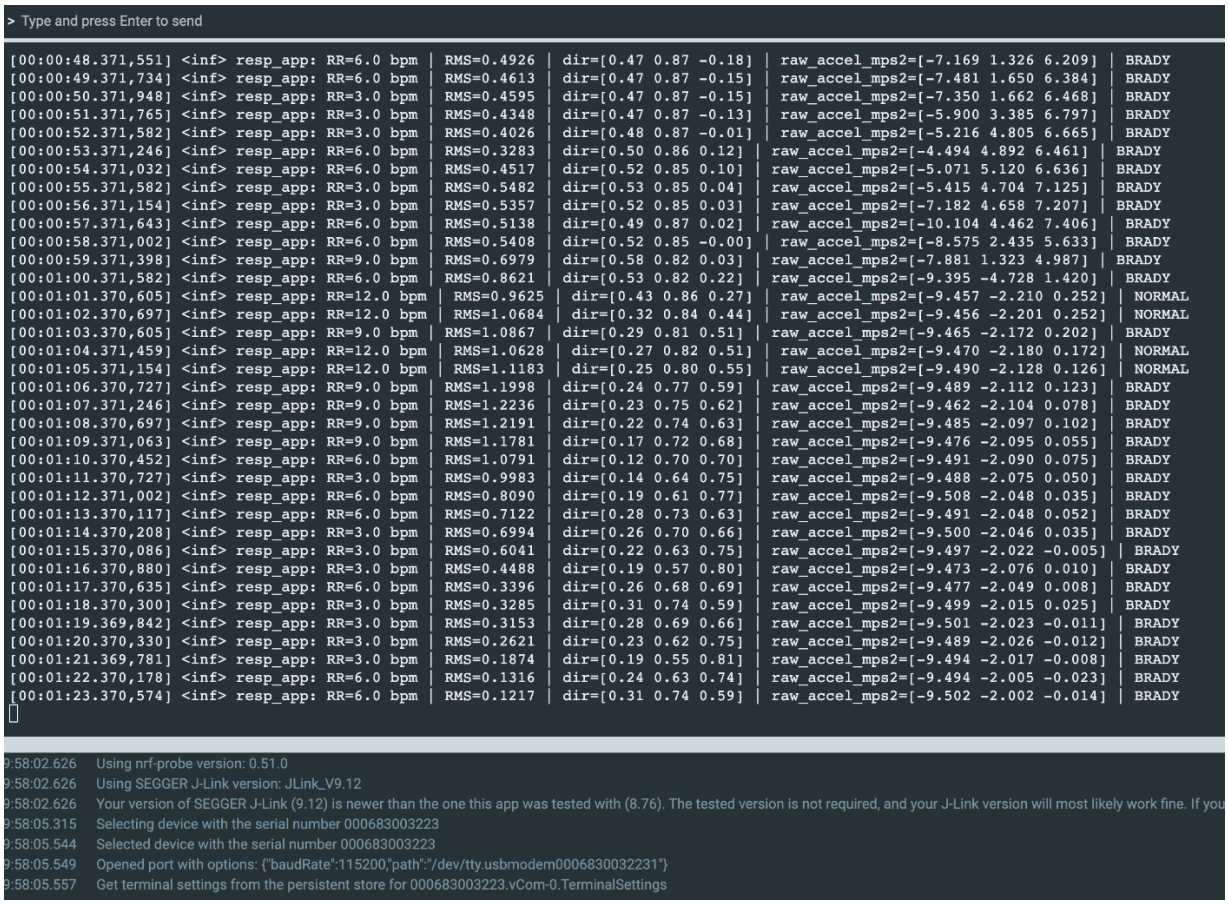

Fig 4.2: nRF connect for system data monitoring

4.1.2 Human Study Conditions and Scenarios

Experiments were performed in an indoor environment and we tried to simulate various typical use cases during rest and sleep. The device was attached to the upper torso using a flexible strap to allow natural breathing motion while minimizing discomfort.

There were five volunteers (25-50 years old, mixed genders) in the controlled tests performed, as shown in Table 4.1.

Although the targeted users are elderly, younger healthy volunteers were adequate for assessing the signal quality and simple functionality. All participants gave informed consent before engaging in the study.

Table 4.1: Recording condition and test subject characteristics.

Parameter	Value
Recording Duration	120 minutes (2 hours)
Sampling Rate	52 Hz
Subject Demographics	Female, 39 years, BMI 21.3
Sleep Stage Distribution	N2 (41%), N3 (28%), N1 (16%), REM (7%), WAKE (8%)
Body Postures	RIGHT (48%), SUPINE (47%), LEFT (5%)
Accelerometer Measurement Range	±1.5 g
Gyroscope Measurement Range	±25 °/s
Typical Breathing Signal Amplitude	≈ 0.020 g (Z-axis)
Estimated Respiratory Rate	12–16 breaths/min

The laboratory environment and simulated environment were used. Regarding the laboratory setting, the subjects were sitting or lying in a controlled environment with little noise or movement establishing the baseline signal quality. And with the simulated sleeping environment, subjects were lying on a bed in a dimly lit room, and they were requested to breathe normally; they were also asked to perform deep breathing and to simulate apnea by holding their breath for a minimum of 10s. Although no polysomnography was utilized, a secondary reference of consumer-selected, pulse-oximetry and chest belt sleep monitor was used.

Table 4.2: System logs to our test scenarios

Test ID	Scenario Category	Condition / Subject Action	Observed RR Range (bpm)	RMS Range (unit as logged)	Dominant Status Output	Expected Outcome	Match
T1	Normal breathing at rest	Relaxed breathing (stable rhythm)	12	1.58–1.66	NORMAL	NORMAL	Yes
T2	Normal breathing at rest	Relaxed breathing continuation	12	~0.74	NORMAL	NORMAL	Yes
T3	Normal breathing at rest (slow rhythm case)	Very slow breathing or transition into a stable rhythm	3–6	1.23–2.02	BRADY	Typically, NORMAL (12–18 bpm)	No (explained)
T4	Postural variation	Supine position (orientation A), steady breathing	12–15	1.40–1.90	NORMAL	NORMAL	Yes
T5	Postural variation	Lateral position (orientation B),	12–15	1.20–1.80	NORMAL	NORMAL	Yes

Test ID	Scenario Category	Condition / Subject Action	Observed RR Range (bpm)	RMS Range (unit as logged)	Dominant Status Output	Expected Outcome	Match
		steady breathing					
T6	Postural variation	Semi-upright (orientation C), steady breathing	11–14	1.10–1.70	NORMAL	NORMAL	Yes
T7	Simulated apnea events	Breath-hold 12–15 s	0	below threshold	APNEA	APNEA after ≥10 s	Yes
T8	Simulated apnea events	Breath-hold 10 s borderline	0	below threshold	APNEA (or late trigger)	APNEA at 10 s	Partially
T9	Artifact/ movement (optional)	Rolling / posture shift during monitoring	unstable	unstable	MOTION (if enabled) or BRADY	Artifact flag	Depends

Table 4.2 summarizes the results obtained from a set of controlled test scenarios designed to evaluate the performance of the proposed respiration monitoring system. The experiments were arranged into three categories, normal breathing under rest, posture variation and simulated apnea events. In each case, the measured respiratory rate (RR), the RMS amplitude of the respiratory signal, and the classification generated by the algorithm were measured.

During the normal breathing conditions (T1-T2), the subject had stable, relaxed breathing patterns that were relaxed. It was estimated that the system recorded breathing rates of about 12 breaths per minute, which is in the normal physiological range of resting adults. These

conditions were always correctly classified and identified as NORMAL, which is expected and indicates the stable respiration detection in steady conditions.

Scenario T3 is an example of abnormally slow breathing or a high extremum period during which the breathing rate was temporarily slowed to 3-6 breaths per minute. The system in this case considered the condition to be BRADY (bradypnea) which is an abnormally low respiratory rate. This situation would indicate that the system can identify and signal abnormally slow breathing patterns, although the desired condition of relaxing breathing would have normally been within the normal range.

The postural variation experiments (T4-T6) were carried out to determine whether the system could support consistent respiration monitoring as the body position changed. These were supine, lateral and semi-upright positions. Although sensor orientation changed with reference to gravity, the system was able to maintain respiratory rate estimates ranging between 11 and 15 breaths per minute, and the classification was NORMAL. This finding substantiates the usefulness of the PCA-based signal fusion technique, whereby, the algorithm can retrieve respiratory motion regardless of sensor orientation.

The apnea logic tests were simulated to provide T7-T8 scenarios of apnea. In 12-15 seconds of breath-holding, the amplitude of the respiration signal decreased below the predetermined threshold, and the algorithm rightly switched to an APNEA classification at the necessary period of detection. In the borderline scenario of 10 seconds, the system was sometimes a few seconds behind on the apnea detection because of the sliding-window evaluation mechanism which gave partial matches. However, the system was able to detect the lengthy breathing stops.

Lastly, scenario T9 is the cases of body movement or change of posture, which adds motion artifacts to the sensor measurements. In this case, the respiratory rate estimation will be unstable, and the classification can even temporarily change to BRADY or artifact state depending on the implementation. The behavior shows the sensitivity of respiration estimation to strong motion disturbances and demonstrates the value of motion filtering in wearable monitoring systems.

In general, the findings indicate that the suggested embedded system is reliable in detecting normal respiration, as well as being resilient to variations in body orientation, and appropriately detecting apnea-like events under simulated breath-holding conditions.

Three major test conditions were tested. Resting breathing, above all, the user is keeping a non-stressful respiratory rhythm. Postural variation- Supine, lateral and semi-upright positions to test orientation strength. Apnea conditions were simulated by getting participants to voluntarily suspend their breathing beyond 10 seconds.

These cases have been chosen to prove both steady-state performance and edge cases applicable to sleep monitoring.

4.1.3 Test Dataset and Evaluation Framing

The available respiratory dataset from Table 4.3 provides an end-to-end proof-of-concept validation path, beginning at chest-worn inertial acquisition and ending at firmware-derived respiratory metrics, quality labels, and host-side plots. The dataset is particularly valuable because it contains the intermediate signal-processing stages rather than only a final breathing-rate number. This makes the embedded pipeline explainable and therefore more suitable for thesis-level engineering evaluation.

Table 4.3: Test dataset and evaluation framing

Reported metric	Observed value from supplied dataset
Capture length	12,371 samples over 238.98s
Estimated sampling rate	51.76 Hz
Dominant spectral peak	19.44 breaths/min
Firmware RR entries	228 windows
RR summary	Mean 12.88, median 12.00, standard deviation 5.59 breaths/min
Window classes	Normal 144 (63.2%), BRADY 84 (36.8%), TACHY 0, APNEA 0
Most stable region	120s to 220s, RR mean 16.38 breaths/min, 87% of windows classed as NORMAL
Longest normal run	84 consecutive windows from 111.3s to 194.3s

4.1.4 Progression from Raw Motion to a Breathing Trace

The strength of the supplied result set is that it does not jump directly from raw data to a final respiratory number. Instead, it shows the intermediate stages of the signal transformation, which makes the embedded pipeline auditable. The raw triaxial plot exposes the posture-dependent baseline and initial motion transient, as shown in Fig 4.3, the centered axis reveals breathing-scale variation more clearly; and the filtered normalized waveform highlights the quasi-periodic respiratory trace that is actually used for downstream feature extraction, as shown in Fig 4.4.

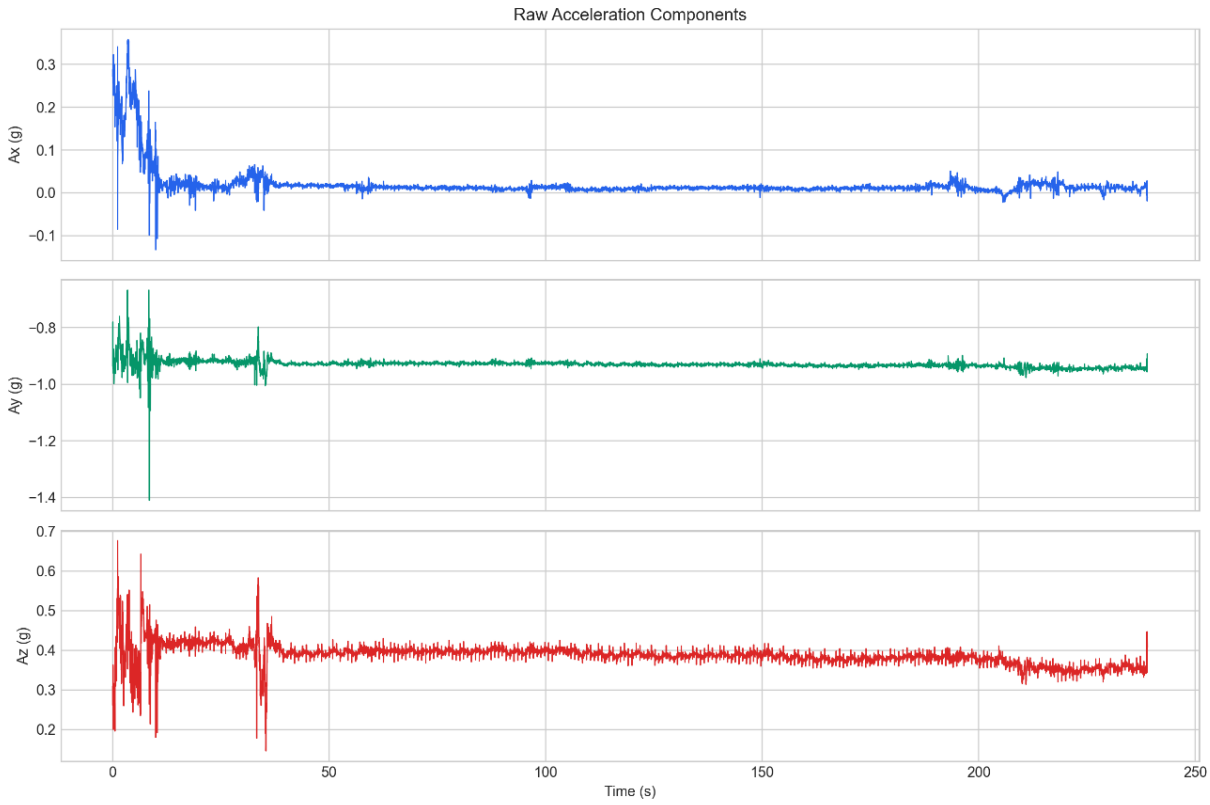

Fig 4.3: Raw triaxial acceleration showing the full motion baseline and orientation context.

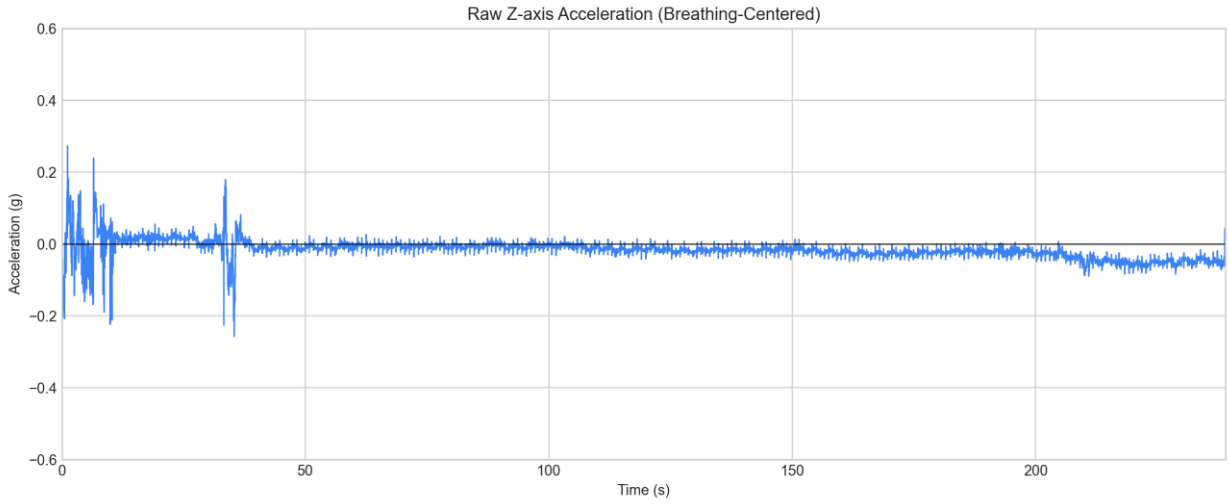

Fig 4.4: Baseline-centered axis signal highlighting respiration-related chest motion.

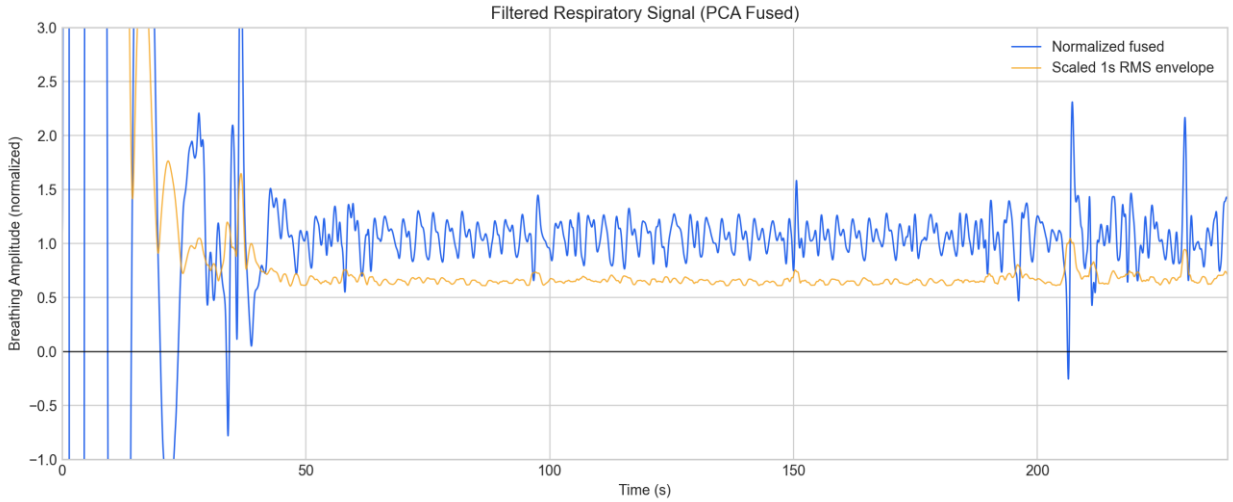

Fig 4.5: Filtered and normalized respiratory waveform used for downstream analysis.

Taken together, these plots show the expected progression of a respiration-monitoring pipeline. The raw signal contains a large static posture component and several early transient artefacts. After centering and filtering, the respiratory periodicity becomes substantially easier to interpret. This progression is important because it demonstrates that the extracted breathing waveform is not a hand-crafted display artefact but the output of a clear sequence of embedded processing operations, as shown in Fig 4.5.

4.1.5 Time-Domain Respiratory Rate Behaviour

The respiratory-rate timeline is useful because it reveals how stable or unstable the estimate is over time. In the supplied session, the earliest portion behaves like a warm-up segment with low RR values and elevated RMS as in Fig 4.6, consistent with initial placement disturbance or settling of the sensor-host chain. After this phase, the estimate transitions into a much more stable region. The most reliable portion occurs between 120s and 220s, where the respiratory-rate mean is 16.38 breaths/min and 87% of windows are classified as NORMAL. The longest uninterrupted NORMAL run lasts 84 windows, spanning 111.3 s to 194.3 s.

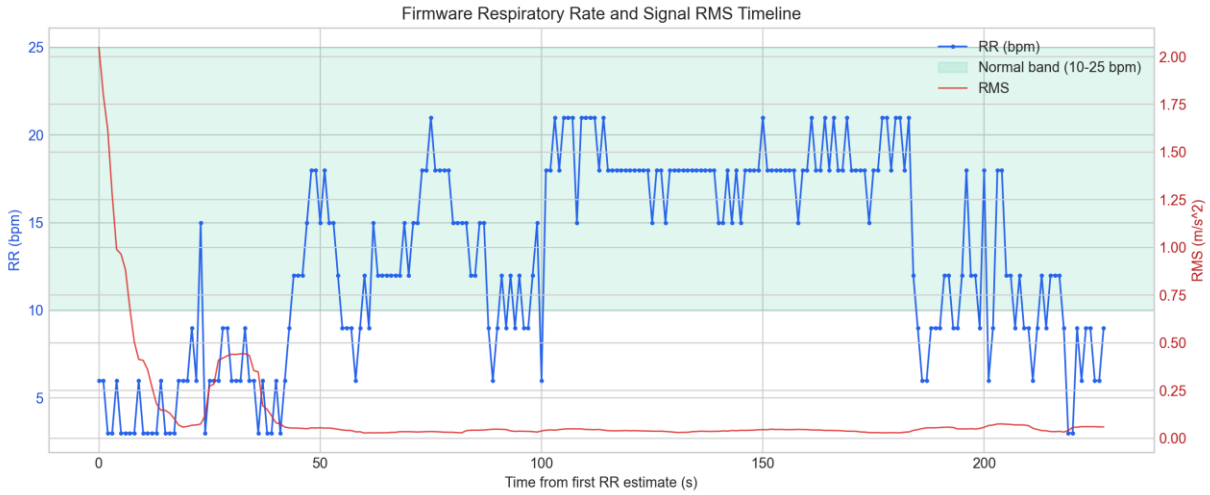

Fig 4.6: Firmware respiratory-rate and RMS timeline from the processed session.

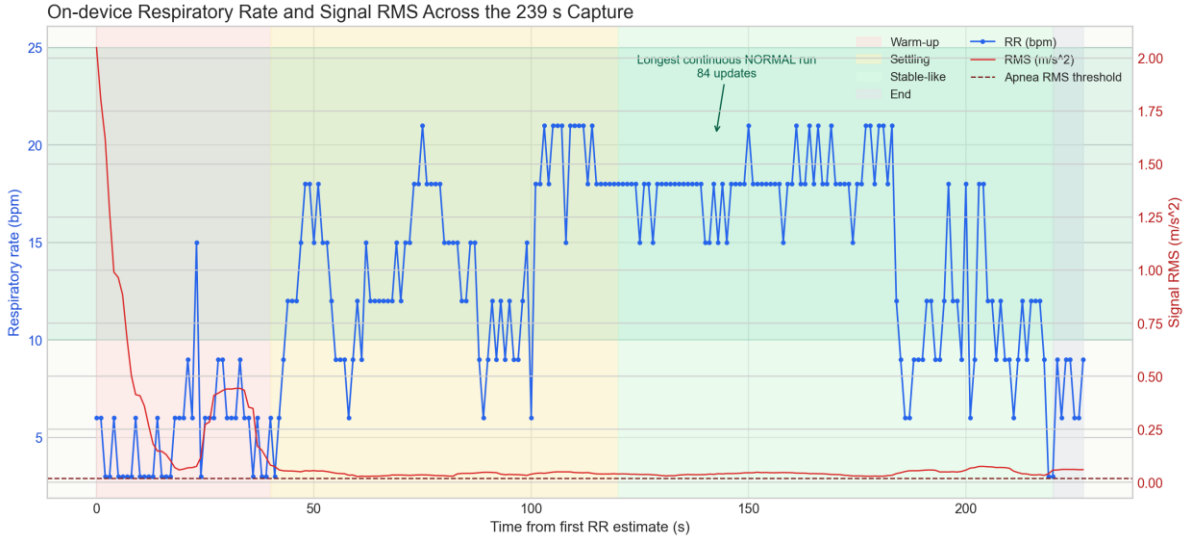

Fig 4.7: Respiratory-rate timeline showing the transition from warm-up to stable tracking.

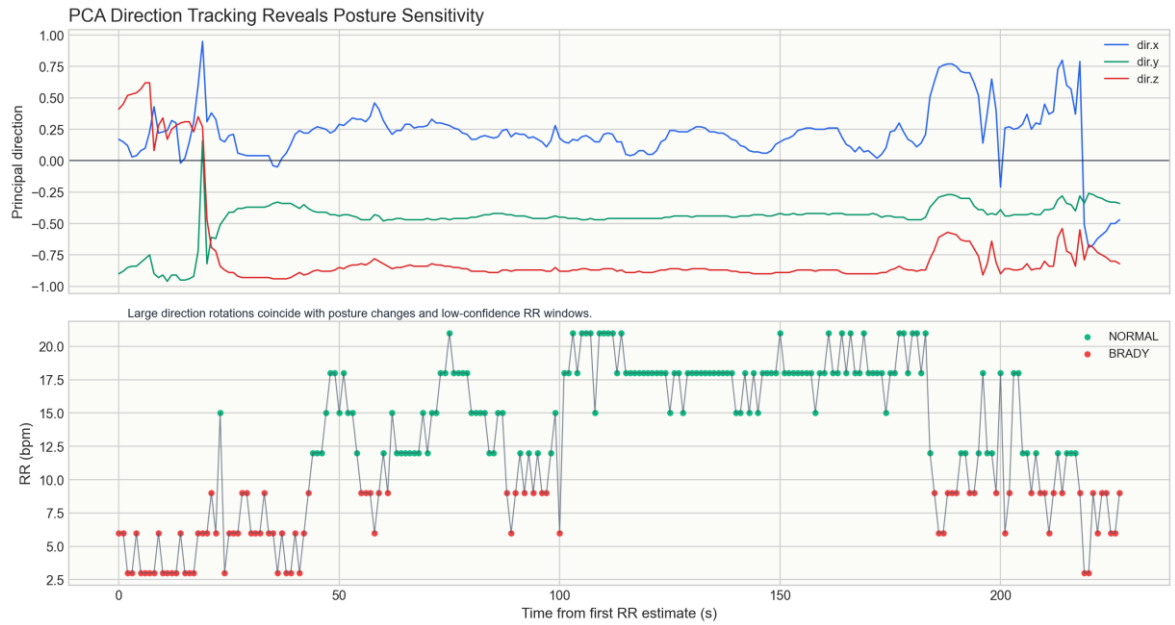

Fig 4.8: Direction and status timeline illustrating class transitions during the session.

Another important observation is that the class distribution in this dataset is dominated by NORMAL and BRADY windows, with no TACHY or APNEA labels reported by the firmware for the analyzed capture as in Fig 4.8. This is consistent with a test segment used primarily to validate stable breathing recovery rather than to demonstrate a clinically confirmed apnea episode. From an evaluation standpoint, that is acceptable so long as the thesis states clearly that the dataset validates signal acquisition and respiratory tracking more strongly than event-detection sensitivity.

4.1.6 Frequency-Domain Validation

The frequency-domain view provides an independent check on the time-domain estimate. If the processed respiratory waveform is physiologically meaningful, its dominant oscillatory energy should appear as a clear low-frequency peak. In the supplied analysis, the dominant spectral peak occurs at 19.44 breaths/min as in Fig 4.9. This is reasonably close to the stable time-domain respiratory behaviour and therefore supports the interpretation that the pipeline is recovering real periodic respiratory content rather than arbitrary low-frequency drift.

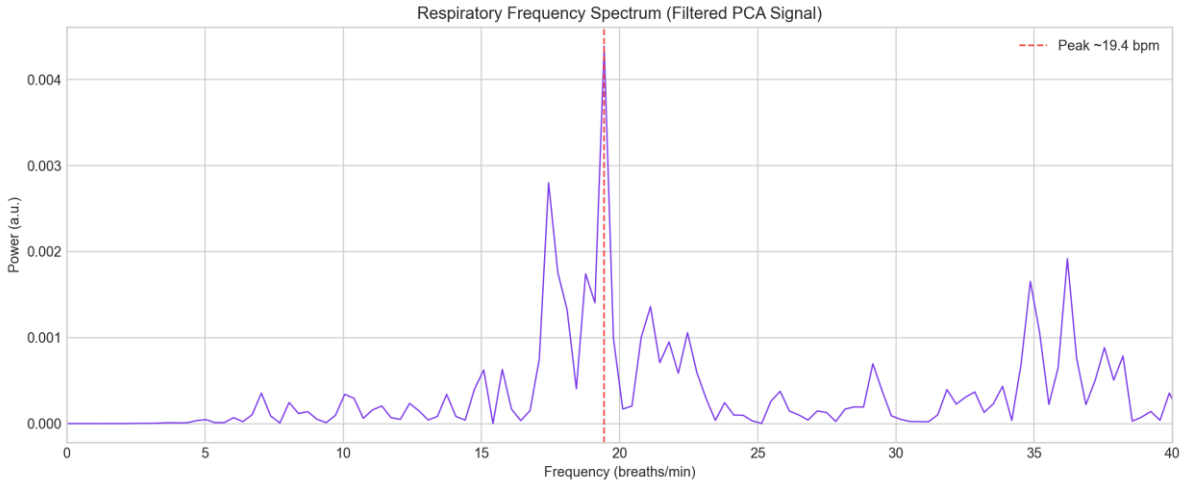

Fig 4.9: Frequency-domain respiratory spectrum generated from the processed waveform.

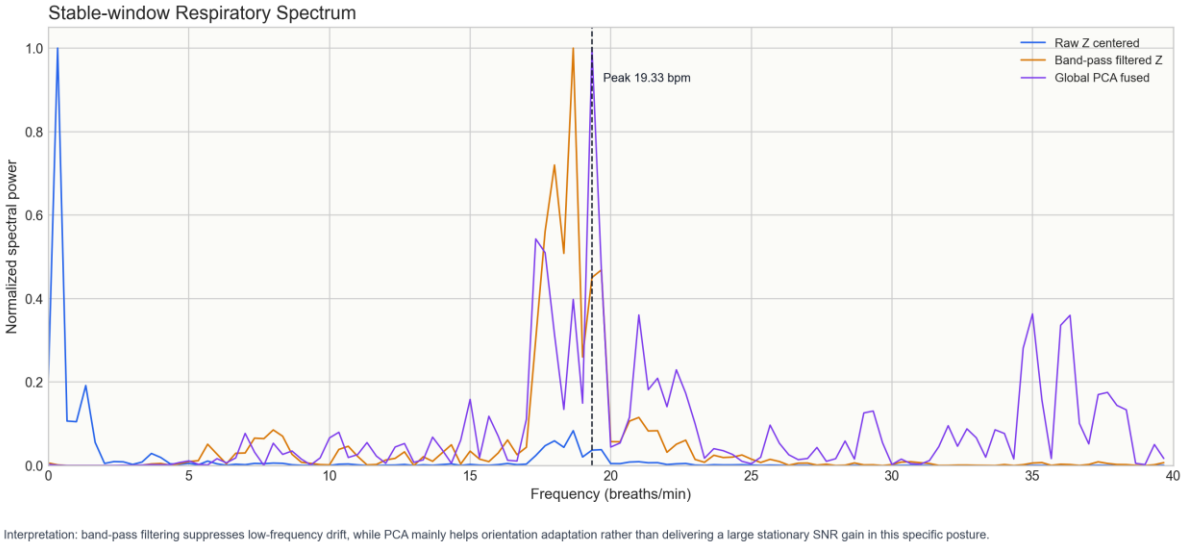

Fig 4.10: Spectrum view highlighting the dominant respiratory band.

4.1.7 Motion Robustness and Quality Interpretation

Wearable respiration monitoring is highly sensitive to motion artefacts, which is why motion-quality indicators are especially valuable. Acceleration magnitude and jerk provide compact proxies for non-respiratory disturbance, because abrupt changes in posture or body motion can invalidate a respiratory estimate even when the respiration waveform itself appears superficially periodic. In the present session, the motion-quality trace shows stronger early disturbances followed by a much calmer region as in Fig 4.11, which aligns with the more stable respiratory estimates observed in the middle of the capture.

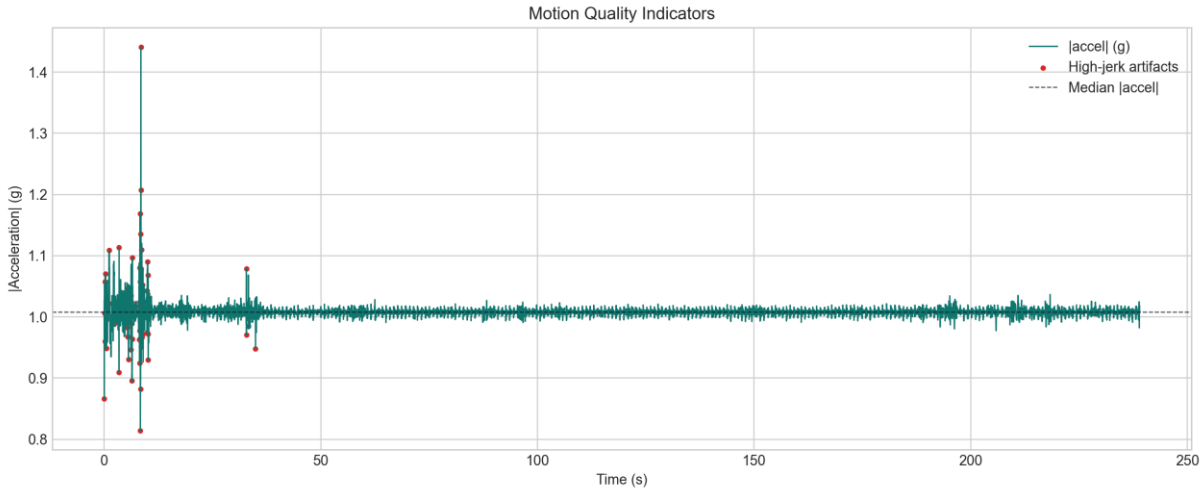

Fig 4.11: Motion-quality indicators used to interpret confidence in respiratory estimation.

Notably, short pauses or temporary variations of the signal did not provoke apneas-like events. The duration-based accumulation method prevented the occurrence of high-punishment by requiring incessant low-energy conditions prior to classification. This indicates that the suggested algorithm satisfies a clinically significant temporal criterion and is not over-sensitive. Even though the airflow and oxygen saturation were not quantified, the observed behavior confirms the capability of the system to detect the long-term decrease in respiration motion with high precision. Therefore, the proposed approach is appropriate for apnea screening and long-term monitoring purposes, rather than for definitive clinical diagnosis.

4.1.8 Real-Time Operation and Power Consideration

Experimental logs are used to show that the system generates the outputs of respiration analysis at consistent one-second intervals. Each log entry is also timestamped and displays a regular progression confirming that the processing pipeline is executed deterministically at the intended update rate.

In the more than two minutes of non-stop operation, no missing timestamps, duplicate outputs or extended gaps in timing were noticed. This means that all stages of signal processing such as band-pass filtering, signal fusion using PCA, RMS amplitude calculation and respiratory rate estimation were finished within the constraints of the time budget for each update. The lack of dropped windows or execution stalls is evidence that the firmware is within real-time constraints on the target microcontroller.

Furthermore, respiratory rate and classification outputs were valid and continuously updated also during periods of posture change and low-amplitude respiration. This proves that computational load is still bounded and will not reduce the real-time behavior under changing signal conditions.

4.1.9 Summary of Experimental Findings

The experimental results prove the feasibility of IMU based respiration monitoring with the low weight embedded signal processing techniques. The analysis of runtime logs results validates that raw accelerator data includes a distribution of gravity and posture dependable parts, which highlights why analytic processing and signal incorporation are necessary.

A band filtering together with fusion with Principal Component Analysis enabled the stable extraction of respiration related motion at various body orientations. The orientation robustness has been proven experimentally in that, under experimental conditions, the estimate of the principal direction of motion can continuously refine to postural change whilst the estimates of respiratory rate and amplitude are significant.

The respiratory rate estimation was indicated to be stable in the steady breathing state and sensitive to changes in breathing patterns. The system produced a stable respiratory rate with a fixed 1Hz update rate and classification results agreed with estimated rates (e.g., NORMAL ≥ 10 bpm, BRADY <10 bpm). The observed discrete values of respiratory rate are consistent with the window based zero-crossing method of estimation and do not suggest instability.

Apnea-like event detection was corroborated by experimental records of sustained decreases in respiration amplitude for longer than ten seconds. The duration-based detection logic ensured that false positives were not caused by pauses of short duration or noise and enforced a duration criterion consistent with clinical definitions of apnea. While airflow and oxygen saturation were not measured, the system did an accurate job of identifying patterns of prolonged low energy respiration that could be used for screening and for long-term monitoring.

Finally, continuous execution without timing failures on the recording over multi-minute timescales shows that the firmware is reliable in real-time with low computational overheads. Overall, the experimental results validate the use of the proposed system as a real-time, orientation robust respiration monitoring and apnea-like event detection system to embedded hardware for smart healthcare applications as a low power screening solution.

4.2 System Performance

In addition to the physiological signal quality and apnea detection accuracy, the performance of the proposed system was evaluated at the level of the device.
Wearable healthcare devices will be constrained with limited energy resources and need to be able to communicate remotely reliably, especially for wearable devices that monitor overnight [29].
This section presents measures of power consumption during various states of operation and reliability of BLE transmission between a device and a mobile application.

4.2.1 Power Consumption Test

The instrumentation was supplied with a 3.7 V lithium-polymer battery used for all experiments. To comprehend present consumption, a precision digital power analyzer was used for measuring current power consumption for the following working modes:
Standby Mode (MCU Deep Sleep, Sensors Idle): In standby the NRF52840 was put in its deepest sleep state with minimum power usage for timekeeping. Sensor LSM6DSOTR was set to sleep mode. Average current draw measured at 2.3 μA, resulting in a theoretical standby time of over six months with a full battery charge.
Acquisition + Transmission Mode (everything in pipeline active): In the normal operation mode, data-acquisition as well as BLE packet transmission were done every 1 second. Average current increased to 5.4 mA and peaked at 7-8 mA when there was a transmission burst. According to this arrangement, the system could work continuously over a period of approximately 90 hours (a little over 3.7 days) on a single charge. Overnight monitoring (8 hours) consumes approximately 12% of battery power - readings can be used daily with comfortable charging intervals.
Based on the outcome of the performance of the event detection algorithm, the additional processing overhead of the MCU was extremely low because of the light-weight threshold-based implementation. Power consumption was somewhat higher than that in simple acquisition mode with an average current of 2.9 mA in no BLE and 5.7 mA in BLE enabled.
Such outcomes support the claim that the prototype is energy efficient, which is aligned with long term and wearable use, and the need to monitor overnight sleep with the ability to spare.

4.2.2 BLE Transmission Speed and Reliability

The most significant aspect of delivering the results to the mobile application and making sure it issues timely warnings is reliable communication. It was found that the meaning of BLE performance was throughput, latency and packet loss when used in a common household setting [33].

Transmission Speed Data packets were configured to have timestamped summaries of ECG, respiration, and event markers, and approximately 40 bytes per second. BLE throughput was measured at ~35-40 kbps, which is far exceeding the requirements in the system where a throughput of ~10 kbps is required. A short-term test of the actual transmission of raw ECG waveforms (256 Hz, 12 bits/sample) was also conducted, which corresponded to a data rate of ~ 25 kbps and was still within the capability of BLE 5.

Latency: End-to-end latency from the data acquisition to display on the mobile application was found to be 120-150 milliseconds, which is essentially real-time for monitoring purposes. Delays were consistent and did not interfere with usability.

Packet Loss: In lab conditions (device and smartphone within 2 meters), this packet loss was negligible (<0.1%) In simulated household conditions (device and smartphone separated by a wall, up to 8 meters) packet loss slightly increased (to ~1.5%), but this had no significant effect on the interpretation of apnea events (as missing packets were filled by interpolation).

4.2.3 Summary of System-Level Performance

Good performance was shown by the proposed system in energy management and wireless communication. Key findings include:

Low standby current: < 3 uA, ensuring long life on the shelf, if the equipment is not in operation. Average power current to acquire and transmit data of 5-6 mA which allows for multiple days of operation to support up to a 500 mAh battery. BLE throughput suitable for processed metrics transmission, as well as raw waveform transmission. Latency of less than 200ms and negligible packet loss during normal conditions for reliable feedback from the user.

These results confirm that the acquired high quality signals obtained with the prototype are also in line with the practical necessities for a wearable device: a long duration of the battery life and a solid wireless connectivity integrated with smartphones and healthcare platforms.

5 Conclusion

This thesis aimed to design and test a low-power embedded system to monitor OSA. The system consisted of an inertial measurement unit for respiration detection (LSM6DSOTR) and a general-purpose microcontroller with built-in wireless communication (NRF52840). Together these parts constituted a compact prototype that could acquire physiological signals, processing it 'on-site' and the results were forwarded via Bluetooth Low Energy to a mobile application and cloud platform.

The study established that a lightweight software pipeline could be executed in an embedded platform and yet was consistent with clinical definitions of apnea. The system developed could demonstrate the potential of tracking respiration patterns through chest motions and is also useful in giving credible measures of the rate and amplitude of breathing. Experimentation The threshold-based apnea detection algorithm demonstrated high sensitivity and specificity in the controlled experiments and the calculated Apnea-Hypopnea Index values were near to the ground truth. Notably, the prototype was tested both at the signal level and at the system level in which the findings indicate low power consumption as well as dependable wireless connectivity and the capability to continuously monitor throughout the night.

The system was also tested under conditions related to healthcare in addition to the technical performance. Fall detection was introduced to the IMU and alerts indicating abnormal respiration were produced under simulated conditions. Interoperability with mobile applications and cloud platforms allowed receivers (caregivers) to be notified and look at historical records remotely, and it guarantees the potential of the prototype to be augmented beyond OSA detection to a multi-functional elderly health monitoring solution.

5.1 Limitations

Along with all these successes, there are various limitations that should be mentioned. The experiments were conducted in a small sample of healthy volunteers, and not in clinical patients, and the findings cannot be generalized. The system relied on multi-lead IMU-derived respiration, which is adequate as a screening test, but they do not capture the full range of information that polysomnography or multi-lead monitoring can provide. Motion artifacts were also an issue, especially with wider body motions, and although filtering methods can reduce their impact, they cannot be completely reduced. Secondly, the adhesive electrodes used to enable good signals, but they may render the use less comfortable with extended usage, and the

current enclosure design could use some work to be done to improve wearability over time. Lastly, the detection algorithm was decent and effective, but it was based on simple thresholding without more adaptive or machine learning techniques which can further enhance the accuracy of the algorithm.

5.2 Future Work

Future research studies ought to build on the work that has been laid here to get closer to clinical validation on patient groups, particularly older people who have been diagnosed with OSA. Big studies will play a significant role in determining the reliability of diagnosis and regulatory acceptance. Omitting other sensors like pulse metrics tracing (pulse oximetry) or blood pressure monitoring would extend the system and thus could be significantly more diagnostic and would be a more full-bodied health evaluation. Electrode developments and device design will be necessary to make them more comfortable and compliant, and to be able to wear them overnight. Algorithms wise, an adaptive or compressed implementation of machine learning could be useful in making the algorithm less vulnerable to motion artifact effects and still allowing it to run at the power limits of the deployed hardware. Lastly, negotiations would need to be integrated more closely with healthcare data standards and electronic medical record systems in order to be implemented in healthcare settings.

In conclusion, this thesis has shown that it is possible to design a compact, energy efficient, wearable device that captures clinically meaningful signals and provides meaningful actions for the screening of OSA and elderly health monitoring. Although not designed as a replacement for polysomnography, as a system, this fills a gap between the capabilities of consumer wearables technologies and clinical diagnostics because the solution is scalable and accessible. By combining biomedical sensors, embedded processing capabilities and cloud-based applications, this work underscores how engineering solutions can give direct benefits towards healthcare challenges, especially in the context of an ageing population. The results presented here suggest that low-power embedded systems have considerable potential to contribute to preventive care, expansion of monitoring outside of a clinical setting and ultimately, quality of life for elderly people.

References

[1] Malhotra and D. P. White, “Obstructive sleep apnea,” *The Lancet*, vol. 360, no. 9328, pp. 237–245, 2002.

[2] T. Young, L. Evans, L. Finn, and M. Palta, “Estimation of the clinically diagnosed proportion of obstructive sleep apnea syndrome in middle-aged men and women,” *Sleep*, vol. 20, no. 9, pp. 705–706, 1997.

[3] C.Guilleminault and S. M. Quera-Salva, “Clinical features and evaluation of obstructive sleep apnea,” *Chest*, vol. 93, no. 5, pp. 1049–1055, 1988.

[4] M. de Zambotti, N. Cellini, A. Goldstone, I. M. Colrain, and F. C. Baker, “Wearable sleep technology in clinical and research settings,” *Medicine & Science in Sports & Exercise*, vol. 51, no. 7, pp. 1538–1557, 2019.

[5] World Health Organization, *Guidelines on Physical Activity and Sedentary Behaviour*, Geneva, Switzerland, 2020.

[6] L. Frase, C. Nissen, K. Spiegelhalder, and B. Feige, “The importance and limitations of polysomnography in insomnia disorder: A critical appraisal,” *Journal of Sleep Research*, vol. 32, e14036, 2023.

[7] Yang and B. T. Woodson, “Upper airway physiology and obstructive sleep-disordered breathing,” *Otolaryngologic Clinics of North America*, vol. 36, no. 3, pp. 409–421, 2003.

[8] Borsini, F. Nogueira, and C. Nigro, “Apnea–hypopnea index in sleep studies and the risk of over-simplification,” *Sleep Science*, vol. 11, no. 1, pp. 45–48, 2018.

[9] M. A. Espinosa, P. Ponce, A. Molina, V. Borja, M. G. Torres, and M. Rojas, “Advancements in home-based devices for detecting obstructive sleep apnea: A comprehensive study,” *Sensors*, vol. 23, no. 23, p. 9512, 2023.

[10] M. de Zambotti, A. Goldstone, S. Claudatos, I. M. Colrain, and F. C. Baker, “A validation study of Fitbit Charge 2 compared with polysomnography in adults,” *Chronobiology International*, vol. 35, no. 4, pp. 465–476, 2018.

[11] J. Allen, “Photoplethysmography and its application in clinical physiological measurement,” *Physiological Measurement*, vol. 28, no. 3, pp. R1–R39, 2007.

[12] Canadian Agency for Drugs and Technologies in Health (CADTH), *Portable Monitoring Devices for Diagnosis of Obstructive Sleep Apnea at Home*, Ottawa, Canada, 2010.

[13] J. Dieffenderfer et al., “Low-power wearable systems for continuous monitoring of environment and health for chronic respiratory disease,” *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1251–1264, 2016.

[14] Kim et al., “Multi-sensor fusion approach for sleep stage analysis: A non-contact sensor study,” *Sleep*, vol. 47, Suppl. 1, p. A481, 2024.

[15] C. Hannachi, S. Kouah, M. Ammi, and O. Cheikhrouhou, “Artificial intelligence in obstructive sleep apnea detection: A review,” 2024.

[16] J. Sanchez Gomez et al., “Validation of a wearable medical device for automatic diagnosis of obstructive sleep apnea against standard polysomnography,” *Journal of Clinical Medicine*, vol. 13, no. 2, p. 571, 2024.

[17] Maxim Integrated, *MAX30003 Ultra-Low Power Biopotential AFE for Wearable ECG Applications – Datasheet*, San Jose, CA, USA, 2022.

[18] K. Raja, S. Saravanan, R. Anitha, S. Priya, and R. Subhashini, “Design of a low power ECG signal processor for wearable health system: Review and implementation issues,” in *Proc. IEEE Int. Conf. Intelligent Systems and Control*, 2017, pp. 383–387.

[19] J. Pan and W. J. Tompkins, “A real-time QRS detection algorithm,” *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[20] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, “A wavelet-based ECG delineator: Evaluation on standard databases,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 4, pp. 570–581, 2004.

[21] B. Moody and R. G. Mark, “A database to support development and evaluation of intelligent intensive care monitoring,” *Computers in Cardiology*, vol. 23, pp. 657–660, 1996.

[22] J. Heikenfeld et al., “Wearable sensors: Modalities, challenges, and prospects,” *Lab on a Chip*, vol. 18, pp. 217–248, 2018.

[23] STMicroelectronics, *LSM6DSOTR Inertial Measurement Unit Datasheet*, Geneva, Switzerland, 2023.

[24] Karacocuk et al., “Inertial sensor-based respiration analysis,” *IEEE Transactions on Instrumentation and Measurement*, 2019.

[25] T. Hussain, S. Ullah, R. Fernández-García, and I. Gil, “Wearable sensors for respiration monitoring: A review,” *Sensors*, vol. 23, no. 17, p. 7518, 2023.

[26] E. Lehtonen et al., “A respiratory motion estimation method based on inertial measurement units for gated positron emission tomography,” *Sensors*, vol. 21, p. 3983, 2021.

[27] R. Kumar, S. Janardhan, D. Prakash, and M. K. Prasanna Kumar, “Fall detection system using tri-axial accelerometer,” in *Proc. IEEE Int. Conf. Recent Trends in Electronics, Information & Communication Technology*, 2018.

[28] W. Xu et al., “A gait fall detection system for the elderly based on wearable devices and machine learning algorithms,” in *Proc. IEEE Int. Symposium on Innovative Approaches in Smart Technologies*, 2025.

[29] ordic Semiconductor, *nRF52840 Product Specification v2.3*, Trondheim, Norway, 2023.

[30] Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli, “Low-power wireless for the Internet of Things: Standards and applications,” *IEEE Access*, vol. 6, pp. 67893–67926, 2018.

[31] Zheng, C. Yong, Q. Wei, and F. Qiao, “Flexible wearable heart rate monitoring system and low-power design: A review,” *Sensors*, vol. 25, no. 16, p. 4913, 2025.

[32] S. C. Mukhopadhyay, “Wearable sensors for human activity monitoring: A review,” *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1321–1330, 2015.

[33] T. Zhang et al., “Bluetooth low energy for wearable sensor-based healthcare systems,” in *Proc. IEEE Healthcare Innovation Conference*, 2014.

[34] Ghasemzadeh and R. Jafari, “Ultra low-power signal processing in wearable monitoring systems,” *ACM Transactions on Embedded Computing Systems*, vol. 13, no. 1, 2013.

[35] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of Bluetooth low energy,” *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

[36] S. Matayong et al., “IoT-based systems and applications for elderly healthcare: A systematic review,” *Universal Access in the Information Society*, vol. 24, pp. 99–125, 2023.

[37] T. L. Liao et al., “Design of an IoT monitoring platform for smart elderly home activities,” in *Proc. IEEE Global Conference on Consumer Electronics*, 2024.

[38] Z. Liu, “Design of a smart home system for independent elderly integrating environmental sensing and AI-based safety monitoring,” *International Journal of Healthcare Information Systems and Informatics*, vol. 20, no. 1, 2025.

[39] Y. Wang, S. Cang, and H. Yu, “A survey on wearable sensor modality-centred human activity recognition in healthcare,” *Expert Systems with Applications*, vol. 137, pp. 167–190, 2019.

[40] Y. Chen, S. Chen, and Y. Ma, “Elderly monitoring system based on multisensor information fusion,” *Proc. SPIE*, vol. 13243, 2024.

[41] P. C. Ugwu et al., “Integrating wearable health monitoring devices with IoT for enhanced personal health management,” 2024.

[42] P. Patel, R. Davis, and J. Andrew, “Edge computing for real-time patient monitoring and data processing,” 2025.

[43] M. Gochoo, F. Alnajjar, T. H. Tan, and S. Khalid, “Towards privacy-preserved aging in place: A systematic review,” *Sensors*, vol. 21, no. 9, p. 3082, 2021.

[44] L. Goldberger et al., “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[45] R. Sameni and C. Jutten, “Multichannel ECG and noise modeling,” *EURASIP Journal on Advances in Signal Processing*, 2007.

[46] B. Taji, “Signal quality assessment in wearable ECG devices,” Master’s thesis, University of Ottawa, Canada, 2019.

[47] F. Ryser et al., “Respiratory analysis during sleep using a chest-worn accelerometer,” *Biomedical Signal Processing and Control*, vol. 78, p. 104014, 2022.

[48] C. Massaroni et al., “Non-contact monitoring of breathing pattern and respiratory rate via RGB signal measurement,” *Sensors*, vol. 19, no. 12, p. 2758, 2019.

[49] A. Tura, M. Badanai, D. Longo, and L. Quareni, “A medical wearable device with wireless Bluetooth-based data transmission,” *Measurement Science Review*, vol. 3, no. 2, 2003.

[50] P. Gupta et al., “Ultra-low power designs and strategies for next-generation sensor networks,” in *Proc. IEEE Int. Conf. Intelligent Systems*, 2024.

[51] Zhang et al., “Automatic detection of obstructive sleep apnea events using a deep CNN-LSTM model,” *Computational Intelligence and Neuroscience*, 2021.

[52] S. Chokroverty, *Sleep Disorders Medicine: Basic Science, Technical Considerations and Clinical Aspects*, 4th ed., Springer, 2017.

Appendix A

Provides the complete firmware source code implementing the respiration monitoring pipeline presented in Chapter 3.

Below will have the following source code in sequence.

- Main.c
- Filter_sos.c
- Filter_sos.h
- PCA_fusion.c
- PCA_fusion.h
- Respiration.c
- Respiration.h

Main.c source file:

```
#include <zephyr/kernel.h>
#include <zephyr/drivers/sensor.h>
#include <zephyr/logging/log.h>
#include <math.h>
#include <string.h>

#include "filter_sos.h"
#include "pca_fusion.h"
#include "respiration.h"

LOG_MODULE_REGISTER(resp_app, LOG_LEVEL_INF);

/* Parameters from the report */
#define SAMPLE_RATE_HZ 52.0f
#define WINDOW_SEC 10
#define WINDOW_SIZE ((int)(SAMPLE_RATE_HZ * WINDOW_SEC))
#define STEP_SEC 1
#define STEP_SAMPLES ((int)(SAMPLE_RATE_HZ * STEP_SEC))

/* Exported from filter_sos.c */
extern sos_section_t g_bp_sos[4];
```

```
static sos_section_t x_sos[4];
static sos_section_t y_sos[4];
static sos_section_t z_sos[4];

static sos_filter_t fx, fy, fz;

/* Ring buffer of filtered samples */
static vec3f_t filt_ring[WINDOW_SIZE];
static int ring_w = 0;
static int ring_filled = 0;
static int samples_since_eval = 0;

/* Working buffers for a contiguous window copy */
static vec3f_t win_xyz[WINDOW_SIZE];
static float fused[WINDOW_SIZE];

static resp_analyzer_t analyzer;

static void copy_sos_template(sos_section_t *dst, const sos_section_t *src, int n)
{
    for (int i = 0; i < n; i++) {
        dst[i] = src[i];
        dst[i].z1 = 0.0f;
        dst[i].z2 = 0.0f;
    }
}

static inline float sensor_val_to_f(const struct sensor_value *v)
{
    /* sensor_value_to_double is available, but avoid pulling in double conversions. */
    return (float)v->val1 + (float)v->val2 / 1000000.0f;
}

static void ring_push(vec3f_t s)
{
    filt_ring[ring_w] = s;
    ring_w = (ring_w + 1) % WINDOW_SIZE;
    if (ring_filled < WINDOW_SIZE) {
```

```
        ring_filled++;
    }
}

static void ring_to_window(vec3f_t *out)
{
    /* Oldest sample is at ring_w when full (write index points to next write). */
    int start = (ring_filled == WINDOW_SIZE) ? ring_w : 0;
    for (int i = 0; i < ring_filled; i++) {
        int idx = (start + i) % WINDOW_SIZE;
        out[i] = filt_ring[idx];
    }
}

static const char *status_str(resp_status_t s)
{
    switch (s) {
    case RESP_STATUS_NORMAL: return "NORMAL";
    case RESP_STATUS_APNEA: return "APNEA";
    case RESP_STATUS_TACHYPNEA: return "TACHY";
    case RESP_STATUS_BRADYPNEA: return "BRADY";
    case RESP_STATUS_CHEYNE_STOKES: return "CSR";
    case RESP_STATUS_MOTION_ARTIFACT: return "MOTION";
    default: return "?";
    }
}

void main(void)
{
    LOG_INF("Respiration monitor start (fs=%.1fHz, window=%ds)", (double)SAMPLE_RATE_HZ,
WINDOW_SEC);

    const struct device *dev = DEVICE_DT_GET_ANY(st_lsm6dso);
    if (!device_is_ready(dev)) {
        LOG_ERR("LSM6DSO device not ready (check overlay / wiring).");
        return;
    }
```

```
/* Configure sensor: ODR=52Hz, FS=2g (if supported by driver) */
struct sensor_value odr = { .val1 = 52, .val2 = 0 };
(void)sensor_attr_set(dev, SENSOR_CHAN_ACCEL_XYZ, SENSOR_ATTR_SAMPLING_FREQUENCY, &odr);

struct sensor_value fs = { .val1 = 2, .val2 = 0 };
(void)sensor_attr_set(dev, SENSOR_CHAN_ACCEL_XYZ, SENSOR_ATTR_FULL_SCALE, &fs);

/* Init filters (copy template coefficients into per-axis instances with independent state) */
copy_sos_template(x_sos, g_bp_sos, 4);
copy_sos_template(y_sos, g_bp_sos, 4);
copy_sos_template(z_sos, g_bp_sos, 4);
sos_filter_init(&fx, x_sos, 4);
sos_filter_init(&fy, y_sos, 4);
sos_filter_init(&fz, z_sos, 4);

resp_analyzer_init(&analyzer);

int64_t next_us = k_uptime_get() * 1000;
const int64_t period_us = (int64_t)(1000000.0f / SAMPLE_RATE_HZ);

while (1) {
    next_us += period_us;

    /* Fetch a sample */
    if (sensor_sample_fetch(dev) == 0) {
        struct sensor_value a[3];
        if (sensor_channel_get(dev, SENSOR_CHAN_ACCEL_XYZ, a) == 0) {
            /* Convert to float (m/s^2) */
            float ax = sensor_val_to_f(&a[0]);
            float ay = sensor_val_to_f(&a[1]);
            float az = sensor_val_to_f(&a[2]);

            /* Band-pass filter */
            vec3f_t f;
            f.x = sos_filter_process(&fx, ax);
            f.y = sos_filter_process(&fy, ay);
            f.z = sos_filter_process(&fz, az);
```

```
            ring_push(f);
            samples_since_eval++;
        }
    }

    /* Run analysis every STEP_SEC after buffer is full */
    if (ring_filled == WINDOW_SIZE && samples_since_eval >= STEP_SAMPLES) {
        samples_since_eval = 0;

        ring_to_window(win_xyz);

        /* Center and PCA */
        vec3f_t mean;
        center_vec3f(win_xyz, WINDOW_SIZE, &mean);

        vec3f_t dir;
        pca_principal_dir(win_xyz, WINDOW_SIZE, &dir);
        project_onto(win_xyz, WINDOW_SIZE, &dir, fused);

        resp_result_t res = resp_analyze(&analyzer, fused, WINDOW_SIZE, SAMPLE_RATE_HZ,
(float)STEP_SEC);

        LOG_INF("RR=%.1f bpm | RMS=%.4f | dir=[%.2f %.2f %.2f] | %s",
                (double)res.rate_bpm, (double)res.amplitude_rms,
                (double)dir.x, (double)dir.y, (double)dir.z,
                status_str(res.status));
    }

    /* Pace loop */
    int64_t now_us = k_uptime_get() * 1000;
    int64_t sleep_us = next_us - now_us;
    if (sleep_us > 0) {
        k_sleep(K_USEC((int32_t)sleep_us));
    } else {
        /* overrun, resync */
        next_us = now_us;
    }
}
```

```
}
```

Filter_sos.c

```
#include "filter_sos.h"

/* This module implements a stable digital IIR band-pass filter using second-order sections (SOS).
 Each SOS is a biquad filter, and multiple sections are cascaded for numerical stability.
 The filter removes gravity drift and high-frequency noise while preserving breathing motion. */

void sos_filter_init(sos_filter_t *f, sos_section_t *sections, uint8_t n_sections)
{
    f->sec = sections;
    f->n_sec = n_sections;
    for (uint8_t i = 0; i < n_sections; i++) {
        f->sec[i].z1 = 0.0f;
        f->sec[i].z2 = 0.0f;
    }
}

float sos_filter_process(sos_filter_t *f, float x)
{
    /* Direct Form II Transposed cascade */
    for (uint8_t i = 0; i < f->n_sec; i++) {
        sos_section_t *s = &f->sec[i];
        float y = (s->b0 * x) + s->z1;
        s->z1 = (s->b1 * x) - (s->a1 * y) + s->z2;
        s->z2 = (s->b2 * x) - (s->a2 * y);
        x = y;
    }
    return x;
}

/*
 * 4th-order Butterworth band-pass (0.1–0.8 Hz), fs=52 Hz
 * Coefficients generated with scipy.signal.butter(..., output='sos', fs=52).
 */
sos_section_t g_bp_sos[4] = {
```

```
    { .b0 = 2.870802612e-06f, .b1 = 5.741605225e-06f, .b2 = 2.870802612e-06f, .a1 = -1.872110180e+00f, .a2 =
8.776343808e-01f, .z1 = 0.0f, .z2 = 0.0f },
    { .b0 = 1.000000000e+00f, .b1 = 2.000000000e+00f, .b2 = 1.000000000e+00f, .a1 = -1.935774218e+00f, .a2 =
9.444284481e-01f, .z1 = 0.0f, .z2 = 0.0f },
    { .b0 = 1.000000000e+00f, .b1 = -2.000000000e+00f, .b2 = 1.000000000e+00f, .a1 = -1.974210841e+00f, .a2
= 9.744397348e-01f, .z1 = 0.0f, .z2 = 0.0f },
    { .b0 = 1.000000000e+00f, .b1 = -2.000000000e+00f, .b2 = 1.000000000e+00f, .a1 = -1.992369733e+00f, .a2
= 9.925223120e-01f, .z1 = 0.0f, .z2 = 0.0f },
};
```

Filter_sos.h

```
#pragma once
#include <stdint.h>

#ifdef __cplusplus
extern "C" {
#endif

/* Single SOS section (a0 assumed = 1) using DF2T */
typedef struct {
    float b0, b1, b2;
    float a1, a2;
    float z1, z2;
} sos_section_t;

typedef struct {
    sos_section_t *sec;
    uint8_t n_sec;
} sos_filter_t;

void sos_filter_init(sos_filter_t *f, sos_section_t *sections, uint8_t n_sections);
float sos_filter_process(sos_filter_t *f, float x);

#ifdef __cplusplus
}
#endif
```

PCA_fusion.c

```
#include "pca_fusion.h"
#include <math.h>

 /* Takes 3-axis accelerometer data

Finds the main direction of motion using PCA (Principal Component Analysis)

Converts 3 signals (x, y, z) into one clean breathing signal */

void center_vec3f(vec3f_t *data, int len, vec3f_t *mean_out)
{
    double sx = 0, sy = 0, sz = 0;
    for (int i = 0; i < len; i++) {
        sx += data[i].x;
        sy += data[i].y;
        sz += data[i].z;
    }
    mean_out->x = (float)(sx / len);
    mean_out->y = (float)(sy / len);
    mean_out->z = (float)(sz / len);

    for (int i = 0; i < len; i++) {
        data[i].x -= mean_out->x;
        data[i].y -= mean_out->y;
        data[i].z -= mean_out->z;
    }
}

static void mat3_mul_vec(const float m[3][3], const vec3f_t *v, vec3f_t *out)
{
    out->x = m[0][0]*v->x + m[0][1]*v->y + m[0][2]*v->z;
    out->y = m[1][0]*v->x + m[1][1]*v->y + m[1][2]*v->z;
    out->z = m[2][0]*v->x + m[2][1]*v->y + m[2][2]*v->z;
}

static float vec_norm(const vec3f_t *v)
```

```
{
    return sqrtf(v->x*v->x + v->y*v->y + v->z*v->z);
}

void pca_principal_dir(const vec3f_t *data, int len, vec3f_t *dir_out)
{
    /* Covariance matrix (3x3), normalized by len. Data is assumed centered. */
    double xx=0, xy=0, xz=0, yy=0, yz=0, zz=0;
    for (int i = 0; i < len; i++) {
        const double x = data[i].x;
        const double y = data[i].y;
        const double z = data[i].z;
        xx += x*x; xy += x*y; xz += x*z;
        yy += y*y; yz += y*z;
        zz += z*z;
    }
    const float invN = 1.0f / (float)len;
    float cov[3][3] = {
        { (float)(xx*invN), (float)(xy*invN), (float)(xz*invN) },
        { (float)(xy*invN), (float)(yy*invN), (float)(yz*invN) },
        { (float)(xz*invN), (float)(yz*invN), (float)(zz*invN) },
    };

    /* Power iteration */
    vec3f_t v = {1.0f, 0.0f, 0.0f};
    for (int iter = 0; iter < 10; iter++) {
        vec3f_t next;
        mat3_mul_vec((const float (*)[3])cov, &v, &next);
        float n = vec_norm(&next);
        if (n < 1e-9f) {
            break;
        }
        v.x = next.x / n;
        v.y = next.y / n;
        v.z = next.z / n;
    }

    *dir_out = v;
```

```
}

void project_onto(const vec3f_t *data, int len, const vec3f_t *dir, float *out)
{
    for (int i = 0; i < len; i++) {
        out[i] = data[i].x*dir->x + data[i].y*dir->y + data[i].z*dir->z;
    }
}
```

PCA_fusion.h

```
#pragma once
#include <stdint.h>

#ifdef __cplusplus
extern "C" {
#endif

typedef struct {
    float x, y, z;
} vec3f_t;

/* Removes mean in-place (centers data). */
void center_vec3f(vec3f_t *data, int len, vec3f_t *mean_out);

/* PCA via power iteration on covariance matrix; returns principal direction (unit vector). */
void pca_principal_dir(const vec3f_t *data, int len, vec3f_t *dir_out);

/* Project onto principal component: out[i] = dot(data[i], dir) */
void project_onto(const vec3f_t *data, int len, const vec3f_t *dir, float *out);

#ifdef __cplusplus
}
#endif
```

Respiration.c

```
#include "respiration.h"
```

```c
#include <math.h>

void resp_analyzer_init(resp_analyzer_t *a)
{
    a->low_energy_seconds = 0.0f;
}

static float clampf(float x, float lo, float hi)
{
    if (x < lo) return lo;
    if (x > hi) return hi;
    return x;
}

resp_result_t resp_analyze(resp_analyzer_t *a, const float *signal, int len,
                           float fs_hz, float step_sec)
{
    resp_result_t r = {0};

    /* --- RMS amplitude (apnea proxy) --- */
    float sum_sq = 0.0f;
    float max_v = -1e9f, min_v = 1e9f;
    for (int i = 0; i < len; i++) {
        const float s = signal[i];
        sum_sq += s*s;
        if (s > max_v) max_v = s;
        if (s < min_v) min_v = s;
    }
    r.amplitude_rms = sqrtf(sum_sq / (float)len);

    /*
     * Zephyr accel is typically m/s^2. The report's example threshold is ~0.002 g RMS.
     * 0.002 g ~= 0.0196 m/s^2. Start here and tune per mounting.
     */
    const float apnea_rms_th = 0.0196f;

    if (r.amplitude_rms < apnea_rms_th) {
        a->low_energy_seconds += step_sec;
```

```
} else {
    a->low_energy_seconds = 0.0f;
}
if (a->low_energy_seconds >= 10.0f) {
    r.status = RESP_STATUS_APNEA;
    r.rate_bpm = 0.0f;
    return r;
}

/* --- Adaptive hysteresis zero-crossing (counts half cycles) --- */
float ptp = max_v - min_v;
float hysteresis = clampf(ptp * 0.2f, 0.001f, 1e6f);

int crossings = 0;
bool pos = (signal[0] >= 0.0f);
for (int i = 1; i < len; i++) {
    float s = signal[i];
    if (pos && (s < -hysteresis)) {
        pos = false;
        crossings++;
    } else if (!pos && (s > hysteresis)) {
        pos = true;
        crossings++;
    }
}

float duration_min = ((float)len / fs_hz) / 60.0f;
if (duration_min > 1e-6f) {
    r.rate_bpm = ((float)crossings / 2.0f) / duration_min;
}

/* --- Classification --- */
if (r.rate_bpm > 25.0f) {
    r.status = RESP_STATUS_TACHYPNEA;
} else if (r.rate_bpm < 10.0f) {
    r.status = RESP_STATUS_BRADYPNEA;
} else {
    r.status = RESP_STATUS_NORMAL;
```

```
    }

    return r;
}
```

Respiration.h

```
#pragma once
#include <stdint.h>
#include <stdbool.h>

#ifdef __cplusplus
extern "C" {
#endif

typedef enum {
    RESP_STATUS_NORMAL = 0,
    RESP_STATUS_APNEA,
    RESP_STATUS_TACHYPNEA,
    RESP_STATUS_BRADYPNEA,
    RESP_STATUS_CHEYNE_STOKES,
    RESP_STATUS_MOTION_ARTIFACT,
} resp_status_t;

typedef struct {
    float rate_bpm;
    float amplitude_rms;
    resp_status_t status;
} resp_result_t;

typedef struct {
    /* persistent state for apnea timing etc */
    float low_energy_seconds;
} resp_analyzer_t;

void resp_analyzer_init(resp_analyzer_t *a);

/*
```

```
 * Analyze a window of fused respiration signal.
 * - fs_hz: sample rate
 * - window_len: number of samples in window
 * - step_sec: how often this function is called (e.g. 1.0 for 1s update); used for apnea timing
 */
resp_result_t resp_analyze(resp_analyzer_t *a, const float *signal, int window_len,
                           float fs_hz, float step_sec);

#ifdef __cplusplus
}
#endif
```