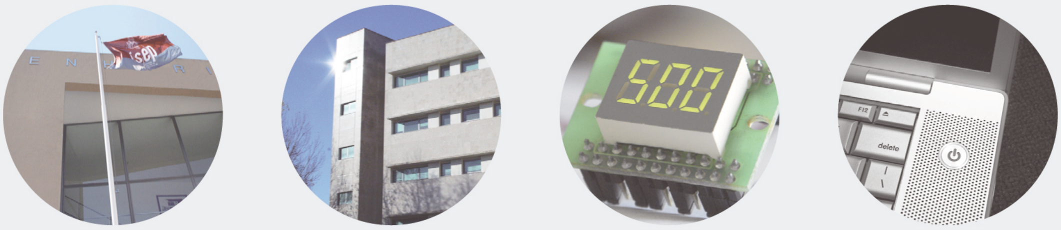
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



Dispositivo de Deteção do Bruxismo do Sono

EDUARDO LUÍS NETO SANTOS
novembro de 2020

POLITÉCNICO
DO PORTO

Sleep Bruxism Detection Device

Mestrado em Engenharia Eletrotécnica e de Computadores

Eduardo Luís Neto Santos (1150646)

Orientadores
Lino Manuel Baptista Figueiredo (LBF)
António José Matos De Meireles (AEM)

Ano Letivo: 2019-2020

Instituto Superior de Engenharia do Porto
Departamento de Engenharia Eletrotécnica
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Abstract

This thesis aims to explore and, ultimately, develop a system capable of monitoring physiological signals to detect bruxism events. Bruxism is a disorder characterized by the habit of pressing and grinding the teeth. These events can either occur during the day (Awake Bruxism) or during the night (Sleep Bruxism). Studies suggest that 20% of the adult population suffer from Awake Bruxism, and 8-16% from Sleep Bruxism.

The consequences of this disorder are several, ranging from tooth wear, dental fractures, or abfraction, resulting in headaches, or facial myalgia. This dissertation focuses on the Sleep Bruxism type since it's harder to detect and treat.

First, a study about the evolution of technology in healthcare is carried out, fundamentally about how it was introduced and how did it get to the point it is now. The topic of wearable devices is also explored, in the sense that it's where the market is going and how these devices can transform healthcare. Then, the study converges on the devices developed especially for bruxism, namely which devices, and what type of techniques are used.

Subsequently, the general concept for the system is elaborated, exploring several options both in terms of devices and physiological data to be parameterized. However, some restrictions exist for the construction of the system. For the construction of an intraoral system, the device has to be of small dimensions and with low energy consumption.

With these constraints, the system has implemented an Inertial Measurement Unit to estimate the orientation of the patient's sleeping position, and force sensors to measure the force exerted between the teeth. For compactness, a System-on-Chip is used, since it includes an ARM Cortex M4 processor, several peripherals, and an RF transceiver in one package.

The system is not only responsible for the data acquisition, but also the data transmission. This is accomplished by using Bluetooth Low Energy, which is one of the most common protocols for low-power devices. Customized service is

developed for this purpose, consisting of three different characteristics: the force characteristic, the accelerometer characteristic, and the gyroscope characteristic. The reason is for maximizing efficiency.

The last step was to develop the prototype, testing its functionalities and try to project next iterations of the prototype.

Keywords

Bruxism, Force, Accelerometer, Gyroscope, Sleep

Contents

List of Figures

List of Tables

Glossary

Abbreviation	Description
AB	*Awake Bruxism*
ADC	*Analog-to-Digital Conversion*
AFH	*Adaptive Frequency Hopping*
AI	*Artificial Intelligence*
ALE	*Average Life Expectancy*
ATT	*Attribute Protocol*
BAC	*Blood Alcohol Content*
BAN	*Body Area Network*
BLE	*Bluetooth Low Energy*
BLS	*Bilateral Stimulation System*
BP	*Blood Pressure*
BPF	*Band-Pass Filter*
BR	*Basic Rate*
BSF	*Band-Stop Filter*
C	*Capacitor*
CCCD	*Client Characteristic Configuration Descriptor*
CI	*Connection Interval*
CSS	*Chirp Spread Spectrum*
CT	*Computed Tomography*
DFT	*Discrete Fourier Transform*
DMP	*Digital Motion Processor*
DSP	*Digital Signal Processing*
DWT	*Data Watchpoint and Trace*
EasyDMA	*Easy Direct Memory Access*
ECG	*Electrocardiogram*
ECMO	*Extracorporeal Membrane Oxygenation*
EDR	*Enhanced Data Rate*
EEBMC	*Embedded Microprocessor Benchmark Consortium*
EEG	*Electroencephalography*
EHR	*Electronic Health Records*
EMG	*Electromyogram*

Abbreviation	Description
EOG	*Electrooculogram*
ESB	*Enhanced ShockBurst*
ETM	*Embedded Trace Macrocell*
FCC	*Federal Communications Commission*
FDA	*Food and Drug Admnistration*
FES	*Functional Electric Stimulation*
FFT	*Fast Fourier Transform*
FIR	*Finite Impulse Response*
FPU	*Floating Point Unit*
FSR	*Force Sensitive Resistor*
GAP	*Generic Access Profile*
GATT	*Generic Attribute Profile*
GFSK	*Gaussian Frequency-Shift Keying*
GPU	*Graphics Processing Unit*
HAL	*Hardware Abstraction Layer*
HMM	*Hidden Markov Model*
HPF	*High-Pass Filter*
HR	*Heart Rate*
HRV	*Heart Rate Variability*
IC	*Integrated Circuit*
IFA	*Inverted F Antenna*
IIP	*Implanted Insulin Pump*
IIR	*Infinite Impulse Response*
IME	*Ingestible Event Marker*
IMU	*Inertial Measurement Unit*
IT	*Information Technology*
ITM	*Instrumentation Trace Macrocell*
KMU	*Key Management Unit*
KNN	*K-Nearest Neighbor*
L	*Inductor*
LASER	*Light Amplification by the Stimulated Emission of Radiation*
LDO	*Low-Dropout*
LFCLK	*Low-Frequency Clocl*
Li-ion	*Lithium-ion*
Li-Po	*Lithium Polymer*
LLR	*Left Lateral Recumbent*
LPF	*Low-Pass Filter*
LR-PAN	*Low Rate PAN*
MAD	*Mandibular Advancement Device*
MD	*Mahalanobis Distance*
MEMG	*Masseter Electromyographic*
MEMS	*Micro-Electromechanical Systems*
MICS	*Medical Implant Communication System*
ML	*Machine Learning*

Abbreviation	Description
MRI	*Magnetic Resonance Imaging*
NFC	*Near Field Communication*
NiMH	*Nickel-Metal Hydride*
OpAmp	*Operational Amplifier*
PFA	*Prime-Factor Algorithm*
PGM	*Probabilistic Graphical Model*
PPCP	*Peripheral Preferred Connection Parameters*
PPG	*Photoplethysmogram*
PPI	*Programmable Peripheral Interconnect*
PSG	*Polysomnograph*
PTT	*Pulse wave Transit Time*
QDEC	*Quadrature Decoder*
R	*Resistor*
RIP	*Respiratory Inductance Plethysmography*
RLR	*Right Lateral Recumbent*
RMMA	*Rhythmic Masticatory Muscle Activity*
RSSI	*Received Signal Strength Indicator*
RTC	*Real-Time Clock*
SAADC	*Successive-Approximation ADC*
SB	*Sleep Bruxism*
SCL	*Serial Clock*
SDA	*Serial Data*
SoC	*System on Chip*
SVM	*Support Vector Machine*
SWD	*Serial Wire Debug*
TDMA	*Time Division Multiple Access*
TMJ	*Temporomandibular Joint*
TWI	*Two-Wire Interface*
ULP	*Ultra Low-Power*
UUID	*Universally Unique Identifier*
WBAN	*Wireless Body Area Network*
Wh	*Watt-hour*
WHD	*Wearable Health Device*
WHO	*World Health Organization*
WLAN	*Wireless Local Area Network*
Wi-Fi	*Wireless Fidelity*
WPAN	*Wireless Personal Area Network*

Chapter 1

Introduction

In this chapter, a brief introduction to the project is made, as well as a presentation of its schedule. The goals of the work and a small context will also be mentioned.

1.1 Contextualization

With the evolution of technology, new ways of thinking and implementing different systems have undergone a revolution. One of the areas where this revolution has been affecting is healthcare. The possibility of developing systems capable of monitoring diseases, disorders, or physiological signals, in a continuous and non-invasive way for the patient, allows the collection of information, previously difficult to obtain, which may lead to treatment, detection, prevention, and, ultimately, saving lives.

Bruxism, a concept first presented by Marie Pietkiewicz in 1907, is a disorder characterized by the clenching and grinding of teeth in an involuntary or semi-voluntary manner. The effects of bruxism are several, including the wear on the teeth, leading to fractures, and abfraction, headaches, hypersensitivity, and facial myalgia. There are two types of bruxism, including Sleep Bruxism, where the events occur during sleep, and Awake Bruxism, where the events occur while the person is awake.

Although the cause of bruxism was not yet defined, its occurrence has been increasing, especially in the younger population. Additionally, studies suggest that 20 % of the adult population suffers from awake bruxism, and 8-16 % suffer from sleep bruxism. This thesis aims to develop a system capable of monitoring some physiological signals to detect events of bruxism, preventing some of the consequences enumerated [31].

1.2 Goals

The main goal of this thesis is to develop a system that can monitor data to determine the activity of a person while sleeping, sending that information to an external device. To achieve that goal, we have defined a few guidelines:

- Study the evolution of technology in healthcare
- Approach the subject of wearable devices
- Review some bruxism ambulatory systems already developed
- Understand some of the technologies used on wearable devices
- Define the system's architecture
- Choose the various components
- Test, analyze, and learn the functioning of these components
- Develop code for the SAADC peripheral, for retrieving the force sensors data
- Develop code for interacting with the MPU-6050
- Develop a BLE service
- Integrate and test the different parts
- Develop the prototype

1.3 Timeline

According to the established goals, the crucial points to the accomplishment of this project are scheduled and presented in the Figure 1.1.

1.4 Structure

This report contains seven chapters. The present chapter briefly introduces the subject of the thesis, describing some of the objectives established and presenting the timeline for the project.

In the second chapter, we explore bruxism disorder, which is the main reason for the project, and we dive into the evolution of technology in healthcare. Furthermore, we approach the concept of wearable devices and how these systems can affect healthcare. Next, we present some bruxism ambulatory systems already developed, as well as some of their particularities. Finally, to better

Table 1.1: Project Schedule

	Maio	Junho	Julho	Agosto	Setembro	Outubro
Report Writing	■	■	■	■	■	■
Technology in Healthcare	■	■				
Wearable Devices	■	■				
Bruxism Ambulatory Systems	■	■				
Technologies Review		■				
Structure of the System		■	■			
Composition of the System		■	■			
nRF52832 First Look			■	■		
MPU-6050 Study				■	■	
SAADC Related Work				■	■	
BLE Service Development				■	■	
Integration of the System				■	■	■
Prototype Development					■	■
Tests			■	■	■	■

understand the requirements for our system, some of the technologies used on these types of devices are studied.

In the third chapter, we idealize our system, involving all the components used, how they can communicate with each other, its construction, and how could it be implemented.

In the fourth chapter, we explore what kind of devices are used to develop the project. The reasons why those devices were chosen and their characteristics are enumerated. The protocol used for data transmission is also analyzed.

In the fifth chapter, we start to approach the development of the system. The IMU and the estimation of the patient's sleep position, the SAADC peripheral, and the analog data acquisition from the force sensors, the BLE, and the service created for data transmission, the application logic, that is, how all the components merge, and the prototype development, are the main topics of this chapter.

In the sixth chapter, we examine our system and validate some of the implementations approached previously.

Finally, in the last chapter, we reflect on the main conclusions of this project, including posterior improvements.

Chapter 2

State of the Art

Bruxism is a disorder characterized by the habit of pressing and grinding the teeth, whether or not they produce sounds. The implications and what techniques are used in the diagnosis of this condition will be detailed.

However, to fully understand the problem and the possible solution, an evolutionary review of technology in medicine will be demonstrated and how those to correlate.

With continuous technological advances, new ways of measuring the various biological signals that can help, often, to detect anomalies and even treat diseases more efficiently have emerged. Given that, a new trend is currently underway, which is the use of Wearable Health Devices (WHD), which will also be addressed.

Finally, some ambulatory systems for the registration of bruxism events will be analyzed and a discussion on the construction of the architecture of a possible device will be carried out.

2.1 Bruxism: A Review

The concept of bruxism was first presented by Marie Pietkiewicz in 1907, described as "*la bruxomanie*". According to a study by Shilpa Shetty et al. [32], it never attracted much interest until 1996. Since then many articles and studies have been carried out on this subject.

Bruxism is a disorder characterized by the clenching and grinding of teeth, in an involuntary or semi-voluntary manner, with the application of excessive forces on the masticatory muscles, causing wear on the teeth that can affect their integrity and compromise oral health.

Bruxism can be characterized by occurring during sleep, Sleep Bruxism (SB), or during the day, Awake Bruxism (AB). AB is characterized by a semi-voluntary activity of the jaw, with the clenching of teeth during daily life and where there is generally no creaking, being related to a tic, habit, or a reaction to stress. SB is an involuntary activity where the pressure of the teeth occurs, usually with the existence of sounds, during sleep.

Studies suggest that this disorder has an incidence of about 20% (AB) of the adult population, and between 8-16% (SB). Concerning AB, there is a greater predominance in females, and in SB there is no differentiation between genders. This disorder is also expected to be increasingly common in the younger population.

2.1.1 Aetiology

Initially, bruxism was associated with morphological factors such as malocclusion or dental arch form. However, studies have not shown a clear association between these factors and bruxism. However, more potential causes have been pointed out for bruxism, such as pathophysiological and psychological factors [33].

Pathophysiological factors may include sleep disturbances/arousals, biological causes, consumption of certain drugs, smoking (nicotine), brain chemistry, among others. One of the strongest hypotheses is the correlation between sleep bruxism and micro-arousals. Rhythmic masticatory muscle activity (RMMA) is directly associated with sleep micro-arousals, with RMMA being observed during SB. SB episodes have been shown to occur during brain activation and increased heart rate. There is also the hypothesis that bruxism is associated with disorders in the central neurotransmitter system, due to an imbalance in the pathways of the basal ganglion. Some indications indicate an increase in bruxism activity related to the use of dopamine precursors, serotonin reuptake inhibitors, or amphetamine. Chronic gastrointestinal diseases may have some influence on the onset of bruxism events [34, 35, 33, 36].

Psychological factors are associated with stress, anxiety, depression, competitiveness, etc. Some studies support the theory that there is a relationship between bruxism and a person's psychological state. Patients with SB presented elevated levels of urinary catecholamine, psychological stress, and salivary cortisol. It was also observed that more competitive, anxious, and nervous patients were more likely to exhibit bruxism [36, 37].

2.1.2 Consequences

Bruxism can affect bones, muscles, and nerves. During SB, the magnitude of the bite can exceed the amplitude of the maximum voluntary bite force. Since this disorder directly affects the teeth, an individual may suffer from tooth wear, dental fractures, or abfraction. Another cause may be headaches, and it is estimated that bruxers are more likely to suffer from headaches. Since muscles are an active part of this disorder, bruxers can experience facial myalgia. Due to irritation of the nerves surrounding the teeth, it can cause hypersensitivity. With the existence of bites, it is likely that there are lesions on the lips, cheeks, and tongue. In the long run, it may contribute to temporomandibular joint (TMJ) syndrome.

2.1.3 Assessment

The path to the detection of bruxism is not always linear, since it is not a disorder that is particularly known and whose symptoms can vary from person to person. Also, bruxism is a condition that can fluctuate in the level of incidence, leading to some negligence on the part of the patient.

The most used form for the identification of AB and SB is medical observations, which can be initiated by patients due to some complaints either by family members (noise during SB), by pain/discomfort, or by awareness of clenching. These observations are followed by questionnaires and orofacial examinations, such as tooth wear, tooth indentation, fractures, and masseter muscle hypertrophy. As mentioned, all of these observations are relative and can only serve to have a certain degree of certainty about the presence of bruxism [38, 39, 40].

To definitively diagnose bruxism, physiological measurements will be necessary. Polysomnography (PSG) is the study of sleep, measuring multiple parameters for the diagnosis of any condition. PSG can be performed in a laboratory setting, monitoring brain activity, electroencephalography (EEG), masticatory muscle activity, electromyogram (EMG), cardiac activity, electrocardiogram (ECG), respiration, and eye activity, electrooculogram (EOG). This option is the most effective, since it is carried out in a controlled environment, with access to detailed physiological data and the possibility of confirmation through audio and video, but it is a financially unappealing option [38, 39, 40].

Ambulatory systems have been developed to perform PSG. These systems have the advantage of being low cost and carried out in a natural environment for the patient, however, there is no monitoring for so many physiological signals and their accuracy may be lower. The most common systems are the measurement of muscle activity and, eventually, cardiac activity, as well as splint with a force sensor [38, 39, 40].

2.1.4 Management

Unfortunately, in the case of bruxism, it refers to management instead of a treatment since there is still no treatment or cure for this condition. Some strategies have been developed to manage SB, including pharmacological, psychological, and dental.

Some drugs have shown good results in the management of BS. Clonidine, L-dopa, and clonazepam were some of them, the latter leading to a reduction of $42 \pm 15\%$. However, the use of drugs can lead to addiction as well as having psychological side effects. In more extreme cases, it is possible to administer botulinum toxin (BTX type A), which is an inhibitor of acetylcholine release at the neuromuscular junction.

In terms of psychological approaches, their effectiveness is not well established. One of the most used techniques is biofeedback, which is the principle that a given behavior can be changed if it is corrected. This strategy may have better results in the case of AB, as an individual can learn to control his impulses, but it can also be used in SP using vibratory, electrical, and auditory stimulation.

The most common technique for managing bruxism is the use of occlusal splints. These objects serve to protect the teeth and can help to reduce the events of bruxism. They are normally used in the upper jaw and are composed of acrylic resin. There are also mandibular advancement devices (MAD) that have shown good results, being used to treat snoring and OSA.

2.2 Technological Evolution in Healthcare

Technology has been one of the most important factors in the evolution of society as we know it. It has influenced virtually every aspect of our lives. The way we move, the way we communicate, the way we work... Essentially, the way we live.

One of the areas that suffered the most impact was, definitely, medicine and healthcare. This impact translates into an increase in average life expectancy (ALE) as presented in the Fig 2.1, standing near to 30 years in the 18th century, currently reaching a maximum of 88 years in some countries [1]. Besides, there is a significant difference between the more developed and less developed countries, which indicates the relevance that health care has in the average life expectancy of a person at birth.

So, this increase can be explained by innovations in terms of devices, medicines, vaccines, procedures, and systems, which is directly related to the essence of health technology as described by the World Health Organization (WHO) [41]. However, it will be important to detail some of these innovations to give a perspective of their real importance and how far this field has come.

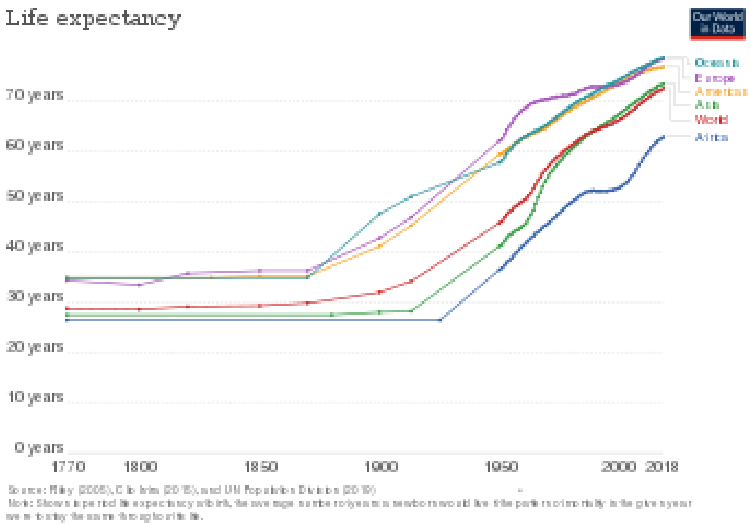


Figure 2.1: Life expectancy throughout history [1]

Surgery Techonology

One of the goals of medicine is to minimize the invasiveness of surgeries, thus leading to a decrease in recovery time and possible complications. Techniques have been developed to perform this type of surgery, whether with robotic assistance or not.

For example, in robotic-assisted surgery, the surgeon has access to controllers that allow him to maneuver robotic arms. With the visualization of 3-dimensional images in high definition, it is able to perform very precise movements [42, 43].

With non-robotic assistance, also known as endoscopic surgery, a thin and flexible tube is used, composed of a camera and channels for the use of surgical medical equipment. That tube is then inserted through a small incision or an orifice and allows the surgeon to perform surgical procedures while viewing the internal organs [43].

There is also Light Amplification by the Stimulated Emission of Radiation (LASER) surgery. This type of procedure, which utilises laser scalpels, is widely used because it is accurate and reduces the impact on the surrounding tissue. It is used for cancer treatments, vision-related surgeries, cleaning clogged arteries, and more [44].

Medical Imaging

Another field of interest is medical imaging, which serves to treat and/or diagnose diseases or conditions by scanning the human body, giving doctors the chance to make adequate decisions and accurate diagnoses. The most common devices for performing this type of examinations are X-rays, computed tomography (CT) and magnetic resonance imaging (MRI) [45].

Both X-rays and CT are a form of radiation that, when crossing the human body, can be blocked or not depending on the type of tissue that comes across. This radiation is then captured, revealing an image of the body [46].

MRI, unlike those previously mentioned, does not use the damaging ionizing radiation of X-rays. Instead, it uses magnets that produce a magnetic field, forcing protons in the body to align with the field. Using radio waves, it is possible to measure the energy dissipated by the protons, through MRI sensors, switching the electromagnetic field generated by these waves on and off. Depending on the type of tissue, this dissipated energy will be different, allowing the image of the human body to be formed [47]. In Fig 2.2 is described all electric and electronic components involved in a MRI machine.

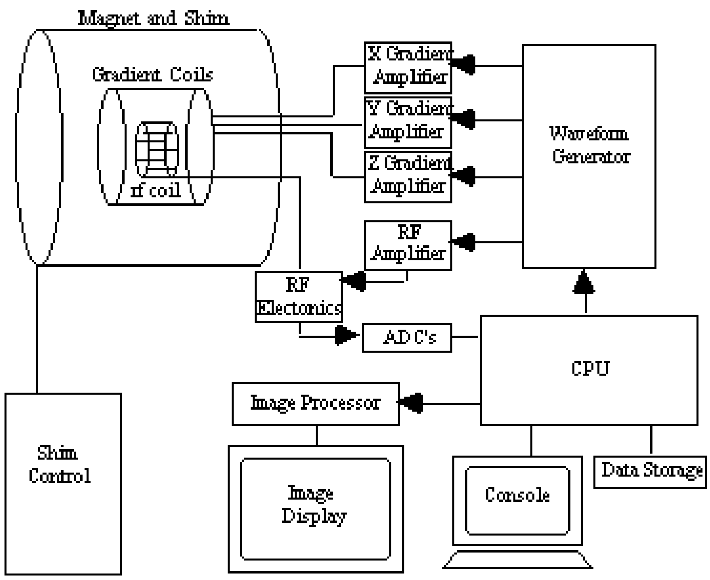


Figure 2.2: Schematic diagram of MRI machine [2]

Life Support

There are machines capable of literally extending a person's life span as is the case with extracorporeal membrane oxygenation (ECMO). It is a form of life support that replaces the heart and lungs allowing their recovery. The ECMO contains an artificial external circulator that transports the patient's venous blood through an oxygenator. This allows for the removal of carbon dioxide and the oxygenation of the blood, introducing it again into the patient's body using a pump [48]. The inner workings of the system are presented in Fig 2.3.

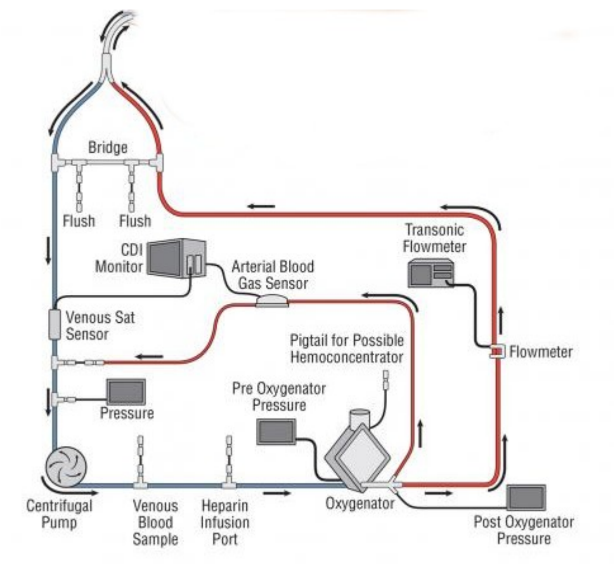


Figure 2.3: ECMO's diagram [3]

Health Information Technology

As for health information technology (IT), there are electronic health records (EHR). These records document the entire medical history of a patient, medication they take, treatment plans, immunizations, allergies, tests, among others. Thus, any doctor who is in charge of a patient may be able to make the best decisions based on his or her history and not be at risk of omission on the part of the patient [49].

Big Data and Machine Learning (ML) are being explored, seeking to improve diagnostics, prognosis and drug development. These two technologies are interconnected since it is necessary to have large amounts of information for these learning algorithms to be applied. The amount of information available about the medical history of patients, tests, treatments, as already mentioned in the EHR, is extremely important for this reason. ML is normally used by cardiologists to recognize electrocardiogram (ECG) patterns, diagnosing an existing condition. Besides, it can also be used to estimate risks, that is, to categorize patients in terms of risk, using ECG data [50]. It can also be used to detect pulmonary nodules using an X-ray [51]. In terms of drug development, ML has been used to identify new antibacterial agents with high accuracy [52].

Complementing what has been said about robotic-assisted surgery, there is already the possibility of doing this type of surgery remotely, that is, without the surgeon being present at the site. One example is given by Mehran Anvari who can operate, from a console at St Joseph's Hospital in Hamilton, Canada, a patient 400 km away [53].

2.3 Wearable Health Devices

The term Wearable Health Device may seem recent, however, it has been around for a long time. A WHD consists of the continuous monitoring and control of vital signs, whether in everyday life or in medical environment. The purpose of these devices is to acquire information in a non-intrusive and comfortable way for the patient.

The fact that this technology allows the continuous monitoring of vital signs, will generate large amounts of data that can have a great impact on diagnostics, prognosis, and treatment. Also, as they are personal devices, they can serve to raise health awareness, giving people an active role in their health status. With the expected increase in population, it will be important to carry out efficient resource management, making use of remote monitoring and teleconsultations. However, these WHDs will have to be small and compact, low power consumption, comfortable, inexpensive, safe, and reliable and ergonomic, encouraging their use by patients.

The application of these devices can vary, from monitoring mental status, sports medicine, weight control, chronic lung diseases and, probably the most expressive, monitoring of physical activity. One of the reasons for the recent buzz around theses types of devices is the introduction of low-power and compact sensors, based on micro-electromechanical systems (MEMS)[1], leading to compact and more affordable devices and expanding the possible applications [6, 54, 55].

2.3.1 Market Projections

With the increase in awareness concerning health, people seek solutions that allow them to monitor different physiological signals, taking proactive rather than reactive approaches and even less invasive and increasingly autonomous devices. Also, health care institutions seek to increase the amount of information available about the patient to make appropriate decisions.

This demand led to the growth of this market and the adoption of these connected devices. A study published in 2019 shows that the global WHD market was valued at 24,571.8 million (USD) in 2018. The study suggests that it is expected to reach around 140,000 million (USD) in 2026 [4].

[1] **MEMS** are systems that can be defined as microscopic and sophisticated devices, consisting of electrical and mechanical components, which are used to interact and act in controlled environments.

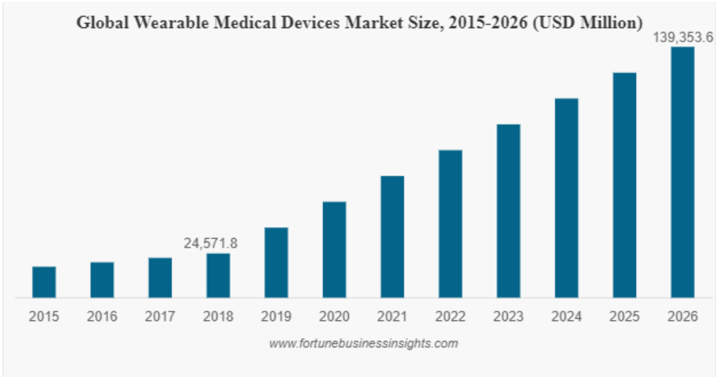


Figure 2.4: Global WHD Market Size [4]

Companies such as Apple, Fitbit, Samsung, and Philips Healthcare are those with the largest presence in the market. The dimension of these companies also gives an idea of the importance of this market and its potential [4].

The study carried out the market segmentation by product, including diagnostic and monitoring devices, as well as therapeutic ones. Diagnostic and monitoring devices include fitness trackers, smartwatches, biosensors, and hold most of the market. Therapeutic devices such as defibrillators, pacemakers, drug delivery devices fill the rest of the market [4].

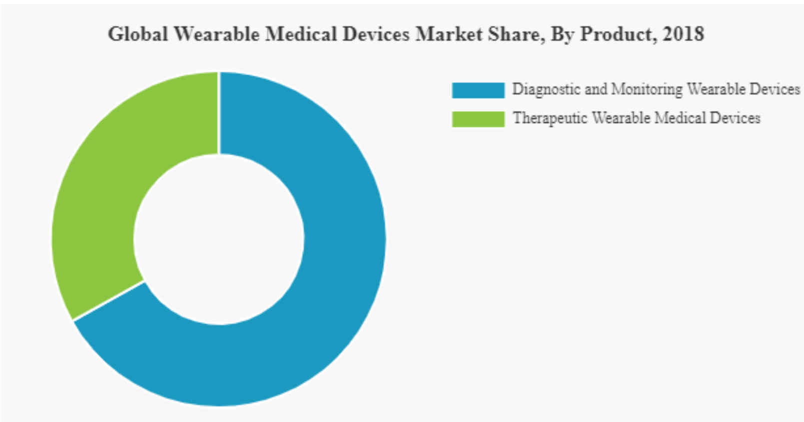


Figure 2.5: Global WHD Segmentation [4]

2.3.2 System Architecture

The architecture of a WHD consists of a set of phases. The presence of sensors allows the collection of physiological data from the human body. Subsequently, these signals will undergo some type of conditioning, be it amplification, filtering, and/or analog-to-digital conversion. The data is then sent to a central processing unit, which processes the received data. From this point on, data can either be transmitted wirelessly to a server or smartphone, in the case of real-time monitoring or stored in a storage device for later control as shown in Figure 3.1.

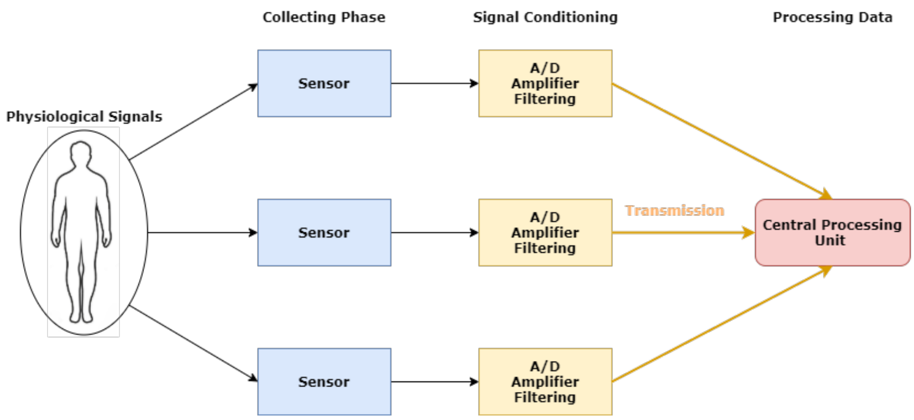


Figure 2.6: WHD architecture

The concept of Body Area Network (BAN) has emerged and it relates to the actual sensor nodes. A BAN can be composed of many sensor nodes connecting to a central unit and can be employed at any part of the human body. The need to increase mobility leads to the creation of Wireless BAN (WBAN) which allows for the wireless interconnection of different sensor nodes. A WBAN sensor is normally composed of a sensor module, which includes the actual sensor and signal conditioning techniques, a memory module, and a radio module.

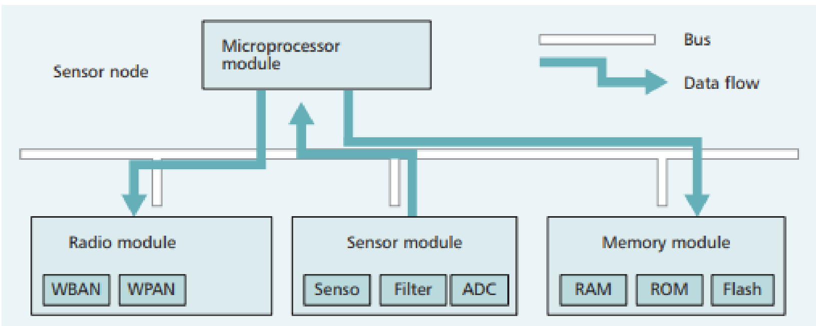


Figure 2.7: General WBAN sensor [5]

Sensors are a central part of this process since they collect raw data, making a connection between the physical world and electronics. Much of the compaction that is done and required in these devices was due to the evolution of the design and manufacturing techniques of these components. The Table 2.1 lists the sensors used to measure certain physiological signals.

Bio-signal	Sensor
Electrocardiogram (ECG)	Skin/Chest electrodes
Temperature	Temperature probe
Respiration rate	Piezoelectric/Piezoresistive sensor
Oxygen saturation	Pulse Oximeter
Perspiration	Galvanic Skin Responde
Electromyogram	Skin electrodes
Movement	Accelometer
Heart sound	Phonocardiograph

Table 2.1: Biosensors and bio-signals [28]

One of the most important aspects of these devices is their portability and flexibility. To fully accomplish that, they need to transmit data wirelessly. Several protocols fall into this category such as the Medical Implant Communication System (MICS), Zigbee, Bluetooth Low Energy (BLE), Wireless Fidelity (Wi-Fi), and, more recently, LoRa.

MICS

MICS was created by the Federal Communications Commission (FCC) in 1999. This specification operates in the radiofrequency range of 402 MHz up to 457 MHz, and, more recently, the spectrum between 2360-2400 MHz was also allocated for this specification. It allows for a short communication range (2 m) and establishes a low maximum transmit power of only 25 mW [56].

Bluetooth Low Energy

Bluetooth Low Energy is a Wireless Personal Area Network (WPAN) technology developed by the Bluetooth Special Interest Group. Uses frequency hopping in the 2.4 GHz band to connect with nearby devices. It features maximum transmission rates of 1 Mbps with an energy consumption between 1-50 mW, depending on the application. Its theoretical maximum range is 100 meters (m), however, the normal range varies between 10-20 m.

Wi-Fi

Wi-Fi is a Wireless Local Area Network (WLAN) technology (IEEE 802.11 standard) developed by the Wi-Fi Alliance. It can operate in the 2.4 GHz band and also in the 5 GHz band. Depending on the type of radio band, the transmission rate will be different (higher at 5 GHz), although it affects its range (higher at 2.4 GHz). This technology differs from other WPANs since it was not developed for the same purposes. Although it is possible to form a WSN, it presents excessive transmission rates for the type and amount of data to be exchanged, weighing on its energy consumption and cost of devices.

ZigBee

The ZigBee protocol is a standard communication protocol, developed by the ZigBee Alliance, for wireless mesh networks, whose belonging devices require low power and a high-reliability factor for their operation. It is based on the IEEE 802.15.4 standard, referred to as Low Rate PAN (LR-PAN), operating in the range between 868 and 868.8 MHz, with only one channel and a transmission rate of up to 20 Kbps (Europe), among the 902 and 928 MHz, where there are 10 channels (2 MHz spacing) and data transmission can reach 40 Kbps (USA), and between 2,400 and 2,44835 GHz, with 16 channels (5 MHz spacing) and a flow which can reach 250 Kbps.

LoRa

LoRa is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology which uses radiofrequency bands like 433 MHz, 868 MHz (Europe), 915 MHz (Australia and North America) and 923 MHz (Asia). LoRa enables long-range transmissions (up to 10 km) with low power consumption.

2.3.3 Data Analysis

The data obtained by monitoring vital signs can be used for different tasks, including fitness, prediction, diagnosis, and anomaly detection. In the Figure 2.8, the arrangement of these tasks is observed in three-dimensional relationships. The first dimension corresponds to where the monitoring is done, being in a clinical environment, or at home/remotely. The second dimension corresponds to who is being monitored, i.e., to a person with or without medical history. The last dimension corresponds to how monitoring is carried out, that is, whether online or offline.

Anomaly Detection

Anomaly detection is a process that consists of data analysis, identifying abnormal patterns concerning the expected behavior. This is extremely important in the medical field since it allows doctors to make adequate decisions. This can be used to identify long inactivity, falls, heart rate (HR), blood pressure, among others. One of the main setbacks of anomaly detection are the false-positives produced by the limited resources on WHD. Due to these limitations, these devices are prone to faulty measures, which can be misleading and affect

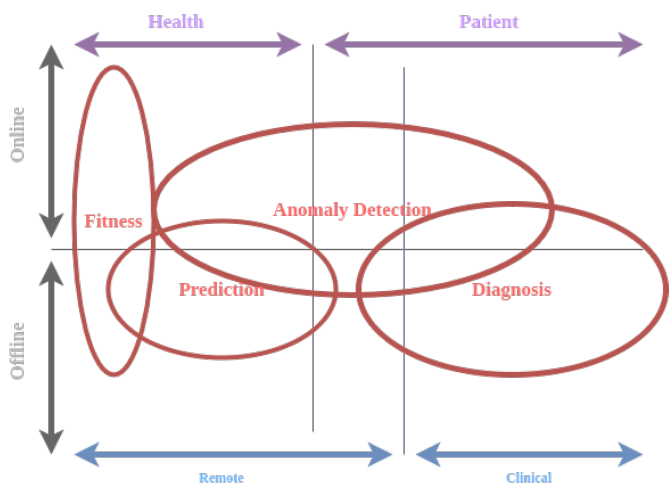


Figure 2.8: Monitoring purposes [6]

medical diagnosis. Many approaches have since been developed to detect these anomalies such as ML algorithms for classification, neural networks, k-nearest neighbor (KNN), Mahalanobis distance (MD), support vector machines (SVM), etc [57, 58].

Prognosis

Prognosis is a method for predicting pathologies or conditions or even to forecast the outcome of a disease. Although it is an area of great interest, it comes with challenges since it requires the modeling of patterns acquired from vital signs. This is known as supervised learning and it involves data mining, model training, and testing. However, it is already been implemented in some applications such as for blood glucose levels, risk estimation of cardiac patients, stress levels, and more [58].

Diagnosis

Diagnosis is a crucial step for physicians because it directly influences the patient's health. Although data from vital signs is important to provide some additional information about the patient's condition, a diagnostic cannot rely purely on this information. A broader view is necessary, so this data is complemented with tests and medical history about the patient. Nevertheless, it can be really helpful to classify the patient's condition, track the disease development, and even in treatment [58].

Fitness

Sedentarism has become a real problem in society and it has negative implications in people's health such as obesity, cardiopulmonary problems, and functional regression. Fitness devices are becoming a worldwide trend because they provide the quantification of exercise, allowing people to set goals and thus increasing their motivation. But these devices are not just for regular use. There are medical applications in which these indicators have real importance like physiotherapy or cardiology. Registering this information allows physicians to have an insight into a patient's behavior.

2.3.4 Monitoring Systems

The continuous tracking of patients' vital signs in real-time produces invaluable information for early diagnosis of diseases, treatment results, remote monitoring of the patient, among others. The popularity of wearable medical devices is steadily increasing due to their extensive applications and promising solutions. Some technologies may not even be particularly recent however, with technological advances, the opportunities for their implementation have grown exponentially.

Hearing Aids

The first wearable health device may date back to the 17th century in the form of ear trumpets. These devices were used to amplify sound and were extremely common at the time. The first electronic hearing aids device was developed by 1870 and it used a carbon transmitter. Nowadays, there are digital hearing aids that can even be implanted.

A current hearing aid consists of 4 components: a power supply, an amplifier, receiver, and a microphone. The source of power supplies power to the system via a battery. The microphone is a transducer that converts the sound signal into an electrical signal. The amplifier is usually composed of transistors that are inserted in an integrated circuit and serve as amplification of the electrical signal coming from the microphone. The receiver receives the amplified signal, transforming it into a sound signal.

Cardiac Implantable Devices

It is possible to consider that one of the most common WHD is the pacemaker. The first definitive electronic pacemaker was implanted by Senning and Elmqvist in Sweden on 8 October 1958. A pacemaker is a device used to control arrhythmias[2], being able to accelerate a low heart rate or to control a higher heart rate, and monitor electrical activity between different parts of the heart.

Over the years these devices have been reduced, reaching sizes of just a few centimeters, thus being minamilly invasive. In 2015, the smallest pacemaker in the world with only 2.5 cm was implanted in Portugal [59, 60, 61].

Diabetes Monitoring

These devices are also important to control and monitoring of diabetes[3]. Previously, the control of blood sugar levels and administration of insulin was done manually, which was extremely uncomfortable for the patient, since it involved the use of syringes and needles.

At this moment, there are already less intrusive ways to carry out this control, as is the case with tethered pumps that are composed of a pump that can be placed in the pocket and that can give indications about the amount of insulin remaining, the blood glucose levels, among others. There are also patch pumps where the pumps are placed on the skin's surface. It contains a remote control, which in itself can be a meter of blood glucose levels. The other two options are rarer, however, they may be the most effective and comfortable for the patient. These are the implanted insulin pump (IIP), which are implanted inside the cup and administer insulin very efficiently, and closed-loop insulin pumps, which work as an artificial pancreas, continuously monitoring glucose levels and administering the amount of insulin autonomously. The latter type was approved by the Food and Drug Administration (FDA) in 2019 in the United States of America [62, 63, 64, 65, 66].

Blood Alcohol Content

Blood alcohol content (BAC) refers to the percentage of ethanol in a person's blood. This measurement can be obtained through blood, urine, sweat, and breathing. The most effective way is through blood analysis but it is not the easiest [67].

The most used test is the breath test, through breathalyzers, being used by police forces. This test measures the alcohol that passes through the lungs, through the blood vessels. Despite being a useful and portable device, it can lead to ineffective results, as it does not take into account the person's physiological characteristics [68].

However, there are already devices capable of measuring BAC through sweat. When alcohol is ingested, ethanol ends up reaching the bloodstream, with a small part being diffused through the skin. PROOF™ created a device with an

---

[2] **Arrhythmias** is a condition that causes irregular heartbeat.

[3] **Diabetes** is a chronic metabolic disease that causes a constant increase in blood glucose due to defects in the normal action of insulin, produced by the pancreas.

enzymatic electrochemical sensor that converts ethanol into an electric current, using an algorithm to estimate BAC [69].

Mental Health

Mental health is a very sensitive issue and, until recently, it was underestimated. In 2017, it was calculated that about 11% of the population suffered from some form of mental disorder. Efforts have been made to minimize the effects of these disorders, but it is not an easy task. Technology can provide ways to anticipate and even help in the treatment of patients, measuring behavioral, physiological, and social signals [70].

Otsuka and Proteus developed a product capable of recording medication intake, establishing communication with the patient and his healthcare provider. A drug is combined with an Ingestible Event Marker (IME) sensor. The goal is to register as much detail as possible the drug ingestion over time of a patient with serious mental illness [71] [72].

Amy Serin developed TouchPoints, a bilateral stimulation system (BLS), which are wireless devices capable of altering the neurological response to stress. At the press of a button, micro-vibrations are used to calm the brain during a stressful situation [73].

Fall Detection

With the increase in ALE, the population is expected to age. According to data from World Population Prospects, in 2050, 16% of the population is expected to be over 65 years old. As people age, a person's motor skills are reduced, leading to a large number of falls that end up disabling them [74].

Systems capable of detecting these falls have been developed to alert a responsible entity to provide support. Most of these systems use accelerometers, usually with 3 axes, which measure the acceleration concerning the X, Y, and Z axes. The readings of these sensors can be used in ML algorithms or together with other sensors such as heart rate variability (HRV) or gyroscope, distinguishing normal daily movements from falls [75].

Blood Pressure

The measurement of blood pressure (BP) is given by the systolic that measures the pressure that the blood exerts in the artery during the contraction of the heart and the diastolic that measures the pressure during its relaxation. The number of people suffering from hypertension, high blood pressure, is estimated to be around 1.13 billion, a condition that can have serious health consequences [76].

Usually, the measurement of these values is made in a clinical environment, however, this measurement is punctual and can produce false results due to the patient's nervousness.

One of the commonly used techniques involves a device, placed on the wrist and capable of automatically inflating, which relates the external pressure to the magnitude of the arterial volume. However, these measurements can have negative effects due to interruptions in blood flow [77].

Another method used is by measuring the pulse wave transit time (PTT). Traditionally, it uses an ECG sensor and a photoplethysmogram (PPG)[4] sensor, measuring the time that a pulse takes between points [77].

2.3.5 Bruxism Ambulatory Systems

Due to the high cost of laboratory polysomnography, there was a need to develop systems capable of monitoring physiological signals that would help diagnose or identify conditions associated with sleep disorders and mandibular parafunction (bruxism included). Although these systems are not completely reliable, nor can they be used to make a definitive diagnosis of any condition, they are useful to get a first indication about the situation.

BiteStrip

The **BiteStrip** is a disposable system that can be used to identify sleep bruxism, with reduced dimensions (7 x 2 cm) and lightweight (4g). It consists of 2 EMG electrodes, an amplifier for the EMG signal, lithium battery, display, and processor [7], which can be seen in Fig 2.9.

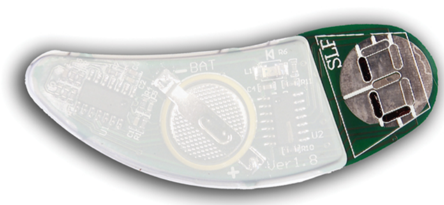


Figure 2.9: BiteStrip model

The system must be applied externally, over the masseter muscle. The user will have to perform 3 or 4 voluntary clenches to establish a baseline for the

[4]**Photoplethysmography** is an optical technique to detect blood volume variations. It is used to perform measurements non-invasively.

study. The **BiteStrip** will record the muscle activity of the masseter, through the EMG electrodes, and these signals will be amplified, digitized, and analyzed in real-time by the processor [7].

Each bruxism burst is counted and stored internally, and a result is presented to the user according to the number of events recorded [78]:

- **0:** less than 40 events.
- **1:** 40-74 events.
- **2:** 75-124 events.
- **3:** more than 125 events.
- **E:** error.

The **ByteStrip**'s threshold is 30% of the maximal voluntary clenches. A bruxism event is recorded if the EMG activity of the masseter is greater than the threshold for at least 0.25 seconds and has a maximum duration of 1 second. If an event lasts 4 seconds, it will be counted as 4 separate events [7].

According to a study by Tamar Shochat et al., There is a positive relationship between masseter electromyographic activity (MEMG) and **BiteStrip** scores. In Figure 2.10, it is possible to observe the recordings of this activity and its comparison with **BiteStrip** [7].

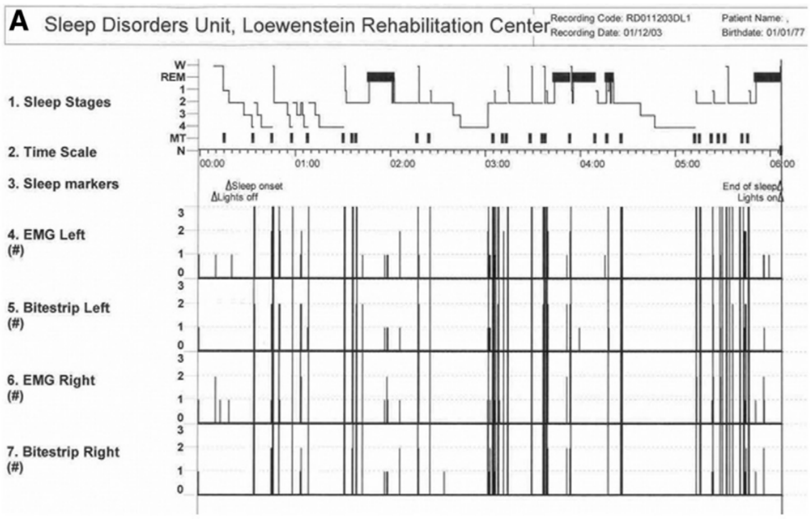


Figure 2.10: Comparision between MEMG and BiteStrip in SB patient [7]

Bruxoff

Bruxoff, in Fig 2.11, is a set of hardware and software used in the diagnosis of bruxism. Hardware-wise, **Bruxoff** is a compact device, consisting of 2 EMG electrodes and 3 ECG electrodes. EMG electrodes are used to record the muscle activity of the masseter muscle bilaterally. ECG electrodes are used to record cardiac activity (heart rate). The electrodes are disposable, being used only once. In terms of software (**Bruxmeter**), it can access the information recorded by the device, presenting it quickly and simply to the user [79, 80].

Figure 2.11: Bruxoff package

To obtain the muscle activity of the masseter, **CoDe** electrodes are used. These electrodes are supposed to have an advantage over conventional electrodes, as they are easier to apply and achieve high-quality data [79].

In terms of effectiveness, it is stated by the producers of **Bruxoff** that this device has a sensitivity[5] of 92% and specificity[6] of 85%. Different studies carried out point to different values, although not very divergent. Sensitivity is expected to be between 92% and 100% and specificity to be between 76% and 92% [80, 81].

One of the advantages of this device is the fact that it uses two physiological signals to identify the disorder. As already mentioned in the aetiology of bruxism, an event of bruxism is usually associated with an increase in heart rate.

GrindCare

The **GrindCare** was produced by Medotech and is used for the monitoring and management of bruxism. The system consists of Grinddock, stimulator, and EMG electrodes, as shown in Fig 2.12 [82, 83].

[5]**Sensitivity**, in medical diagnosis, corresponds to the true positive rate.
[6]**Specificity**, in medical diagnosis, corresponds to the true negative rate.

Figure 2.12: GrindCare device

The system has a biofeedback algorithm, which controls functional electric stimulation (FES), through electrical impulses. These impulses serve to interrupt muscle activity, relaxing the muscle. The level of feedback can be adjusted by the user via **Grinddock**, which communicates via Bluetooth with the stimulator. The stimulator has 3 EMG electrodes, placed on the temporalis muscle, since it is an active part of the mandibular activity and has a large contact surface, to detect muscle contraction during bruxism events. This information is transmitted *a posteriori* to **Grinddock**, which in turn will transfer it to a software program [82, 83].

According to a study by R. Needham and S. J. Davies, 19 patients suffering from sleep bruxism underwent the use of **GrindCare**, to verify its effect. About 58% of people show an improvement in both the effects and the number of bruxism events over time. The remaining subjects did not show any change, however, no aggravation was observed [83].

Nox T3

The **Nox T3** device, shown in Fig 2.13, is used to record physiological signals during sleep on an outpatient basis and can help diagnose various sleep disorders, including sleep bruxism. It is a small device (79x63x21 mm) and lightweight (88 g) and has Bluetooth technology, creating a WBAN [84, 85].

Figure 2.13: Nox T3 apparatus

The **Nox T3** device records signals from three integrated sensors and five external sensors. The integrated sensors include a pressure transducer, allowing snoring and mask/nasal pressure to be recorded, a three-dimensional acceleration sensor for measuring the activity and position of the user and a microphone for real audio recording capabilities, which can be useful in SB diagnosis [84, 85].

In terms of external sensors, there are the thoracic and abdominal respiratory inductance plethysmography (RIP) bands, as well as two channels for the use of thermistors, ECG, EMG, EEG, or pneumotachograph. The presence of Bluetooth allows the recording of signals from pulse oximeters compatible with the technology [84, 85].

BruxRelief

BruxRelief is a product developed by Beta Hacktory and found on Kickstarter. It is an intelligent device that detects and eliminates stress, treating bruxism disorder [8]. Its design is demonstrated in Fig 2.14.

This device has an innovative sensor technology for monitoring muscle activity during teeth grinding. Also, it performs biofeedback therapy to minimize the effects of stress and stop a certain behavior [8].

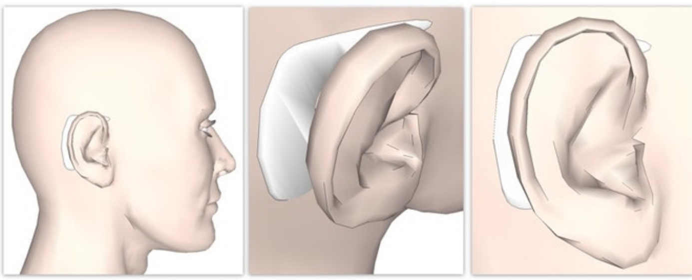


Figure 2.14: BruxRelief design[8]

The device is placed behind the ear and can be used during the day or night, and communicates via Bluetooth with a smartphone. Through the **BruxRelief** application, it is possible to monitor the stress level, occurrences of teeth grinding, and adjustment of vibration and audio signals used in biofeedback therapy [8].

Wireless Monitoring using Pressure-Sensitive Polymer

Kim *et al.*, developed a bite guard to monitor bruxism events and to protect teeth. Since the system approach is intra-oral, monitoring through EMG would not be the most appropriate, so a pressure-sensitive polymer was developed to measure the force exerted by teeth. The system allows real-time monitoring of bruxism events [9]. The actual prototype can be observed in Fig 2.15.

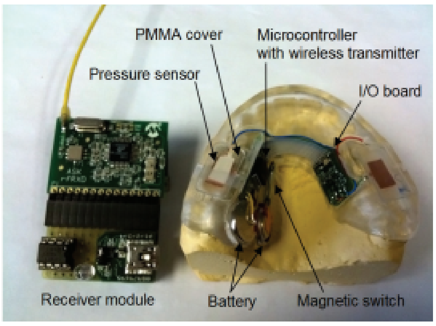


Figure 2.15: Bite guard prototype [9]

The system is divided into several phases. The pressure sensor is used to measure the force exerted, and the signal is conditioned. The data is then processed by an 8-bit microcontroller (rfPIC12F675) with built-in UHF ASK / FSK transmission. The microcontroller will send the information wirelessly, through a carrier frequency of 433.92 MHz. The receiver module contains a radio frequency receiver module (rfRXD420) and a microcontroller that registers the data sent from the bite guard. This checks if the received package has errors, forwarding the data to a computer where the data will be stored and logged [9].

The signal acquisition phase is encapsulated in a traditional bite guard, presenting low energy consumption (battery-powered). In standby mode, the current draw is only 0.76 mA and in transmission mode, it reaches 11 mA. This allows between 100-150 h of continuous transmission, so the system can reach months without needing to change the battery [9].

Intelligent Occlusion Stabilization Splint

Jinxia Gao ***et al***. sought to develop a system for the treatment and diagnosis of bruxism. For this, a pressure sensor was used, since it is an intra-oral system, integrated with a signal acquisition module, main control, and server. Artificial intelligence (AI) algorithms were used to parameterize biofeedback therapy [10].

The system consists of a signal acquisition module, composed of a piezoresistive sensor (BHF350-3AA), together with signal conditioning techniques. The main control module consists of an ultralow-power MCU (NRF52832), which performs the processing of occlusal force signals, performing analog-to-digital conversion (ADC). This System on Chip (SoC) supports Bluetooth, being able to transmit the acquired data to another module [10]. The system is shown in Fig 2.16.

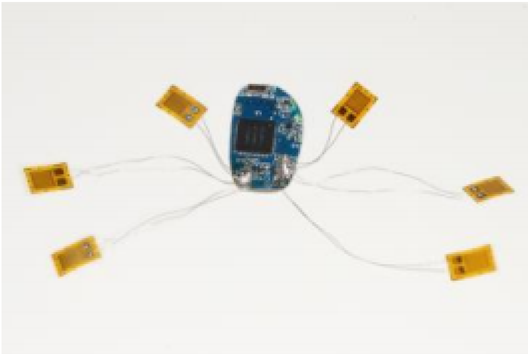


Figure 2.16: Stress-sensitive chips and control chips [10]

The transceiver module (STM32) receives the occlusal data, processes it, and forwards it to the server (HP Z840) with a Graphics Processing Unit (GPU) where the machine learning algorithm will be applied [10].

2.4 Technologies Review

Performing a study of several technologies is necessary to achieve the goal of detecting a bruxism event, leading to the construction of a system that can accomplish that task.

For this, it will be necessary to acquire physiological data from sensors. These signals will be conditioned to obtain the best possible signal. Then, the signal is read and processed by a processing unit that will wirelessly transmit it to an external unit. This process is summarized in the Figure 2.17.

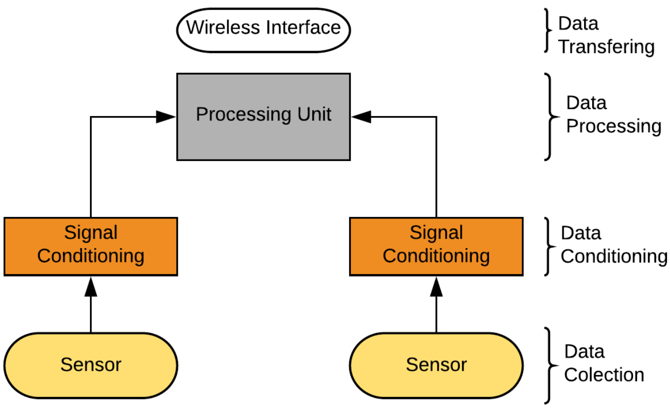


Figure 2.17: Device Architecture

Additionally, there is a system that can aggregate the information transmit-

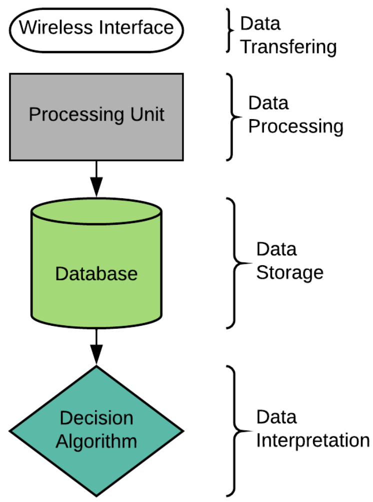


Figure 2.18: External Architecture

ted by the device, storing it in a database. From the collected data, the objective is to apply an algorithm that is capable of detecting a bruxism event.

2.4.1 Bite Force Devices

Since the most important parameter to be recorded is the force or pressure exerted between the patient's teeth, the intended system will be implemented intraorally. Various existing options will be analyzed (design and working principle), taking into account the advantages and disadvantages of their use.

Strain-gauge

A strain gauge is a device made up of a metal foil or fork. It is then coupled to an object that will be the subject of a force, leading to its deformation. The sensor, attached to the object, will also undergo a similar deformation causing a change in its electrical resistance. The type of deformation will slightly modify the conductor's geometry, which can either be stretched (increases the resistance) or shrunk (decreases the resistance) [29].

For this variation in the electrical resistance of the sensor to produce a change in voltage, the sensor is normally connected to a Wheatstone Bridge circuit and can be calibrated using a known weight. The orientation of the sensor can affect the result, since there are different geometries of the strain gauges depending on the type of force expected to be applied to it (uniaxial, biaxial, etc.). Furthermore, variations in temperature will affect the electrical resistance of the sensor

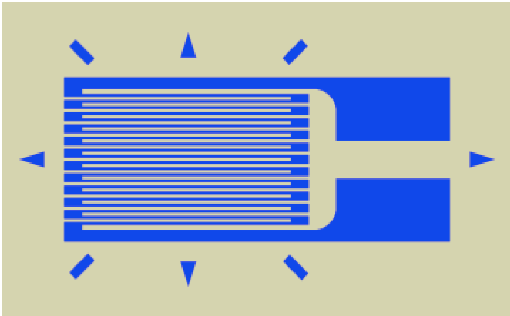


Figure 2.19: Strain Gauge [11]

(thermal expansion) and compensations are necessary to avoid causing inaccurate readings [29].

Although these sensors have a good dynamic range and can be accurate, they may cause some discomfort in the patient. The sensors are usually wrapped in a cover, such as acrylic resin and polyvinyl chloride, to mitigate any fear on the part of the patient. However, this cover will never be the perfect solution as it increases the volume of the package itself [29].

Piezoelectric

A piezoelectric sensor makes use of the piezoelectric effect, discovered by Pierre Curie in 1880, in which changes in pressure, force, temperature, and other physical stimulus, result in a variation in electric charge, subsequently being amplified [29].

Its use has been increasingly emphasized in several industries, such as medical, automotive, aerospace, among others. They hold a high modulus of elasticity[7] and an extremely low degree of deflection[8], which results in an extremely high natural frequency and excellent linearity. They are also recognized for acting in adverse environments, as they are immune to radiation and electromagnetic fields, and can also present stability at high temperatures [29].

The manufacture of piezoelectric transducers in the form of thin foils of crystals has been implemented, with a thickness of only 2 µm, being able to be designed and cut with the desired shape. This is an advantage in measuring occlusal forces, as you need minimal jaw opening. However, the accuracy of these sensors is not yet well established [29].

[7]**Modulus of elasticity** quantifies a material's resistance to elastic deformation.
[8]**Deflection** defines the degree of displacement of an element under a load.

Pressure

A pressure sensor can be constructed in different ways, and may even include some of the technologies mentioned above, but it mainly consists of a chamber composed of air or fluids and which varies its pressure according to the applied force. This pressure is then measured and the applied force can be determined since the pressure is given by the force per unit area [29].

These sensors can be classified according to the type of pressure they measure, such as absolute, gauge, differential, vacuum, and sealed pressure [29].

MEMS Capacitive Sensor

Microelectromechanical systems are integrated devices or systems that combine electronic and mechanical components, which interact and operate in controlled environments. The use of these systems offers some advantages over other alternatives, hence the fact that they can operate on scales microscopic, but also the low consumption of power, high performance, low weight, resulting in a lower cost [86, 87].

The capacitive sensor is part of the pressure sensors mentioned. This type of sensor is used due to its biocompatibility, low energy consumption, and low thermal sensitivity. A capacitive sensor is usually composed of a diaphragm that can influence its functioning by varying its size. The geometry of the capacitor itself can influence the sensitivity and dynamic range [86, 87].

The functioning of the sensor depends on the variation of its capacitance according to the pressure exerted on the diaphragm, which in turn will change the distance between the capacitor plates [86, 87].

2.4.1.1 Available Bite Force Systems

There are several systems for measuring human bite force on the market, mostly consisting of sensors previously mentioned. All systems have advantages and disadvantages. It must be considered that it is a difficult variable to measure, especially if it is done continuously, and it is necessary to think about the patient's comfort. In the Table 2.2, several systems are presented, as well as their characteristics regarding composition, dimension and dynamic range.

Table 2.2: Comparision between different bite force systems [29]

SYSTEM	DEVELOPER	TECHNOLOGY	DIMENSION	DYNAMIC RANGE
Dentoforce 2	ITL AB	Strain gauge with soft rubber cover	11mm (thickness)	<1000N
IDDK	Kratos	Dynamometer	14.6mm (thickness)	<1000N
GM10	Nagano Keiki	Hydraulic pressure gauge	195 (l) x 29 x (w) x 18 (t) mm	<1000N
T Scan III	Tekscan	Pressure sensitive ink grid	0.1mm (thickness)	N/A
Dental Prescale	GC Co. Ltd	Pressure sensitive horse shoe shaped film	97 - 800 \mu m (thickness)	50 - 1200 kgf/cm² 30 - 130 kgf/cm²
MPX 5700	Motorola	Tube to measure air pressure	7mm (diameter)	N/A
FSR No. 151	Interlink Electronics Inc.	Piezoresistive sensor	12mm (diameter), 0.25mm (thickness)	N/A
MPM -3000	Nihon	Force transducer	Plate: 17mm (diameter) Block: 3mm (diameter), 3mm (thickness)	N/A
Flexiforce	Tekscan	Piezoresistve load cell	150 (l) x 10 (w) x 0.2 (t) mm	<4500N

2.4.2 Signal Conditioning

Signal processing is a technique that allows analysis, quantification, and modeling of physical events. A signal can be described as a physical variable that varies over time and/or space. With signal processing, it is possible to manipulate that signal to change its characteristics, extracting information.

Analog Signal Processing

Analog signal processing is performed on continuous analog signals and uses techniques to change the input signal (**x(t)**) and produce a more appropriate output signal (**y(t)**), as described in Fig 2.20. The input signal is usually described as the excitation signal and the output (processed) signal as the response signal [12].

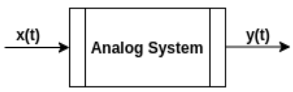


Figure 2.20: Block Diagram of Analog System [12]

Wheatstone Bridge

From its year of invention, 1833, the Wheatstone Bridge is used to measure a variable resistance with extreme precision. This method of instrumentation electronics is one of the most recommended for small-signal conditioning [88].

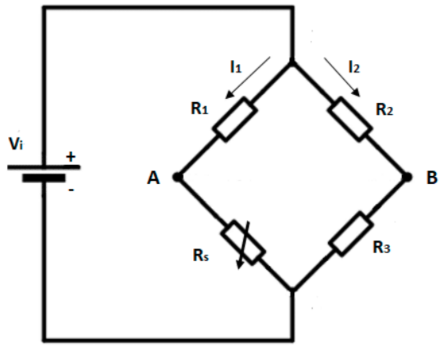


Figure 2.21: Wheatstone Bridge Circuit

The Fig 2.21 represents the scheme of the Wheatstone Bridge, where $\mathbf{R_S}$ represents the variable resistance. The objective is to calibrate the point so that the potential difference between points A and B ($\mathbf{V_{AB}}$) is 0V. As the resistance changes, this potential difference will also change and can be measured using the following equations [88].

$$V_A = \frac{R_S * V_i}{R_S + R_1} \tag{2.1}$$

$$V_B = \frac{R_3 * V_i}{R_3 + R_2} \tag{2.2}$$

Amplifier Circuit

After the 60s, the commercialization of a component that allowed the realization of arithmetic operations between analog signals called an operational amplifier (OpAmp) begun. Being an element consisting of two inputs and one output, it has a high impedance in the input and output low, allowing a high gain for an analog signal. This constituent has several applications, such as process control or frequency filtering, it can also function as a simple amplifier of voltage, as well as a differential amplifier [89].

The differential amplifier thus appears in a period of discoveries of operational amplifier applications. This electronic circuit capable of performing the difference between two analog inputs, amplifying this output voltage. The agglomeration of some characteristics present both in the differential and in the operational amplifier introduces the instrumentation amplifier [89]. The schematic can be analysed in Fig 2.22.

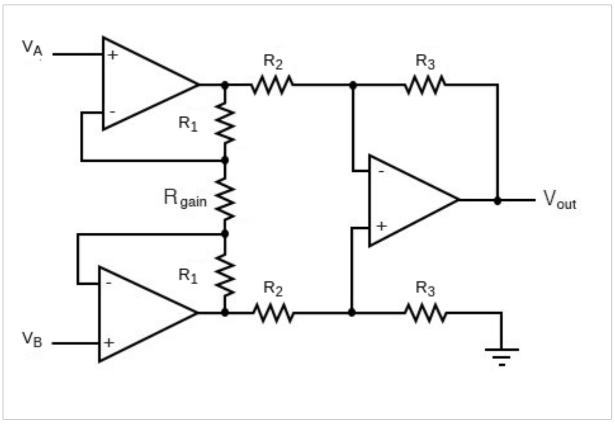


Figure 2.22: Instrumentation Amplifier [13]

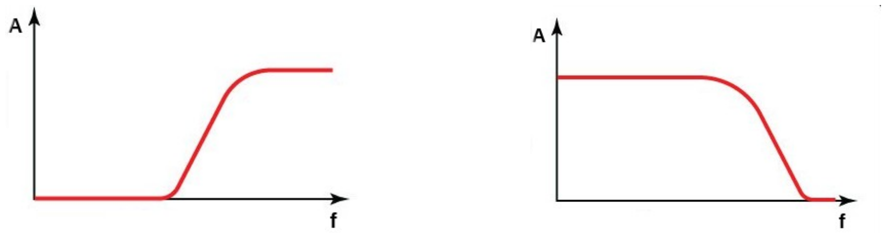
The output voltage of this circuit (**V_{OUT}**) is given by the following equation.

$$V_{OUT} = V_{AB} * (1 + \frac{2R_1}{R_{GAIN}})(\frac{R_3}{R_2}) \tag{2.3}$$

Analog Filters

A filter is one of the most important electronic systems, and its use is to remove unwanted frequency components. There are different types of filters, such as active and passive, linear and non-linear, analog and digital filters, among others [90].

The **high-pass filter (HPF)** and the **low-pass filter (LPF)** are the simplest and perhaps the most common filters. HPF removes the lowest frequency components, leaving the highest ones unchanged. LPF does the reverse, blocking the higher frequencies. The Fig 2.23 shows the behavior of the filter in terms of amplitude as a function of frequency [90].



(a) High-Pass Filter [91] (b) Low-Pass Filter [91]

Figure 2.23: Filters behavior

It is possible, through the combination of HPF and LPF, to build two more types of filters [90]:

- **Bandpass filter (BPF)**: allows a range of frequencies to pass.
- **Bandstop filter (BSF)**: removes a range of frequencies.

It is possible to build simple filters with passive components such as the resistor (R), capacitor (C), and an inductor (L), called RC, RL, and RLC filters. The order of the filters can be changed by adding cascading first-order filters, which defines the filter's attenuation. There are more complex filters like the case of Butterworth, and Chebyshev [90].

Digital Signal Processing

Digital Signal Processing (DSP) represents a series of digital operations on digital signals. These signals are a discrete representation of an analog (or continuous) signal and must be digitized through an ADC. DSP has many advantages such as error detection and correction or even data compression.

Fast-Fourier Transform

Fourier analysis converts equally-spaced samples of a signal to a representation in the frequency domain. Fast Fourier Transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT), in a more efficient and fast way. The N-size DFT can be calculated using the following equation [92].

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \tag{2.4}$$

One of the most common algorithms is Cooley-Tukey, which is based on the decomposition of a DFT of size N into smaller DFTs. Another class of FFT algorithms is the Prime-Factor Algorithms (PFA), in which there is a decomposition of DFT into smaller DFTs and their size is mutually prime [92].

Finite Impulse Response Filter

A Finite Impulse Response (FIR) filter can be used for low-pass, high-pass, bandpass, and notch filtering. These filters have no feedback, making them inherently stable. Furthermore, they can produce linear phases [93].

An FIR filter is designed specifying the transfer function H(w), which represents the desired frequency response for the filter, being then converted to a sequence through the inverse discrete Fourier transform, resulting in h[n], i.e., the coefficients of the filter. Applying the convolution between x[n] and h[n], input signal, and filter sequence respectively will result in y[n], the output signal, as noted in the following equations [93].

$$h[n] = \sum_{i=0}^{N} b_i \cdot \delta[n-i] \tag{2.5}$$

$$y[n] = h[n] \cdot x[n] \iff y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + ... + b_N \cdot x[n-N] \tag{2.6}$$

Infinite Impulse Response Filter

An Infinite Impulse Response (IIR) filter corresponds to a digital filter that combines an FIR filter with feedback. Thus, it depends on a linear set of input samples and previous output samples. An IIR, given a finite set of input samples, can oscillate infinitely if it is not a stable filter, and if it is then the output will drop to zero. The transfer function is given by equation 2.7 and the differential equation that represents the filter is given by equation 2.8 [94].

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 \cdot z^{-1} + ... + b_M \cdot z^{-M}}{1 + a_1 \cdot z^{-1} + ... + a_N \cdot z^{-N}} \tag{2.7}$$

$$y[n] = b_0 \cdot x[n] + ... + b_M \cdot x[n-M] - a_1 \cdot y[n-1] - a_N \cdot y[n-N] \tag{2.8}$$

In an IIR filter, the frequency is a factor that influences both the attenuation and the delay. Furthermore, low frequencies are more delayed than high frequencies. Unlike an FIR filter, an IIR filter is not capable of a linear phase [94].

2.4.3 Processing Unit

The microprocessor module is one of the most important in product design. Lately, there has been an exponential growth in these systems, with practically all products having an intelligent module, managing to transform a normal system into a system able to make decisions.

There is a wide variety of microprocessors or microcontrollers on the market, which can be distinguished by their architecture, processing capacity, speed, energy consumption, and additional peripherals.

nRF52

The nRF52 SoC series, produced by Nordic Semiconductor, is described as being extremely fast and efficient. It is capable of performing complex tasks in a short period, returning to sleep mode, and thus saving battery. The nRF52 has an automatic power management system and includes 6mm x 6mm QFN and 3.0mm x 3.2mm CSP packaging options [95].

nRF52832

The nRF52832 consists of an ARM Cortex M4, at a frequency of 64MHz. Supports BLE (high-speed 2Mbps), NFC, ANT, and other proprietary 2.4GHz protocols [96].

Depending on the package, it can have 512 / 216KB of Flash and 64/32 KB of RAM. Its operating voltage can vary between 1.7 and 3.6V, facilitating its use with a battery. In terms of energy consumption, these devices are designed to have low current consumption, and in sleep mode, it can be only 2-3 μ A [96].

Another important aspect is the peripherals. The nRF52832 has an ADC with a 12-bit resolution and 200 ksps. It also has SPI, I^2S, UART interfaces, and a 32-bit timer/counter. Furthermore, it also has 32 configurable GPIOs [96].

The application of these devices is vast and can be used in home automation, personal area networks, including monitor and medical devices, remote controls, and computer peripherals [96].

nRF24

The nRF24 series represents a set of integrated circuits (IC) using the Nordic Semiconductor's Enhanced ShockBurst protocol (ESB), to allow the implementation of ultra-low-power and high-performance communication. These devices differ from those previously presented in the sense that they are not necessarily SoCs, but instead work in conjunction with host microcontrollers [97].

nRF24L01+

The nRF24L01 + is a low cost, low power operation, and small (4x4 mm) 2.4 GHz transceiver [98].

It has an operating voltage between 1.7- 3.6V and with current consumption of 26μ A in standby mode. This device must be used together with a microcontroller host since its only function is to handle the transmission and reception of packets [98].

It can be used in PC peripherals, game controllers, home automation, active RFID, toys, and others [98].

ESP32

ESP32 is a series of low-cost, low-power SoC microcontrollers. Produced by Expressif Systems, these systems use the Tensilica Xtensa LX6 microprocessor, with integrated Wi-Fi and Bluetooth. These microprocessors can be dual-core or single-core, operating at 240/160 MHz. They also include an ultra low-power (ULP) co-processor and have 520 KB of SRAM [99].

In terms of peripherals, ESP32 has 12-bit ADC with 18 channels, 8-bit DAC, SPI, UART, I²S, I²C, and others [99].

As for energy management, the ESP32 has a current consumption in sleep mode of around 5 mA, which can be activated through interruptions GPIO, timer, ADC. This allows energy consumption to be minimal, with the microcontroller being only active when necessary [99].

As for security, ESP32 supports IEEE 802.11 standard security (WPA / WPA2, for example), it has flash encryption and counts with cryptographic hardware acceleration [99].

There are different chips and modules from the ESP32 series, varying the processing capacity and the number of peripherals between them. The dimensions of these are around 6x6 mm (ESP32-D0WDQ6), 5x5 mm (ESP32-D2WD), and 7x7 mm (ESP32-PICO-D4, a module containing the chip, oscillator, and flash memory) [99].

Module Comparison

In Table 2.3, a comparison is made between some of potential modules used for these systems. The major difference between them is the nRF24L01+, which is not a SoC, but rather a transceiver that should be used in addition with an external processing unit. Due to the amount of constrains, mainly in the size department, this module won't be used. The remaining modules are very similar, differing mainly on cost and memory. ESP32 is, by far, the most powerful module but is not as low-power as the nRF series and, for the application in question, it's unnecessary.

2.4.4 Detection Algorithm

Another important step of the system involves the interpretation of the collected physiological signals. There is a major paradigm shift in terms of diagnosis. It is known that it was common practice to decide on a patient's condition based on the assessment of one or more doctors. Therefore, there is a need to automate this process, by utilizing machine learning algorithms. However, this involves creating large datasets specific to a given condition.

Decision Tree

The decision tree algorithm is a supervised learning algorithm, used to solve regression or classification problems. The use of this algorithm, and practically all machine learning algorithms, involves a learning stage and a prediction stage.

Table 2.3: Comparision between different modules

Device	Voltage Operation [V]	Flash/RAM [kB]	Wireless Interface	Peripherals	Additional Hardware	Low-Power	Package Size [mm]	Cost [€]
nRF52832	1.7 - 3.6	216 - 512 / 32 - 64	BLE, NFC, ANT, and others	ADC, SPI, I^{2}S, UART, and others	No	Yes	3.0 x 3.2	4.91
nRF52833	1.7 - 5.5	512 - 128	Bluetooth 5.1, 802.15.4, and others	ADC, UART, SPI, and others	No	Yes	3.0 x 3.2	6.15
nRF52840	1.7 - 5.5	1024 - 256	Bluetooth, ZigBee, Thread, and others	ADC, UART, SPI, and others	No	Yes	3.0 x 3.2	5.45
nRF5340	1.7 - 5.5	1024 - 512	Bluetooth, ZigBee	ADC, low-power comparator, RTC, and others	No	Yes	7.0 x 7.0	8.09
nRF24L01+	1.9 - 3.6	-	2.4 GHz	-	Yes	Yes	4.0 x 4.0	3.02
ESP32	2.7 - 3.6	4094 - 520	Bluetooth, Wi-Fi	ADC, UART, SPI, and others	No	Yes	7.0 x 7.0	4.05

The decision tree classifier is based on the prediction of a target variable as a function of simple decision rules, as observed in Fig 2.24. It takes into account several variables and, applying the decision rules, makes a prediction [14].

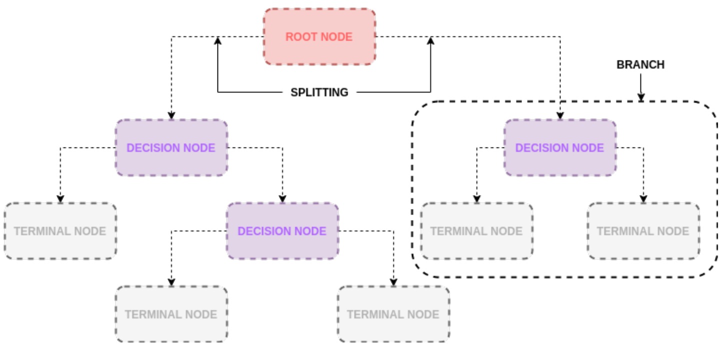


Figure 2.24: Decision Tree Schematic [14]

Several terminologies should be considered in the decision tree, the most important being [14]:

- **Root node** refers to the entire population and is divided into two or more groups.
- The **decision node** represents a rule to be considered in the decision, so there is a split.

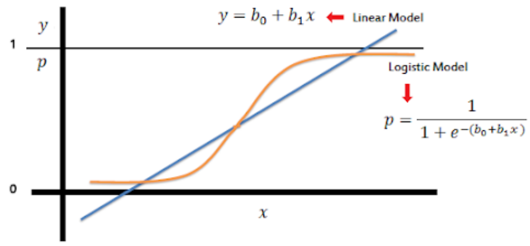


Figure 2.25: Logistic Regression [15]

- The **terminal node** represents a conclusion or prediction.

Logistic Regression

Logistic regression is not only a traditional statistical technique but can be used as a machine learning classifier. In machine learning, logistic regression can be used to classify a variable as true or false, depending on an input variable. This technique results in a "curve" or sigmoid, between 0 and 1, which will dictate the probability of an event [100], as shown in Fig 2.25.

Simple models can be formed, in which the output variable depends only on one input variable, or more complex models, in which the output variable depends on more than one input variable. Furthermore, it is possible to quantify the effect of an input variable on the prediction of the output variable using, for example, Wald's Test [100].

To calculate sigmoid, something similar to maximum likelihood is used. The goal of maximum likelihood is to find the optimal distribution to fit the data. The process is to calculate the likelihood of several curves, finding the optimal curve [100].

K-Nearest-Neighbor

K-Nearest-Neighbors is an algorithm that classifies a sample based on a known dataset. Given a labeled dataset, the algorithm will compare a new sample with the nearest K samples, classifying the new sample according to the type with the closest samples. In the Fig 2.26, this process is explained visually, being simple to understand and apply [101].

A recommended condition for the application of this classifier is to use an odd K so that there is not an integer division of neighboring samples. In this type of algorithm, the initial data clustering is defined as "training". Furthermore, there is no ideal method for choosing K, so its assignment should be based on trial-and-error. However, lower values of K may not be the best option (subject to outliers) and high values of K are also not recommended (category with fewer samples may be outvoted) [101].

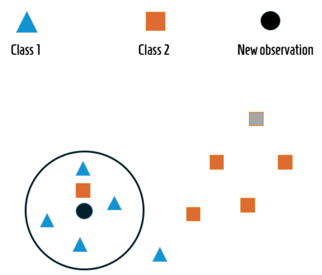


Figure 2.26: Representation of K-Nearest-Neighbor algorithm

Hidden Markov Models

A Markov chain or model is used to predict a certain state based on a number of previous states. The number of previous states used, corresponds to the order of the Markov model.

These models are described with transition probabilities, i.e, the probability of moving from one state to another. The Fig 2.27 depicts the process in which there are n states and a probability is given of a transition from i state to j, shown as $t_{i,j}$.

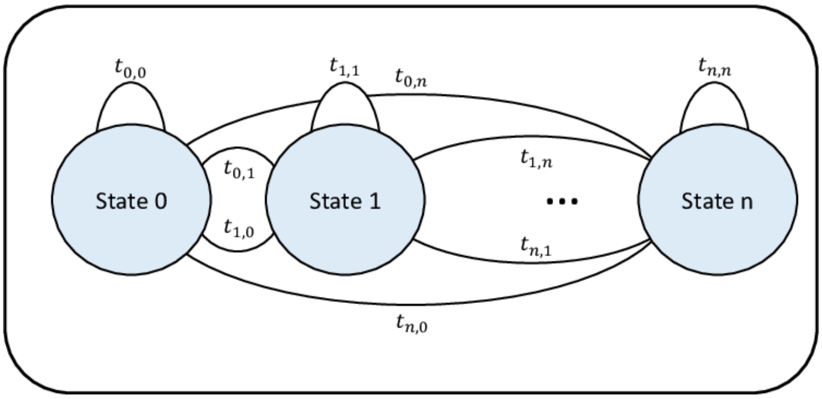


Figure 2.27: Discrete-time Markov Chain [16]

In a Hidden Markov Model (HMM) there is no actual knowledge of the state of the system. Instead, there are observations that indicate its true state.

Mathematically, it results on a model that takes into consideration the probability of an observation given the true state [$p(y_t \mid x_t)$], where y_t) are the observations and x_t is the true state, and the transition probability [$p(x_t \mid x_{t-1})$] which indicates the probability of the true state given its previous state. So, the HMM is described as the following equation:

$$p(X,Y) = p(x_1) \prod_{t=1}^{T-1} p(x_{t+1}|x_t) \prod_{t'=1}^{T} p(y_{t'}|x_{t'}) \tag{2.9}$$

Bayesian Networks

The Bayesian Network is a probabilistic graphical model (PGM) that correlates a certain number of variables and their conditional dependencies. This model can be used to analyze an event and relate it to known causes. The Fig 2.28 represents a Bayesian network in which each node corresponds to a unique variable, and each connection or edge corresponds to a conditional dependency.

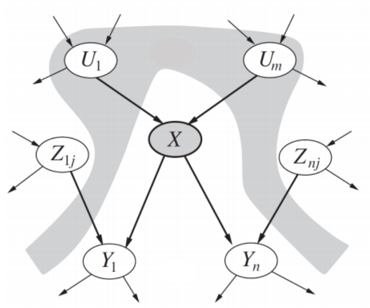


Figure 2.28: Bayesian Network

Bayesian and Markov networks are very similar in their representation of variables and dependencies, but they differ in the fact that Bayesian networks are directed and acyclic and Markov networks are undirected and cyclic. Also, each node is conditionally independent of its non-descendants. The joint distribuition for a Bayesian Network is given by the equation 2.10:

$$P(X_1,...,X_n) = \prod_{i=1}^{n} P(X_i|P(X_i,...,X_{i-1}) = \prod_{i=1}^{n} P(X_i|Parents(X_i)) \tag{2.10}$$

These types of models can be used for disease detection, given a set of symptoms.

Chapter 3

System Architecture

The idealized system involves the construction of an intraoral device that performs the collection of physiological data for a possible diagnosis of sleep bruxism. In the construction of this device, there are some important aspects to take into consideration, such as the dimension, the autonomy, and the parameters to be measured.

General Concept

The system will then be composed of a microprocessor, to manage and process the tasks to be performed by the device, by one or more sensors, which collect the data, power, which provides power to the system, and also a wireless interface, to proceed with the transmission of collected and processed data to an external unit.

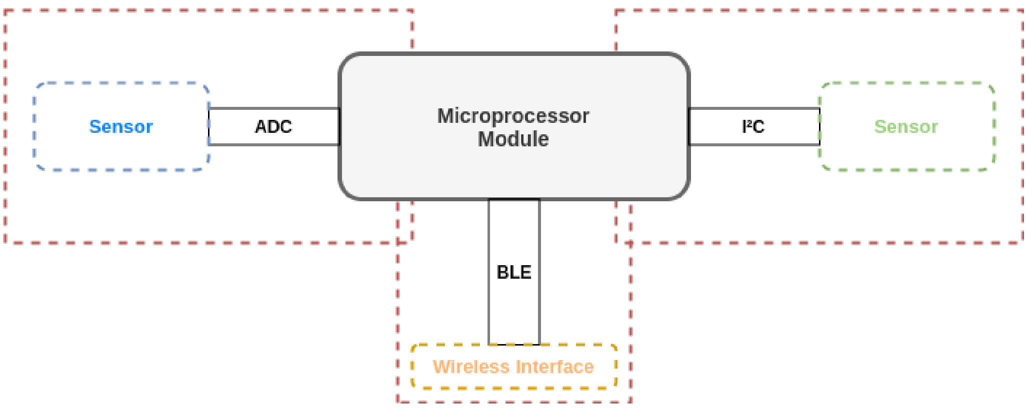


Figure 3.1: System Architecture

The size of the device is one of the fundamental aspects to be taken into account in its construction. This has to involve the least discomfort for the patient for greater precision of the collected data. Since the goal is to integrate the device in an occlusal splint, the device can be as big as the splint. According to a study by Haidi Omar *et al.* [102], the average size of the upper dental arch of a human is 76.51 $\pm$ 9.0 mm, which will serve as a limit for the design of the device. Thus, a 65.00 mm long and 17.50 mm wide device was projected.

In terms of design and/or component layout, the device will consist of a USB port, the circuit referring to the power supply (includes a battery protection circuit and a regulator), the microprocessor, sensors, and also the antenna.

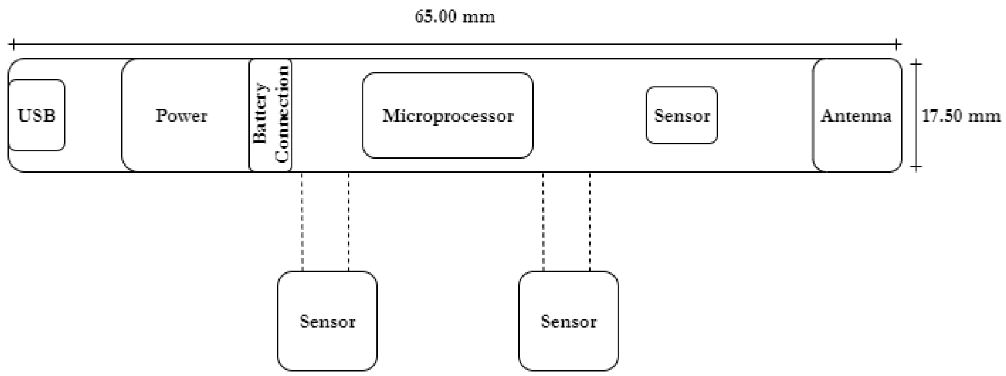


Figure 3.2: System Schematic

The placement conceived for the device can be seen in Fig 3.3, which is placed in the upper dental arch to promote greater utilization area and even greater comfort for the patient.

Figure 3.3: Device placement

Power

The portable device must have autonomy so it needs the presence of a battery. In this regard, several options are applied to different applications and can be divided into two major categories:

- **Primary**, which is a disposable battery, since it cannot be recharged due to changes in its chemical composition. It can produce a current immediately after assembly. Normally, it has a higher energy density than rechargeable batteries. One of the greatest examples of this type of battery is the alkaline batteries. In this category, the lithium cells are a nice alternative.

- **Secondary**, which is a rechargeable battery. These kinds of batteries are more expensive per Watt-hour (Wh) of energy than primary batteries but can be recharged numerous times, which minimizes waste. They are widely used in market applications and can be of different chemical compositions, such as nickel-metal hydride (NiMH), Li-ion (Lithium-ion), and Li-Po (Lithium Polymer).

For this system, there are some constraints to take into account. The size of the battery, its flexibility, its capacity, and whether or not it is rechargeable. Other aspects to consider are battery life, rate of self-discharge, and charge efficiency. What is intended is that it is an autonomous and durable system, but at the same time, it is as compact as possible.

For its implementation, the battery pack must be as thin as possible since it will have to be integrated into a limited space, therefore, Li-ion and Li-Po batteries (secondary) and lithium coin cells (primary) are the only ones that meet the requirements. Li-Po and Li-ion batteries have high energy density and longer runtimes. Also, they don't have the so-called memory effect, which is a phenomenon that can affect the battery's capacity. On the other hand, they do start to wear out over time, due to the repeated charge and discharge cycles. Moreover, the lithium coin cell has a higher energy density.

It's also interesting to examine the volumetric energy density (V.E.D.), which corresponds to the capacity of the battery per liter (Wh/L), and also the gravimetric energy density (G.E.D.), which indicates the capacity of the battery per kilogram (Wh/kg). The Table 3.1 shows the comparison between Li-ion, Li-Po, and lithium coin cells (CR1632) batteries in relation to their V.E.D. and G.E.D.. However, it should be noted that these are approximate values and vary between types of manufacturers.

Battery Type	V.E.D. (Wh/L)	G.E.D. (Wh/kg)
CR1632	$\cong 780$	$\cong 209$
Li-ion	$\cong 540$	$\cong 205$
Li-Po	$\cong 360$	$\cong 145$

Table 3.1: Comparison of battery types

Sensors

Due to the limited space and the framing of the device, the choice of parameters to be monitored is a bit tricky. The most obvious choice, which was discussed in the previous chapter, is the measurement of the force exerted between the teeth.

Another parameter that could be measured, is sound. This is one of the most important variables for detecting bruxism and, given the constant developments in this area, there are microphones so small that makes it possible for their inclusion in a device with so many constraints as this. The main drawback of using a microphone is that, with the use of an occlusal splint, the noise that previously existed, may be reduced or mitigated, and as the device will be in a closed environment the sound can be muffled and difficult to detect.

Other variables that will be considered are the heartbeat since there are studies that correlate the increase in a heartbeat with bruxism events, however, its placement would not be easy and, the acquisition of reliable readings is unlikely. The introduction of an accelerometer will be also considered, as both the movement and the position in which a person sleeps can be a factor to be taken into account, but it would only take into consideration the movement or position of the patient's head. This will be a trial and error process, to try to understand what data can be collected, whether or not they are reliable and how they influence a possible diagnosis.

Communication Protocol

Another very important aspect to consider is the type of communication to be used for data transmission between the device and a central unit. When talking about this unit, it is not referring to the integrated microcontroller, but rather to a separate unit that collects and stores data and that can be used to make a decision on whether it's a bruxism event or not.

Despite the innumerable protocols in the industry, what is intended for this device is that it can transfer data wirelessly, with the lowest possible energy consumption. As it is a device that may eventually be commercialized, compatibility is also a fundamental aspect, as it will be necessary to be able to communicate with devices common to anyone, such as smartphones.

Figure 3.4: Typical FSR model [17]

Given this, there are two main protocols: Wi-Fi or Bluetooth Low Energy. Although Wi-Fi has a much higher bandwidth than BLE, it loses in the aspect of energy consumption and, as this is a key factor, BLE will be the chosen protocol.

3.1 Sensors

To obtain the force exerted between the teeth, some options will be tested. Although there is no ideal solution, since the force to be measured is not uniform, some sensors will be applied to verify the results and those that best suit the application.

One of the options is the sensors from Interlink Electronics, which features three series of Force Sensitive Resistors (FSR): the basic series, the X series, and the UX series. These are very similar in design, varying in their dynamic range, linearity, and type of application. A typical model is shown in Fig 3.4 [17].

As for the differences between the models, and since there a number of them, a comparison is summarized in the Table 3.2, making reference to parameters such as active area, length, and dynamic range. Also, it's important to note that not all of the FSR are the same geometry. There are circular, rectangular, strip, and FSR ring, which make them flexible for a number of applications [17].

These devices are piezoresistive which means they change their internal resistance when subjected to a force. The characteristic graph that correlates force and internal resistance is shown in Fig 3.5, and can be concluded that as more force is exerted on the sensor, its resistance will decrease accordingly [17].

Another type of device that will be evaluated is from Tekscan and, more specifically, from the FlexiForce series. Tekscan is a highly rated company that produces high-quality sensors and offers specific solutions for measuring dental strength. Within the FlexiForce series, there are several options to suit certain applications. The example of a sensor can be seen in Fig 3.6 [19].

Within the FlexiForce series, there are several models of sensors that vary in their geometry, size, and dynamic range. The differences between the models

Table 3.2: Comparasion between different FSR models

Model	Active Area	Length [mm]	Nominal Thickness [mm]	Switch Travel [mm]	Dynamic Range [N]
400	$\phi 5.08mm$	38	0.30	0.05	~0.2 - 20
400 (Short)	$\phi 5.08mm$	20	0.30	0.05	~0.2 - 20
X 400	$\phi 5.08mm$	38	0.30	0.05	0.3 - 50
UX 400	$\phi 5.08mm$	38	0.30	0.05	0.5 - 150
402	$\phi 14.68mm$	56	0.46	0.15	~0.2 - 20
402 (Short)	$\phi 14.68mm$	30	0.30	0.05	~0.2 - 20
X 402	$\phi 14.68mm$	56	0.46	0.15	0.3 - 50
X 402 (Short)	$\phi 14.68mm$	30	0.46	0.15	0.3 - 50
UX 402	$\phi 14.68mm$	56	0.46	0.15	0.5 - 150
UX 402 (Short)	$\phi 14.68mm$	30	0.46	0.15	0.5 - 150
406	34 mm^2	83	0.46	0.15	~0.2 - 20
X 406	34 mm^2	83	0.46	0.15	0.3 - 50
UX 406	34 mm^2	83	0.46	0.15	0.5 - 150
408	10.2 mm	56	0.41	0.15	~0.2 - 20
X 408	10.2 mm	56	0.41	0.15	0.3 - 50
UX 408	10.2 mm	56	0.41	0.15	0.5 - 150

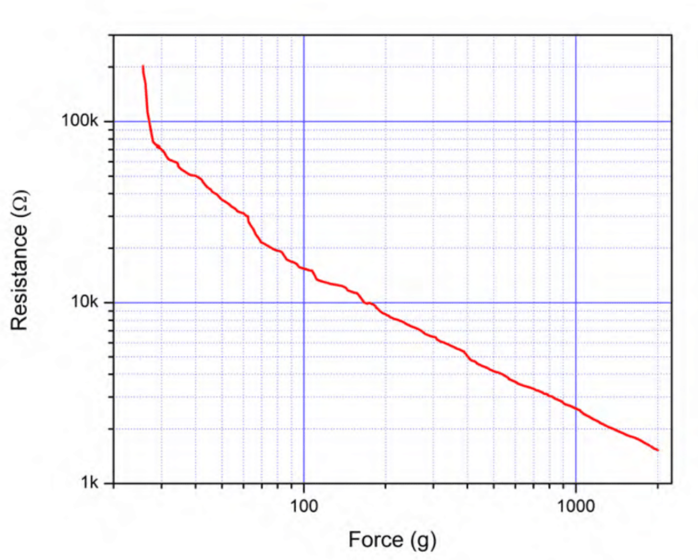


Figure 3.5: Typical FSR graph for resistance over force [18]

Figure 3.6: FlexiForce Sensor [19]

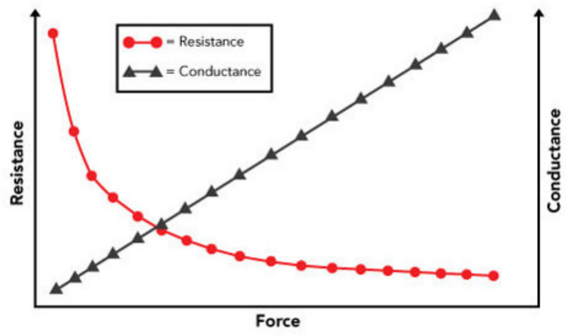


Figure 3.7: Typical variation of resistance and conductance of a FlexiForce sensor [20]

are summarized in the Table 3.3. It's also important to mention, that FlexiForce can be custom designed, which can be important for some applications [19].

Table 3.3: FlexiForce sensors comparision [30]

Model	Active Area [mm]	Length [mm]	Standard Force [N]
A101	3.81	15.7	18 - 44
A201	9.53	190.5 - 152.4 - 101.6 - 50.8	4 - 111 - 445
HT201	9.53	190.5 - 152.4 - 101.6 - 50.8	222
A301	9.53	25.4	4 - 111 - 445 - 4448
ESS301	9-53	25.4	4 - 445
A401	25.4	56.9	111 - 31138
A502	50.8	81.3	222

These sensors are also piezoresistive, varying their internal resistance according to the force applied. This variation is represented in Fig 3.7, observing a decrease of the internal resistance with the increase of the force [19].

As for these sensors, the ideal implementation would be of the integration of, at least, two sensors, one on each side of the mouth. Because of the constraints inherent to the environment, the number of sensors must be limited, so they have to be placed in a strategic place to increase precision.

Ideally, the sensors would be implemented between the molars and bicuspids, since it is there where the force is more noticeable. The position of the sensors could be as the Fig 3.8 shows. On the other hand, the sensor itself should be protected by some sort of cover, in order to protect it against irregular forces exerted by the the teeth, and due to the harsh and irregular tooth surface.

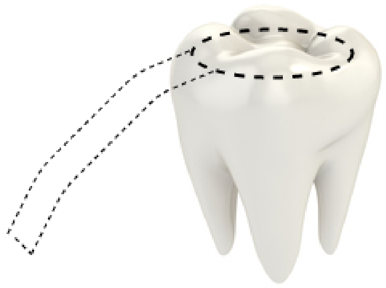


Figure 3.8: Sensor placement

Depending on the results obtained, another option can be explored. This involves the construction of a sensor, using a pressure-sensitive conductive sheet, such as Velostat, which consists of polymeric foil and carbon black. Its composition gives it two important properties, such as the fact that it is electrically conductive and that the internal resistance varies when a force is applied. Not being the ideal option, it can be interesting to understand its performance and the possible results.

As mentioned before, an accelerometer will be used to estimate the sleeping position of the patient. But before exploring the different options available in the market, it's important to understand the working principle of an accelerometer.

An accelerometer is a device that can measure static (gravity) or dynamic forces (movement) of acceleration. An accelerometer can measure the acceleration up to 3-axis (x, y, z), which are the most popular, and can be observed in Fig 3.9.

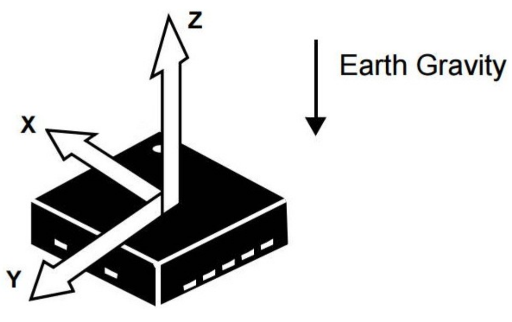


Figure 3.9: Accelerometer axis [21]

This can be achieved by using MEMS, which turns a mechanical variable into an electrical signal. Typically, an accelerometer is composed of a mass attached to a spring, and fixed plates. So, when an acceleration in a particular direction occurs, the mass will move, resulting in a change in capacitance. This change in

Table 3.5: Comparision between IMU

Device	Sensor Type	Communication Interface	Operation Voltage [V]
MPU-6050	Accelerometer Gyroscope Temperature	I^2C, SPI	2.375 - 3.46
LSM6DSMTR	Accelerometer Gyroscope Temperature	I^2C, SPI	1.71 - 3.6
BMI160	Accelerometer Gyroscope	I^2C, SPI	1.71 - 3.6
ICM-42605	Accelerometer Gyroscope	I^2C, SPI, $I3C^{SM}$	1.71 - 3.6
MPU-6000	Accelerometer Gyroscope	I^2C, SPI	2.375 - 3.46

capacitance will be measured, processed, corresponding to a particular acceleration value.

There are a lot of options to be considered since each accelerometer can differ on the range, sensitivity, power, and communication interface. The Table 3.4 shows a comparison between some devices.

Table 3.4: Different accelerometer devices

Device	Axis	Range	Sensitivity [LSB/g]	Operating Voltage [V]	Communication Interface
ADXL362	X, Y, Z	$\pm 2g, 4g, 8g$	$1\ (\pm 2g) - 4(\pm 8g)$	1.6 - 3.5	SPI
BMA423	X, Y, Z	$\pm 2g, 4g, 8g, 16g$	$1024\ (\pm 2g) - 128(\pm 16g)$	1.62 - 3.6	I^2C, SPI
MMA8451QR1	X, Y, Z	$\pm 2g, 4g, 8g$	$4096\ (\pm 2g) - 1024(\pm 8g)$	1.95 - 3.6	I^2C
BMA253	X, Y, Z	$\pm 2g, 4g, 8g, 16g$	$1024\ (\pm 2g) - 128(\pm 16g)$	1.2 - 3.6	I^2C, SPI
LIS2DHTR	X, Y, Z	$\pm 2g, 4g, 8g, 16g$	$1000\ (\pm 2g) - 83(\pm 16g)$	1.71 - 3.6	I^2C, SPI

Alternatively, there are inertial measurement units (IMU), which are sensors that incorporate accelerometers and gyroscopes, and possibly magnetometers and temperature sensors, aimed at movement monitoring, motion tracking, and healthcare. By incorporating these capabilities into a single IC package, it results in great flexibility and minimum discomfort for the user. A comparison between different devices is made in Table 3.5, but it's important to note that these devices are more expensive than accelerometers, and understandably so.

3.2 System on Chip

The choice of the SoC must consider certain aspects such as the processing capacity, the storage capacity, communication interfaces, energy consumption, and also, the number of pins available. For this system, the nRF52832 was the chosen microprocessor. The nRF52832 comes in two different packages: **QFN48** (6 × 6 mm), in Fig 3.10a, and **WLCSP** (3.0 × 3.2 mm), in Fig 3.10b [103].

(a) QFN48 Package (b) WLCSP Package

Figure 3.10: nRF52832 [22]

The nRF52832, developed by Nordic Semiconductor, is built around 32-bit ARM Cortex-M4 processor with a float point unit at 64 MHz. It scores 215 in Embedded Microprocessor Benchmark Consortium (EEMBC) CoreMark, which is a benchmark that measures the performance of MCUs and CPUs. It has an energy consumption of 58 μA/MHz and 51.6 μA/MHz, running from flash memory and RAM, respectively. It also contains a data watchpoint and trace (DWT), which is used to count execution cycles, embedded trace macrocell (ETM), which is useful for hardware tracing and debugging, and instrumentation trace macrocell (ITM) support, for real-time debugging. Additionally, it also supports serial wire debug (SWD). Memory-wise it has two different options, either 512 kB of flash memory and 64 kB of RAM or 256 kB of flash memory and 32 kB of RAM [103].

This chip also has a 2.4 GHz transceiver built-in, allowing for the implementation of protocols such as BLE, ANT, and other 2.4 GHz proprietary protocols. For BLE specifically, it has -96dBm sensitivity, and supports 1 Mbps and 2 Mbps data rates. Furthermore, it has a 5.3 mA and 5.4 mA peak current in TX and RX, respectively, a single-pin antenna interface, and a received signal strength indicator (RSSI) with 1 dB of resolution [103].

Another important aspect of this device is its low-power consumption and flexible power management. Its supply voltage can vary between 1.7V and 3.6V, having an internal low-dropout (LDO) and DC/DC regulator. At 3V, the nRF52832 only draws 0.3 μA in OFF mode, 0.7 μA in OFF mode with RAM retention, and 1.9 μA in ON mode with no RAM retention [103].

It's also a device full of peripherals, as can be seen in Table 3.6. Moreover, it has EasyDMA, which is a module that provides easy-to-use direct memory access, digital microphone interface, temperature sensor, quadrature decoder (QDEC), programmable peripheral interconnect (PPI), and type 2 near field communication (NFC- A) tag [103].

Table 3.6: nRF52832 peripherals

Peripheral	Description
ADC	12-bit, 200 ksps 8 channels
Comparator	64 level 15 level low-power w/ wakeup
I/O	32
PWM	3x 4-channel
Timer	5x 32-bit
I²C	2
SPI	3
I²S	Yes
UART	Yes, w/ Clear-to-send (CTS) and Request-to-send (RTS)
RTC	3

3.3 Bluetooth Low Energy

The requirements for this device are that it can transmit data to an external unit wirelessly and with the least power consumption possible. There are a few alternatives, as mentioned previously, but the protocol chosen for this system was BLE. A brief explanation of the protocol will be made since the study of the intrinsic characteristics are beyond the scope of this project.

Bluetooth Low Energy (BLE), developed by the Bluetooth Special Interest Group (Bluetooth SIG), appeared in version 4.0 of Bluetooth and was designed to improve the performance in energy consumption of mobile devices. Introduced in 2010, the standard was adopted with impressive speed due to its low-power consumption, since a BLE device remains in sleep most of the time, only awaking when it needs to perform transmission. It has a peak current of 15 mA, but the average is 1 μA. Some of its characteristics are presented in Table 3.7 [104, 105].

Table 3.7: BLE characteristics

Technical Specification	BLE
Frequency	2400 - 2483.5 MHz
Range	<100 m
Data Rate	125kbit/s - 1Mbit/s - 2Mbit/s
Application Throughput	0.27 - 1.35 Mbit/s
Modulation	Frequency Hopping
Robustness	Adaptive Frequency Hopping, 24-bit CRC, etc.
Time to send data	>3 ms
Latency	6 ms
Power Consumption	0.01 - 0.50 W
Security	128-bit AES in CCM mode
Modulation	Gaussian Frequency-Shift Keying (GFSK)

Although BLE descends from the Bluetooth Classic, they are not compatible with each other. A device with BLE capabilities is said to be single-mode or smart, and a device with Bluetooth Classic and BLE capabilities is said to be dual-mode or smart-ready [104, 105].

The protocol stack of the Bluetooth Low Energy is divided between **Controller** and **Host**. The **Controller** is responsible for the actual transmission of data, packet reception and transmission, and delivery control. The **Host**, on the other hand, is responsible for the interoperability between application and peer devices. The protocol stack is defined in Fig 3.11 [104, 105].

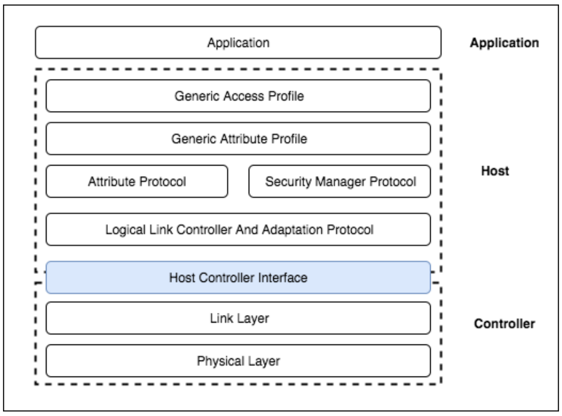


Figure 3.11: BLE Architecture [23]

The BLE is divided by 40 transmission channels and with 2 MHz of spacing. There are two types of channels [104, 105]:

- *advertising channels*, that occupy 3 of the 40 available channels and are used to search for devices, establish a connection, and broadcast transmission.
- *data channels*, that occupy the remaining channels and are used for bi-directional communication between devices.

As it is a protocol that uses the same frequency band as Wi-Fi (most common protocol), there may be collisions between packets. For this purpose, adaptive frequency hopping (AFH) is used, which analyzes the channels, excluding those already used, and tuning master and slave [104, 105].

Link Layer controls the connection status of a device, which can be defined through 5 states [104, 105]:

- *Standby*: the device is on hold, with no packet reception or transmission.
- *Advertising*: a device that is in advertiser mode, sends packets through all advertising channels, and can respond to requests from other devices.
- *Scanning*: a device in scanner mode, performs the monitoring of advertising channels, being able to place additional orders to other devices.

- *Initiating*: a device in initiator mode, tries to make a connection to an advertiser, requesting the start of a connection.
- *Connection*: when the connection is established, both devices enter this state, exchanging information through data channels.

The flow of a communication cycle is shown in Fig 3.12.

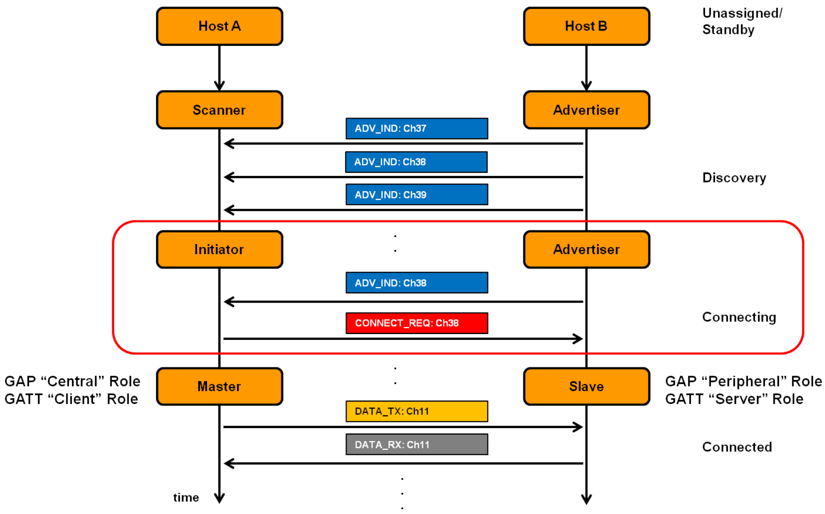


Figure 3.12: Diagram of a typical BLE connection [24]

When a connection is established, devices assume a certain role [104, 105]:

- **Masters**: are the devices that are in constant search for information packages from slave devices. They have higher energy consumption and, therefore, must be taken over by a device without restrictions at that level.
- **Slaves**: these are advertising devices and, as a rule, have limitations in terms of energy consumption.

The slave, as a rule, remains in sleep mode, while the master is responsible for supervising the connections, waking the slaves whenever necessary, through the Time Division Multiple Access (TDMA) scheme [104, 105].

3.4 Software

The **nRF5 SDK** is distributed as a .zip archive, which gives the user freedom to choose its favorite toolchain and provides a broad selection of drivers, libraries, examples, SoftDevices, and proprietary radio protocols. Additionally, Nordic also provides documentation for every piece of code. This acts as the Hardware Abstraction Layer (HAL) which allows the development of the application without the need of direct manipulation on the peripheral registers. This level of abstraction can accelerate the development of an application as well as giving portability to the code if the user decides to switch platform (in the Nordic family). Furthermore, this can be distributed commercially for free, facilitating the launch of an application [106].

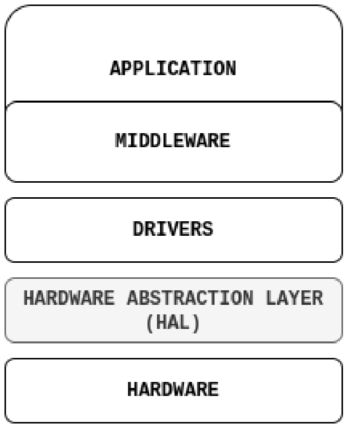


Figure 3.13: Firmware Diagram

A **SoftDevice**, also provided by Nordic Semiconductor, implements the BLE protocol stack (in this case) and deals with radio events. It's distributed as pre-compiled binary files and is located in a specific area of memory. The user can interface with the softdevice on the application code. This **SoftDevice** is programmed separately from the application, which means that there is no need to recompile it each time there is a compilation of the application. This piece of firmware is also pre-certified by the Bluetooth SIG.

This allows the device to act as central, observer, peripheral and transmitter, it has extended support for advertising, it enables the configuration of the connection properties, it gives link layer support for LE 1M PHY e LE 2M PHY, it offers GATT and GAP APIs, it isolates the application memory and the protocol stack, and provides a thread-safe API, among others.

There are several **SoftDevices**, each referring to a determined protocol and associated chip [107].

Chapter 4

System Development

The development of an application encompasses several steps, from its conception to its implementation. In this case, the project will be segmented into several stages that, when connected, will form the final system.

4.1 Sleep position

As already mentioned, one of the information to be determined is the patient's sleep position. The position estimation can be achieved by collecting readings from the axis of an accelerometer. These readings will be used to determine the orientation of the device and, consequently, of the patient. The device used will be the MPU-6050.

MPU-6050 Overview

The MPU6050 is an IMU that integrates a 3-axis accelerometer, 3-axis gyroscope, and Digital Motion Processing (DMP). This device has three 16-bit ADCs for the accelerometer and three 16-bit ADCs for the gyroscope. Also, it is possible to define the dynamic range of the readings, which will influence its accuracy [108].

Additionally, the MPU-6050 has a temperature sensor and an on-chip oscillator. Communication with the device and, consecutively, with the available records, is carried out through I^2C at 400kHz [108].

Registers

As mentioned, the interaction with the MPU-6050 is achieved through registers. There are plenty of records available, each with its function, so it would not be practical to describe them all, so the most important ones will be presented [25].

In I²C, there is always a master and a slave. The master will be the one who controls the communication, while the slave responds when addressed by the master. Therefore, each slave device has a unique address in that network. The MPU-6050 can take on two different addresses (0x68, 0x69), allowing the existence of two MPUs on the same network [25].

Before describing the records to be used, note that all this information can be accessed from the device's register map and descriptions [25].

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE _RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

Figure 4.1: Power Management 1 Register [25]

The first register refers to the first part of power management and clock source. As shown in the Figure 4.1, the register consists of 5 different fields [25]:

- *DEVICE_RESET* - when set (1), it resets the device. Returns to 0 when the reset is complete.
- *SLEEP* - when set, the device goes into sleep mode.
- *CYCLE* - when set and SLEEP is not, the MPU-6050 is configured to switch between sleep mode and take a sample at a certain frequency.
- *TEMP_DIS* - when set to 1, disables the temperature sensor.
- *CLKSEL* - this field defines the clock source for the device. Upon power-up, the device defaults to the internal oscillator. If the clock source is different, this field must be changed.

The second part of power management, Figure 4.2 has two functions: setting the sampling frequency when in Accelerometer Only Low Power Mode and disable accelerometer and gyroscope axes [25].

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6C	108	LP_WAKE_CTRL[1:0]		STBY_XA	STBY_YA	STBY_ZA	STBY_XG	STBY_YG	STBY_ZG

Figure 4.2: Power Management 2 Register [25]

The Only Low Accelerometer Power Mode can be interesting to analyze, in the sense that it allows minimizing the energy consumption of the device (something that is important in this context), collecting the necessary information to estimate the orientation of the device. Since the objective is not so much to characterize the movement of the device, but rather its orientation at rest, the data from the accelerometer may be enough [25].

This mode can be achieved by activating the *CYCLE* and *TEMP_DIS* bits and deactivating the SLEEP bit. It is also necessary to deactivate all axes belonging to the gyroscope (*STBY_XG*, *STBY_YG*, *STBY_ZG*) by setting them to 1. The sampling frequency can be defined via the *LP_WAKE_CTRL* field, which can vary between 1.25 Hz, 5 Hz, 20 Hz, and 40 Hz [25].

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

Figure 4.3: Accelerometer Configuration Register [25]

Then, the register regarding the accelerometer configuration (Figure 4.3). This register, besides providing fields that allow for the device to perform self-testing, allows configuring the full-scale accelerometer range, through the *AFS_SEL* field. This dynamic range can be defined as ± 2 g, ± 4 g, ± 8 g, and ± 16 g [25].

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Figure 4.4: Gyroscope Configuration Register [25]

A similar configuration can be made for the gyroscope, through the dedicated register (Fig 4.4). It also offers the opportunity for the device to perform self-testing, and also to define the dynamic range of the reading, between ±250 $^\circ/s$, ±500 $^\circ/s$, ±1000 $^\circ/s$, and ±2000 $^\circ/s$, through the *FS_SEL* field [25].

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

Figure 4.5: Accelerometer Readings Register [25]

To obtain the values for the accelerometer, these will be contained in six registers (Figure 4.5). Note that each axis of the accelerometer corresponds to a

16-bit value, so it is divided into two registers, relative to the MSBs and LSBs. Thus, to read the entire axis of the accelerometer, the read-only registers seen in the register will be read [25].

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
43	67	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT[7:0]							

Figure 4.6: Gyroscope Readings Register [25]

The same can be said for the gyroscope. The value for each axis is also 16-bit and will be stored with two 8-bit registers, resulting in six registers for the gyroscope data (Figure 4.6). These values are stored in the read-only registers [25].

Accelerometer Data

Moving on to implementation, the first step is to initialize the Two-Wire Interface (TWI) peripheral, which is compatible with I²C. Using the driver present in the SDK, the peripheral starts by specifying the pins referring to I²C (SDA and SCL), which will allow the exchange of information between master (microprocessor) and slave (MPU-6050), and it is still necessary to specify the operating frequency, which in this case will be 400 kHz.

Once the TWI peripheral is configured, it is possible to start communicating with the MPU-6050. Since the goal, at first, is to be able to understand whether it is possible or not to obtain data from the MPU, its configuration will be simple. We will try to obtain the value of all axes of the accelerometer and gyroscope, so the register 0x6B (power management and clock source) will be set to 0x00, which will allow the MPU to run freely, using its internal oscillator.

Then, both the accelerometer and gyroscope need to be configured. Both the gyroscope configuration register (0x1B) and accelerometer configuration register (0x1C) will be set to 0x00, selecting the full-scale range of $\pm$ 250 $^{\circ}$/s and $\pm$ 2 g, respectively.

At this point, the MPU-6050 is configured and recording data from the accelerometer and gyroscope. This information is stored on the registers already presented. For this demonstration, these values are read every 500 ms. First, the accelerometer is read by specifying the register 0x3B, which corresponds to *ACCEL_XOUT*, and proceeds to read six bytes. When the data is received, the same is done for the gyroscope, in which we specify the register 0x43, corresponding to *GYRO_XOUT*, and read the next six bytes.

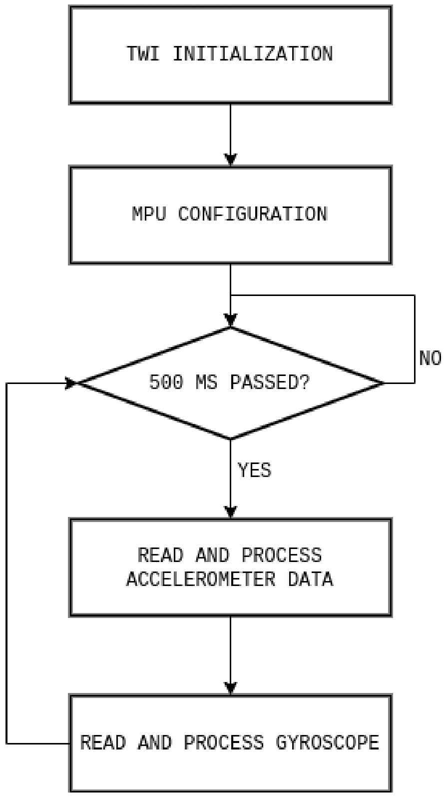


Figure 4.7: Process for reading MPU-6050 data

This process will allow obtaining raw data from the MPU-6050 however, it is necessary to transform this data into something perceptible. When, in the configuration of the accelerometer and gyroscope, the full-scale range is defined, there is another variable that is changed accordingly, which is the sensitivity.

In the case of the accelerometer, at a full-scale range of $\pm$ 2 g, corresponds to a sensitivity of 16384 LSB/g, so the raw data will have to be divided by this value to obtain the value of the axis in g. For the gyroscope, the process is similar, however, for a full-scale range of $\pm$ 250 $^{\circ}$/s, the sensitivity is 131 LSB/$^{\circ}$/s. The values obtained can be seen in the Figure 4.8.

```
ACELEROMETRO -> X: (-0.019775390625), Y: (0.0263671875), Z: (1.01611328125)
GIROSCOPIO -> X: (6.183206106870229), Y: (2.1374045801526718), Z: (-0.022900763358778626)
ACELEROMETRO -> X: (-0.031494140625), Y: (0.02587890625), Z: (1.012939453125)
GIROSCOPIO -> X: (6.129770992366412), Y: (2.2977099236641223), Z: (-0.16793893129770993)
ACELEROMETRO -> X: (-0.018798828125), Y: (0.022216796875), Z: (1.01904296875)
GIROSCOPIO -> X: (6.061068702290076), Y: (2.1755725190839694), Z: (-0.2595419847328244)
```

Figure 4.8: Data retrieved from the MPU-6050

Device Orientation

After collecting the sensor data, the next step will be to be able to estimate the orientation of the device. This is a common technique on smartphones, in which the accelerometer is used to calculate the orientation of the device. The goal will be to try to apply the same process to this system.

For this purpose, the rotations in roll, pitch, and yaw will be calculated, since these reflect changes in the orientation of the device in the three axes (x, y, and z, respectively). The in-depth explanation of these calculations is beyond the scope of this article and can be accessed in [109].

Then, the value of the axes of the accelerometer can be given by:

$$G_p = \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = R(g - a_r) \tag{4.1}$$

where **R** corresponds to the rotation matrix of the device in relation to the earth's coordinate frame, **g** corresponds to the earth's gravitational field, and $\mathbf{a_r}$ is the linear acceleration of the device [109].

Then, it is assumed that initially the device's z-axis is aligned with the earth's gravitational field and that the device is at rest, that is, $a_\mathbf{r}$ = 0, which results in:

$$G_p = \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = Rg = R \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{4.2}$$

Applying the rotation matrices, it is possible to transform a vector into a roll, pitch, and yaw angles. The order in which these rotation matrices are applied can vary. The roll, pitch, and yaw rotation matrices are given by:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{pmatrix} \tag{4.3}$$

$$R_y(\phi) = \begin{pmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{pmatrix} \tag{4.4}$$

$$R_z(\phi) = \begin{pmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.5}$$

The result of part of these variations can be discarded since the output of the accelerometer will be a function of the three angles of rotation. The rest is a function of roll and pitch, and do not depend on the yaw. Note that the accelerometers are insensitive to their rotation around the gravitational field vector (in this case, the z-axis) [109].

After all manipulations and simplifications, it is possible to determine the row and pitch angles through:

$$tan\theta_{yxz} = \frac{-G_{px}}{sign(G_{pz})\sqrt{G_{pz}^2 + \mu G_{py}^2}} \tag{4.6}$$

$$tan\phi_{yxz} = \frac{G_{py}}{\sqrt{G_{pz}^2 + G_{px}^2}} \tag{4.7}$$

Furthermore, the position estimation will only be possible if the device is at rest, which can be checked through the following equation. What this means, is that the device is not suffering any "external" acceleration due to the body movement, and only is affected by gravity itself (1 g).

$$\sqrt{G_{px}^2 + G_{py}^2 + G_{pz}^2} = 1 \tag{4.8}$$

Let's first analyze the situations presented in Fig 4.9, in which the gravitational acceleration directly affects one axis of the MPU. So, if the MPU is at rest with the z-axis pointing downwards [**a)**], that will mean that the measurement will return something close to G_{pz} = 1, G_{px} = 0, and G_{py} = 0. Applying these values to the previous equations will result in a θ_{yxz} and ϕ_{yxz} values of 0º or 180º, meaning that the signal of Gz will determine what solution prevails. If G_{pz} = 1, then those angles should be 0º, and if G_{pz} = -1, then the result should be 180º.

In the second case, where the gravitational acceleration acts purely on the y-axis [**b)**], the measurement will be approximately G_{pz} = 0, G_{px} = 0, and G_{py} = 1. Again, by applying these values to the equations will result in a θ_{yxz} of either 0º or 180º, and a ϕ_{yxz} of either +90º or -90º. In the pitch (ϕ_{yxz}) equation, the solution is determined by the signal of G_{py}, meaning that G_{py} = 1 correlates to a pitch of +90º, and G_{py} = -1 to -90º.

In the last case, the gravitational acceleration points directly to the x-axis of the MPU [**c)**], resulting in a measurement of G_{pz} = 0, G_{px} = 1, and G_{py} = 0. The process is very similar to the second case, but instead is the θ_{yxz} that's either +90º or -90º, and the ϕ_{yxz} either 0º or 180º. Again, to reach the right solution, the signal of G_{px} is used. If G_{px} = 1, then the roll value will be -90º, and if G_{px} = -1 the roll value is +90º.

Lastly, it's safe to say that a change in the x-axis results in a change of the roll value, and a change in the y-axis results in a change of the pitch value.

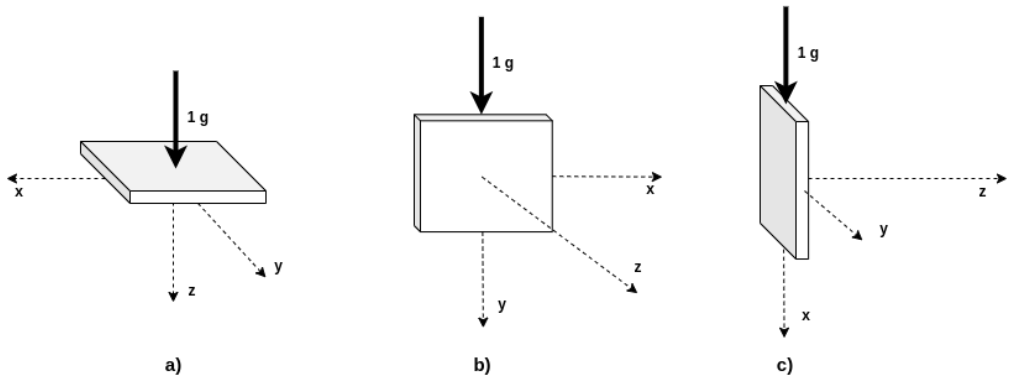


Figure 4.9: Different device positions

The next step is to correlate the device rotation to the patient motion. As mentioned, the device will be located on the patient's mouth, so by determining the device orientation, the patient's head orientation can also be estimated. The Fig 4.10 makes it easy to visualize that translation of motion. Given the patient's head orientation, it's possible to predict the patient's sleep position.

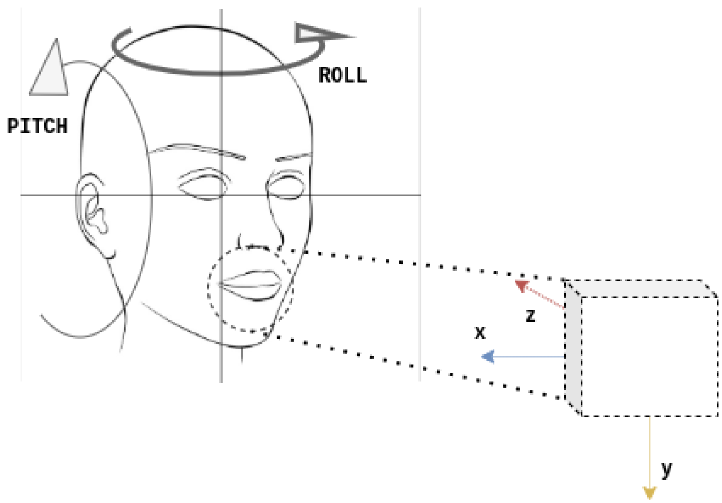


Figure 4.10: Example of device orientation on patient

The head's orientation is not enough to assess the definite patient's sleep position, but it can give a general idea about it. This analysis will take into consideration the standard anatomical positions, which are terms specially used in medicine to describe the patient's position.

A person in a supine position is aligned horizontally, with its back (dorsal side) down, its ventral side up. The patient's face is also facing up, according to Fig 4.11. Some common sleeping positions related to this are "Starfish" [**a)**] and "Soldier" [**b)**].

In this case, the gravitational acceleration will act mostly on the z-axis, and with a positive influence because of the way MPU-6050 will be mounted on the device. So, if the roll and pitch values are small ($G_{pz} \gg G_{px}$ and $G_{pz} \gg G_{py}$), the most probable position of the patient is supine.

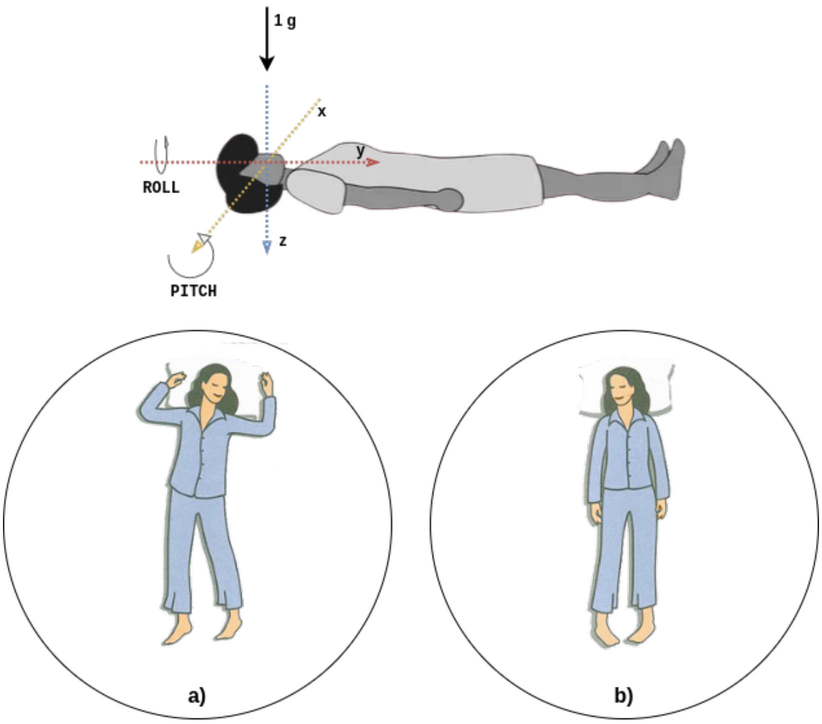


Figure 4.11: Supine position

In the prone position, the patient is also laying horizontally, but with its back facing up, and with its ventral side down. In this situation, the patient's head is also facing down. The "Freefall" [**a)**] is the closest sleeping position, as seen in Fig 4.12.

The gravitational acceleration is expected to be distributed mostly between the z-axis and x-axis. As the patient's head is supposed to be facing downwards, the z-axis should be affected negatively ($G_{pz} < 0$). The G_{px} will depend on which side the patient's head is oriented. The degree of certainty about this position will increase as G_{pz} decreases, that is, $G_{pz} \to -1$. As $G_{pz} \to 0$, the number of possible sleeping positions increases, making it harder to estimate.

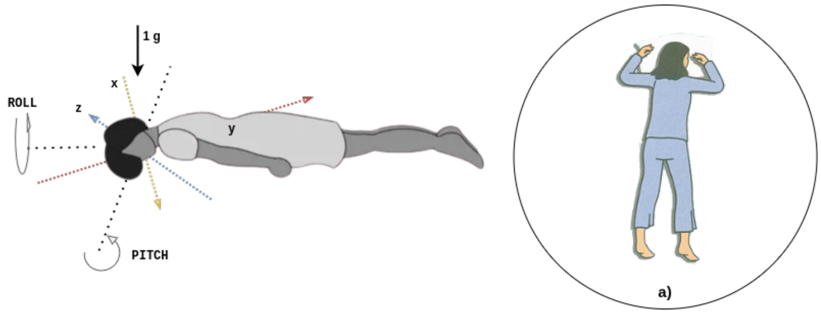


Figure 4.12: Prone position

The left and right lateral recumbent positions indicate that the patient is laying on either their right or left side. There are a few sleeping positions associated with these anatomic positions: the " Yearner" [**a)**], the "Fetal" [**b)**], and the "Log" [**c)**], as shown in Fig 4.13. The problem is that, although they differ from each other, the head orientation is practically the same.

In this case, the gravitational acceleration should affect mainly the x-axis. The type of effect will determine to which side is the patient laying down. If $G_{px} \approx 1$ (positive effect), then the patient is laying on their right, and if $G_{px} \approx$ -1. (negative effect), then the patient is laying on their left. The θ_{yxz} should be approximately -90º or +90º.

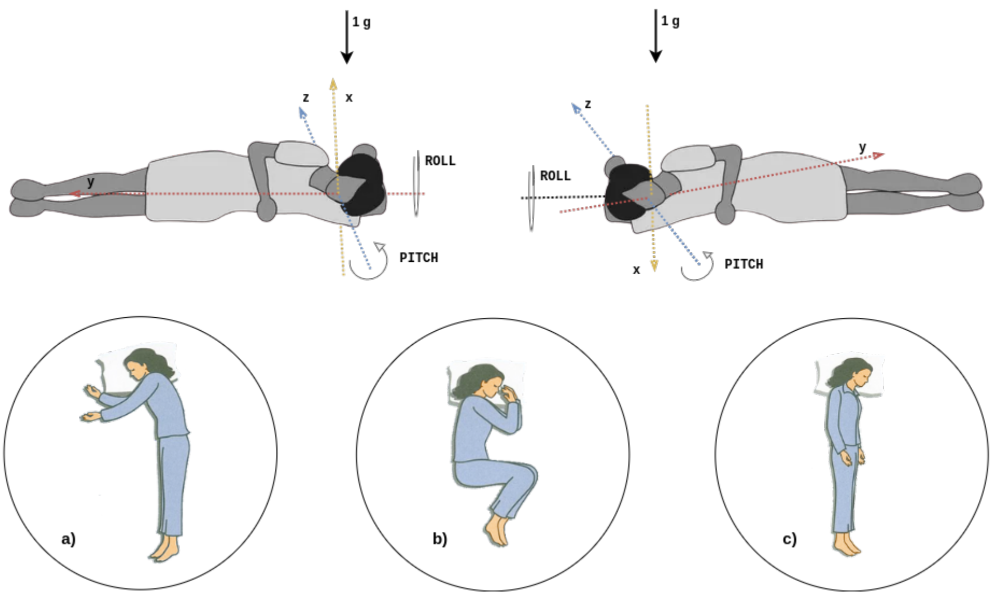


Figure 4.13: Left and right lateral recumbent position

Let's study a practical example. In the first case, the device is at rest, with the device's coordinate frame aligned with the earth's coordinate frame (except the z-axis, which points to the opposite side). The gravitational acceleration acts purely on the z-axis, so the values of the accelerometer should be: $G_z = 1$, $G_x = 0$, $G_y = 0$. By applying these values to the equations, we get a $\theta_{yxz} = 0°$ and $\phi_{yxz} = 0°$.

$$\theta_{yxz} = atan(\frac{0}{\sqrt{1^2 + 0.1 * 0^2}}) = atan(0) = 0° \tag{4.9}$$

$$\phi_{yxz} = atan(\frac{0}{\sqrt{1^2 + 0^2}}) = atan(0) = 0° \tag{4.10}$$

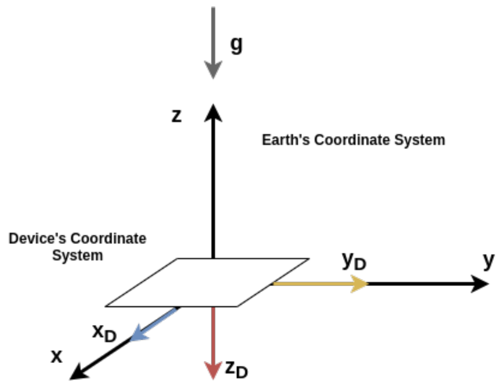


Figure 4.14: Device at rest

In the next example, we try to determine the pitch angle. In this situation, we can see that the x-axis of the earth's coordinate frame and the device's coordinate frame stay aligned. In the left image, the gravitational acceleration is divided between the z-axis and the y-axis of the device. Assuming that $G_z = 0.9$, $G_x = 0$, $G_y = 0.436$, we get a $\theta_{yxz} = 0°$ and $\phi_{yxz} = 23.56°$. In the right image, the gravitational accelerations act purely on the y-axis so, assuming that $G_z = 0$, $G_x = 0$, $G_y = 1$, we get $\theta_{yxz} = 0°$ and $\phi_{yxz} = 90°$.

For the left image:

$$\theta_{yxz} = atan(\frac{0}{\sqrt{0.9^2 + 0.1 * 0.436^2}}) = atan(0) = 0° \tag{4.11}$$

$$\phi_{yxz} = atan(\frac{0.436}{\sqrt{1^2 + 0^2}}) = atan(0.436) = 23.56° \tag{4.12}$$

For the right image:

$$\theta_{yxz} = atan(\frac{0}{\sqrt{0^2 + 0.1 * 1^2}}) = atan(0) = 0° \tag{4.13}$$

$$\phi_{yxz} = atan(\frac{1}{\sqrt{0^2 + 0^2}}) = atan(\frac{1}{0}) \tag{4.14}$$

$$\lim_{x \to 90^\circ} tan(x) = \infty \tag{4.15}$$

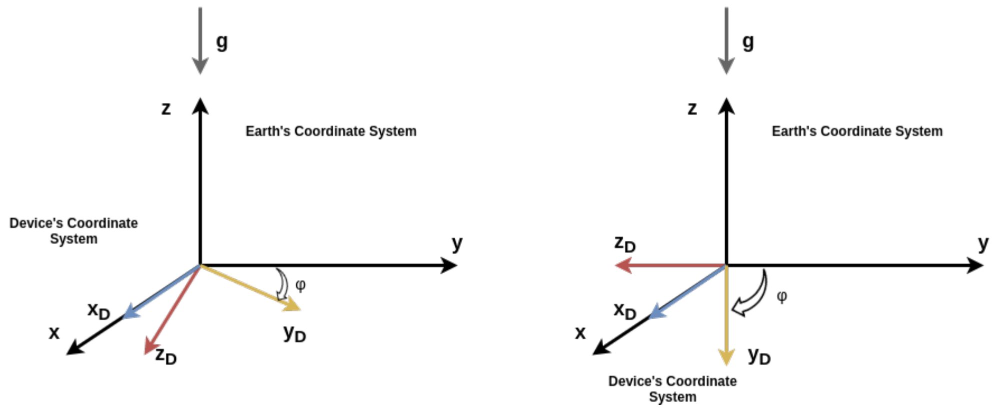


Figure 4.15: Pitch estimation

Then, we apply the same principle for the roll angle. In this case, the y-axis of the device's coordinate frame will stay aligned with the earth's coordinate frame. In the left image, the gravitational acceleration is shared between the x-axis and the z-axis of the device. Assuming that $G_z = 0.75$, $G_x = 0.661$, $G_y = 0$, we get a $\theta_{yxz} = -41.39^\circ$ and $\phi_{yxz} = 0^\circ$. In the right image, the gravitational accelerations act purely on the y-axis so, assuming that $G_z = 0$, $G_x = 1$, $G_y = 0$, we get $\theta_{yxz} = 90^\circ$ and $\phi_{yxz} = 0^\circ$.

For the left image:

$$\theta_{yxz} = atan(\frac{-0.661}{\sqrt{0.75^2 + 0.1 * 0^2}}) = atan(0.881) = -41.39^\circ \tag{4.16}$$

$$\phi_{yxz} = atan(\frac{0}{\sqrt{0^2 + 1^2}}) = atan(0) = 0^\circ \tag{4.17}$$

For the right image:

$$\theta_{yxz} = atan(\frac{-0.661}{\sqrt{0^2 + 0.1 * 0^2}}) = atan(\frac{-0.661}{0}) \tag{4.18}$$

$$\lim_{x \to 90^\circ} tan(x) = \infty \tag{4.19}$$

$$\phi_{yxz} = atan(\frac{0}{\sqrt{0^2 + 01^2}}) = atan(0) = 0^\circ \tag{4.20}$$

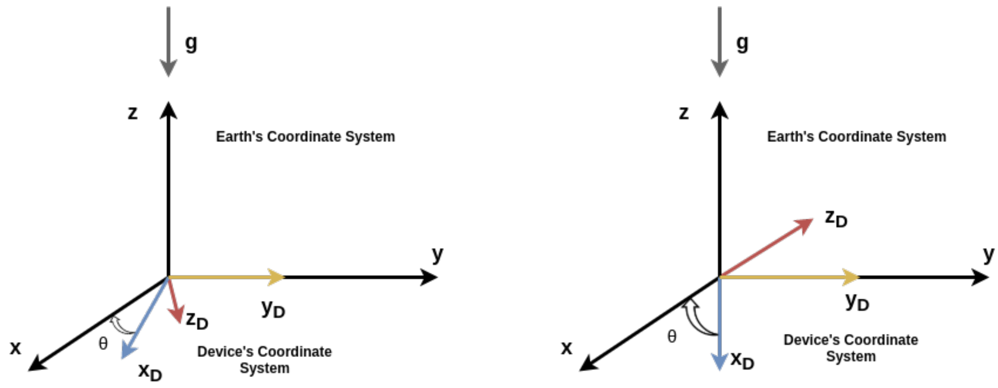


Figure 4.16: Roll estimation

For the last example, we misalign all the axis, so the gravitational acceleration acts on all of them. Assuming that $G_z = 0.75$, $G_x = 0.33$, $G_y = -0.573$, we get $\theta_{yxz} = -24.39°$ and $\phi_{yxz} = -34.97°$.

$$\theta_{yxz} = atan(\frac{-0.33}{\sqrt{0.75^2 + 0.1 * (-0.573)^2}}) = atan(-0.453) = -24.39° \quad (4.21)$$

$$\phi_{yxz} = atan(\frac{-0.573}{\sqrt{0.75^2 + 0.33^2}}) = atan(-0.699) = -34.97° \quad (4.22)$$

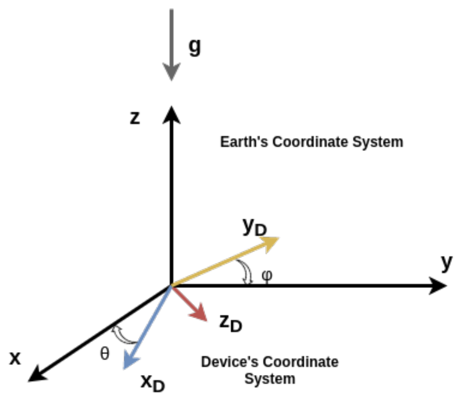


Figure 4.17: Estimation of both angles

Additionally, we could use a combination of the accelerometer and gyroscope to determine the roll and pitch angles. The gyroscope measures the angular rate around one axis, so by summing the product of the angular rate and the sampling interval to the previous angle estimate, the angle can be estimated. The problem is that any bias or noise present in the readings will be summed

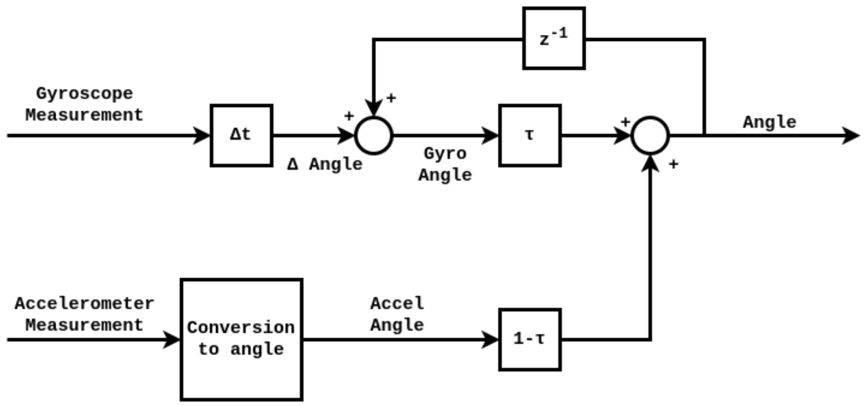


Figure 4.18: Block diagram of the complementary filter

up, affecting the estimate. Briefly, the gyroscope is accurate in detecting small variations in short periods, but, in the long term, the difference between the real and the estimated angle is going to be intolerable. As for the accelerometer, it is not reliable for short periods since it detects every acceleration, but it is stable in the long term because gravity is always pointing down, and, knowing down relative to the IMU reference frame, it can determine the angle.

The complementary filter merges the readings of two or more sensors and, more specifically, the accelerometer and gyroscope, to produce an output. The accelerometer would compensate for the long term variations of the gyroscope, and the gyroscope would allow the measurement of short term variations. The block diagram is shown in Fig 4.18.

The implementation of a discrete complementary filter is given by:

$$\angle = (1-\tau) \cdot \angle_{ACCEL} + \tau \cdot (G \cdot \Delta t + \angle \cdot z^{-1}) \tag{4.23}$$

$$\angle - \angle \cdot \tau \cdot z^{-1} = (1-\tau) \cdot \angle_{ACCEL} + \tau \cdot G \cdot \Delta t \tag{4.24}$$

$$\angle = ((1-\tau) \cdot \angle_{ACCEL} + \tau \cdot G \cdot \Delta t) \cdot \frac{1}{1-\tau \cdot z^{-1}} \tag{4.25}$$

where $\angle$ is the estimated angle, $\angle_{ACCEL}$ is the accelerometer angle, Δt is the sampling interval, G is the gyroscope measurement, and τ is the ratio of one signal against the other. Additionaly, the complementary also provides a discrete low-pass filter. By lowering τ we are allowing the accelerometer to have more preponderance, and vice-versa.

4.2 SAADC

The Successive-Approximation Analog-to-Digital Converter (SAADC) peripheral is what allows the processor to access analog values by turning them into a digital representation.

The technique behind the successive-approximation is to try various output values from the DAC and compare them to the analog input present at the comparator, as demonstrated in Fig 4.19. The process begins by setting the MSB to 1, and the rest of the bits to 0. Then, if $V_{DAC} > V_{IN}$, the MSB is set to 0, and if $V_{DAC} < V_{IN}$, the bit remains as 1. Next, the next bit is set to 1, and the same process repeats *n*-times. This *n* is the resolution of the ADC. A SAADC has a BEGIN CONVERSION input and a CONVERSION DONE output [110].

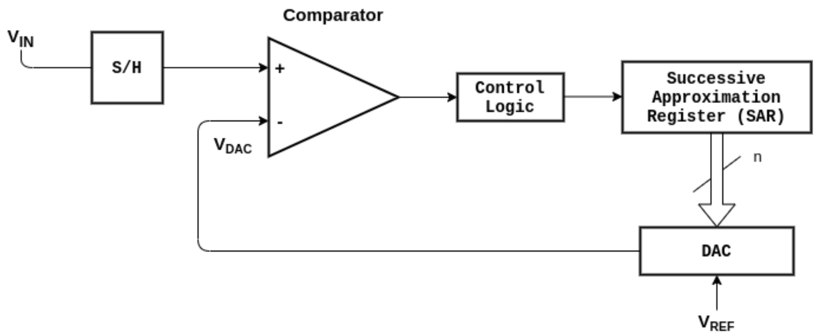


Figure 4.19: Successive-approximation ADC

The Fig 4.20 shows a clear example of a 4-bit successive-approximation and the converging to the final value. It's a binary search, in which the first guess is half the full scale.

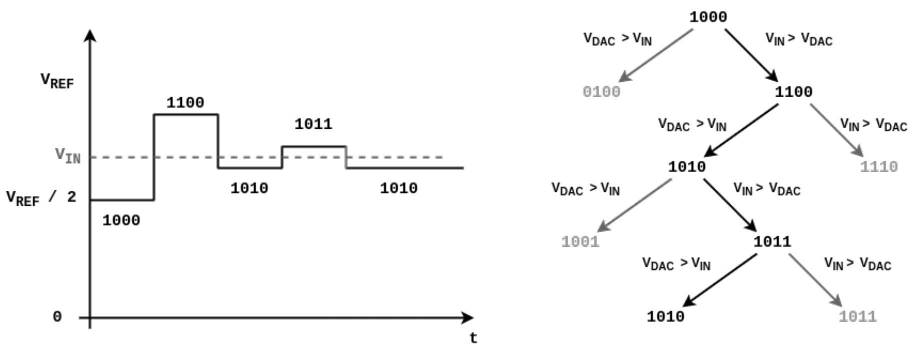


Figure 4.20: Example of a 4-bit successive-approximation

4.2.1 SAADC Peripheral

The SAADC peripheral has a configurable resolution, ranging from 8/10/12-bit, and, with oversampling, it can reach a 14-bit resolution. It has eight input channels and four different operation modes. In **one-shot mode**, the SAADC samples a single channel. In **continuous mode**, only one channel is enabled, and the sampling rate can be configured either by the internal timer in the ADC or through general-purpose timers (via PPI). With **oversampling**, an accumulator is used to average noise on the input channel. The ADC enters in **scan mode** when more than one channel is enabled. In this mode, a sample task will trigger one conversion for each enabled channel [111].

The ADC may be configured as single-ended input (the internal ADC ground is the same as the external ground) or differential input (use a different "reference" point). The ADC also uses Easy Direct Memory Access (EasyDMA) to store the results in RAM [111].

Two different references can be used by the ADC, either internal or V_{DD}. The input range for both references is given by the following equations:

$$InputRange = \frac{V_{DD}}{4} * Gain \tag{4.26}$$

$$InputRange = 0.6V * Gain \tag{4.27}$$

Furthermore, the acquisition time for the ADC is configurable. The Sample & Hold circuit is comprised of a switch and a capacitor. The acquisition time determines the duration the switch is closed, and therefore, how much time is the capacitor connected to the input [111].

Another feature of the ADC is event monitoring, which allows establishing limits on the analog input voltage. A high and low limit can be set and are independent in each channel. If these limits are surpassed, then an event is generated. The Fig 4.21 shows this process. On the topic of events, two main events are used for this application, including the event about SAADC conversion done, and the high limit surpassed. But more about this later [111].

Now that we understand the different configurations of the SAADC peripheral, let's define the best behavior for the system. As previously mentioned, two force sensors will be used, which correlates to the number of analog channels that need to be configured. Next, the operation mode must be defined, and, at this point, some compromises will take place. We know that the system needs to be low-power, which will influence the sampling frequency. It will also influence how the peripheral needs to be managed.

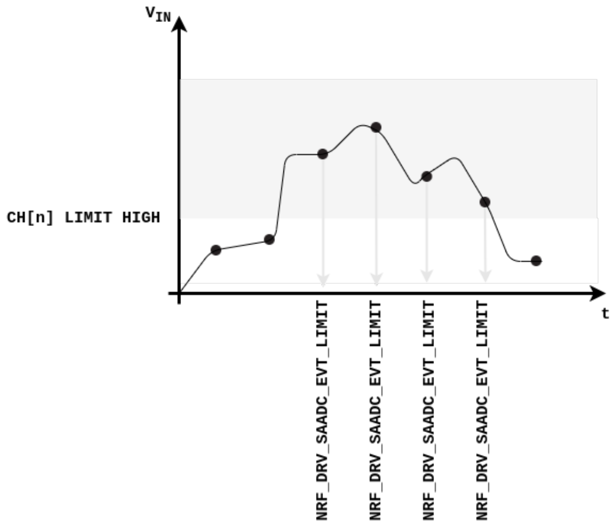


Figure 4.21: Limit Event Monitoring

In one-shot mode, both enabling and conversion of each channel are performed separately. As the sampling frequency of both channels is the same, this mode of operation is redundant when scan mode is available. In continuous mode, the SAADC peripheral is always enabled and, consequently, with constant power consumption, as we will see is due to a specific part. Finally, the scan mode is the most adequate choice because it allows the peripheral to be uninitialized between samples if interrupts are used. Alternatively, the PPI could be used to generate this sampling rate, allowing the CPU to sleep most of the time. The problem is that the SAADC would remain initialized, and so would be the EasyDMA, which would in itself draw around 1.5 mA. Given this, the low-power RTC is used to generate an interrupt each 140 ms, representing a sampling frequency of around 7 Hz. This frequency is not definite and can be adjusted based on our pretensions [111].

The SAADC can also use oversampling. Although ideally, one sample per channel should be enough, it makes the measurements susceptible to noise. Oversampling would act as a kind of filter, namely a moving average filter, in the sense that would average several samples and would minimize the amount of noise present in the signal. But there are disadvantages to this operation mode. The time that would take to terminate a sample task would increase, as the number of events and, subsequently, of interrupts. The following equations describe how much time would it take for a sample task to be done [111].

$$\sum_{n=1}^{M} CH[n] * t_{ACQ} + t_{CONV} \tag{4.28}$$

in which t_{ACQ} represents the acquisition time, and can it's defined in the application, t_{CONV} represents the time of conversion, which is 2 μs, and M that represents the number of enabled channels. If t_{ACQ} = 10 μs, which is a standard value, and 2 channels are enabled, then the total time would be of about 24 μs [111].

$$\sum_{n=1}^{M}(CH[n] * t_{ACQ} + t_{CONV}) * 2^{OVERSAMPLE} \tag{4.29}$$

in which t_{ACQ}, t_{CONV}, and M have the same meaning as the previous equation, and OVERSAMPLE represents the oversampling factor. The number of samples needed increases exponentially with increasing oversampling factor. Assuming a oversampling factor of 2x for both channels, that would result in 4 samples per channel. Applying the equation it would result in a total time of 96 μs. More tests are necessary to determine the best behavior for the system. For now, only one sample per channel will be assumed [111].

Briefly, the ADC peripheral will be configured with a 12-bit resolution, and the two channels will have an acquisition time of 10 μ s. The reference of the channels will be internal, which isolates them from any variations in the supply voltage, and with a gain of 1/5, the analog input being able to vary between 0 V - 3 V.

4.2.2 Signal Conditioning

The Flexiforce sensor, which is one of the sensors that will be used for this system, varies its resistance when force is applied. The typical graph of this variation is shown in 3.5 but, to measure this variation, a circuit needs to be designed.

The simplest way of transforming the variation is resistance to a voltage would be of a voltage divider. It would work fine, except for the fact that the sensor is not linear (in terms of resistance vs force applied). This non-linearity would need to be compensated in software.

In this system, another signal conditioning technique is used, namely a non-inverting op-amp circuit. It is one of the recommended circuits by Tekscan, which is the manufacture of the sensor [112].

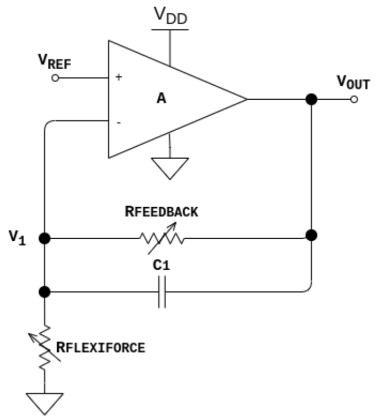


Figure 4.22: Signal Conditioning for Flexiforce

According to Tekscan, V_{REF} should be in the range of 0.25 V - 1.25 V, and it is a reference voltage, and will not change. Let's see how V_{OUT} will change with $R_{FLEXIFORCE}$. In this situation, $R_{FEEDBACK}$ and $R_{FLEXIFORCE}$ will form a simple voltage divider [112].

$$V_1 = \frac{R_{FLEXIFORCE}}{R_{FLEXIFORCE} + R_{FEEDBACK}} * V_{OUT} \tag{4.30}$$

We also know that both inputs of the op-amp are at the same potential:

$$V_{REF} = V_1 \tag{4.31}$$

The voltage gain, A(v), is given by:

$$A(v) = \frac{V_{OUT}}{V_{REF}} \tag{4.32}$$

With some modifications we are left with:

$$A(v) = 1 + \frac{R_{FEEDBACK}}{R_{FLEXIFORCE}} \tag{4.33}$$

Given that, V_{OUT} can be determined by:

$$V_{OUT} = \frac{R_{FLEXIFORCE} + R_{FEEDBACK}}{R_{FLEXIFORCE}} * V_{REF} \tag{4.34}$$

If we define V_{REF} as a fixed value, such as 0.5 V, $R_{FEEDBACK}$ can be determined by analysing the $R_{FLEXIFORCE}$. Note that lower V_{REF} allows for better dynamic range.

When the sensor is not subjected to any external force, the sensor has a $R_{FEEDBACK} > 6$ MΩ. If we make $R_{FEEDBACK}$ a value of two orders of magnitude lower, then $V_{OUT} \approx V_{REF}$.

As higher forces are applied, the $R_{FLEXIFORCE}$ decreases exponentially. For values close to the dynamic range the approximate value for $R_{FLEXIFORCE}$ is 20 kΩ. So, the goal is to make V_{OUT} close to the V_{DD}, which is 3 V. One thing to note is that every sensor is different, which makes $R_{FLEXIFORCE}$ unpredictable. For this reason, $R_{FEEDBACK}$ is calculated to allow the sensor to measure higher forces, and not saturate V_{OUT}.

For $R_{FLEXIFORCE}$ = 20 kΩ, V_{OUT} should be ~2.5V:

$$2.5 = \frac{20k\Omega + R_{FEEDBACK}}{20k\Omega} * 0.5 \tag{4.35}$$

Which results in a $R_{FEEDBACK}$ of 80 kΩ. The other aspect of this circuit is that it acts as an active low-pass filter, which is defined by the capacitor in parallel with the $R_{FEEDBACK}$.

Analysing the reactance of the capacitor, for low frequencies, X_C will be extremely high, hence the parallel impedance between X_C and $R_{FEEDBACK}$, is approximately $R_{FEEDBACK}$. In this situation, the voltage gain of the op-amp is given by 4.33. For high frequencies, X_C is really low, hence the parallel impedance is approximately X_C. In this situation, the voltage gain provided by the op-amp is 1, meaning that $V_{OUT} = V_{REF}$.

$$X_C = \frac{1}{2\pi * f * C} \tag{4.36}$$

The cut-off frequency of the active filter is given by:

$$f_c = \frac{1}{2\pi * R_{FEEDBACK} * C} \tag{4.37}$$

The value of the capacitor (C_1) needs to be tested to achieve the best performance for the system.

As a last note for this sensor, although the non-inverting op-amp circuit was used, the inverting op-amp circuit would be equally valid. The defining factor was that the inverting circuit required a dual-source supply, instead of the one-source supply required by the non-inverting circuit. The downside is that non-inverting circuit only as a dynamic range from $]V_{REF}; V_{DD}[$, instead of the $]0; V_{DD}[$ of the inverting circuit.

For the FSR, the signal conditioning will be very similar since the functioning of the sensors is also very close, only differing in the characteristics of the sensors. The circuit is based on the specified by Interlink Electronics, which is the manufacturer of the sensor, and is shown in Fig 4.23 [26].

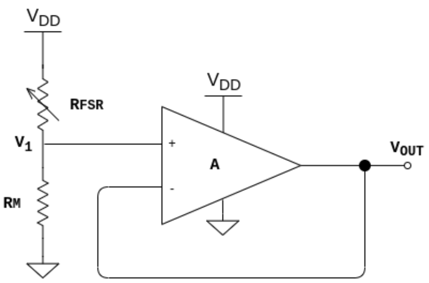


Figure 4.23: Signal Conditioning for FSR

Let's analyze how the system behaves for different values of R_{FSR}. Knowing that both input terminals of the op-amp have the same potential:

$$V_{OUT} = V_1 \tag{4.38}$$

The V_1 is determined by a voltage divider between R_M and R_{FSR}:

$$V_1 = V_{OUT} = V_{DD} * \frac{R_M}{R_M + R_{FSR}} \tag{4.39}$$

The value of R_M will change how the system's response to the force applied in the FSR. Several values were tested by the manufacture and their performance is shown in Fig 4.24 [26].

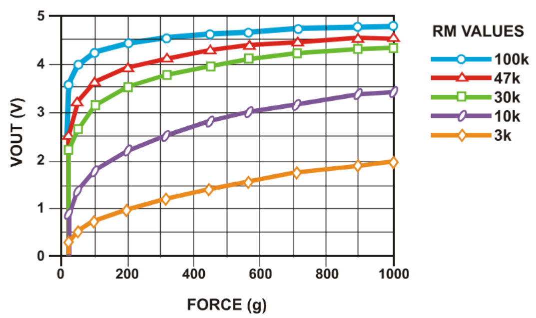


Figure 4.24: Response of the circuit for different R_M values [26]

Comparing the FSR to the Flexiforce, the FSR presents less sensitivity to noise and vibration, but it also presents a smaller dynamic range.

As the FSR is not accurate enough to measure the applied force, a simpler circuit was applied, which served to give the indication on whether the applied force exceeds a certain limit or not.

4.3 Data Transmission

As previously mentioned, the Bluetooth Low Energy is the wireless protocol used for data transmission. Going back to the BLE architecture, the focus is on the Generic Access Profile (GAP), the Attribute Protocol (ATT), and the Generic Attribute Profile (GATT) [113, 27, 114].

Starting with the GAP, this will define the roles of the devices when interacting with each other. It also takes care of advertisement related particularities, such as the advertisement data and parameters. Furthermore, it's responsible for the connection establishment and the security of the connection [113, 27, 114].

One of the most important concepts in GAP is the role of a device. A role serves to impose restrictions and to enforce certain behaviors. Combinations of roles are allowed to communicate with each other, and it's GAP's responsibility to establish these interactions [113, 27, 114].

The roles defined by the GAP are the following [113, 27, 114]:

- **Broadcaster:** is a device that advertises data packets periodically, but doesn't allow for a connection to be established.
- **Observer:** is a device that listens for those data packets but doesn't try to start a connection with the broadcaster.
- **Peripheral:** is a device that does not advertise data packets, but rather its presence, and waits for a connection to be established.
- **Central:** is a device that waits for the peripheral to send the advertising packets, and will start a communication.

The **Generic Access Profile** will define how the connection is handled between the peripheral and central devices, through the connection parameters. These parameters will control the data exchange rate between the devices, being negotiated between central and peripheral. However, the central device defines these values, meaning that the peripheral only gives indications about them [113, 27, 114].

There are three main parameters [113, 27, 114]:

- **Connection Interval (CI):** the master (another denomination for central) sets the CI, which indicates the frequency of each connection event, that is, how often the master will request data from the slave (another denomination for peripheral). The connection interval may vary between 7.5 ms and 4000 ms, with increments of 1.5 ms. Often, the peripheral will indicate a minimum and maximum connection interval, which the central device may not respect.
- **Supervision Timeout:** this parameter will define how long the connection may hold. Only when this value is surpassed, will the connection

be dropped, and eventually an attempt to reconnect. The value may vary between 100 ms and 32000 ms.

- **Slave Latency:** the slave latency will indicate the number of connection events that the peripheral can skip. This is a useful feature that allows the peripheral to minimize power consumption, and only replying either when it exceeds the slave latency or when it has data to send. The value may vary between 0 and 500 connection intervals. Note that this maximum is only valid if the Slave Latency x Connection Interval doesn't exceed the Supervision Timeout.

The Figure 4.25 shows the connection flow between the master and the slave, and how the connection parameters will affect it. In this particular case, the slave latency is set to 3, meaning that slave may skip three connection events [113, 27, 114].

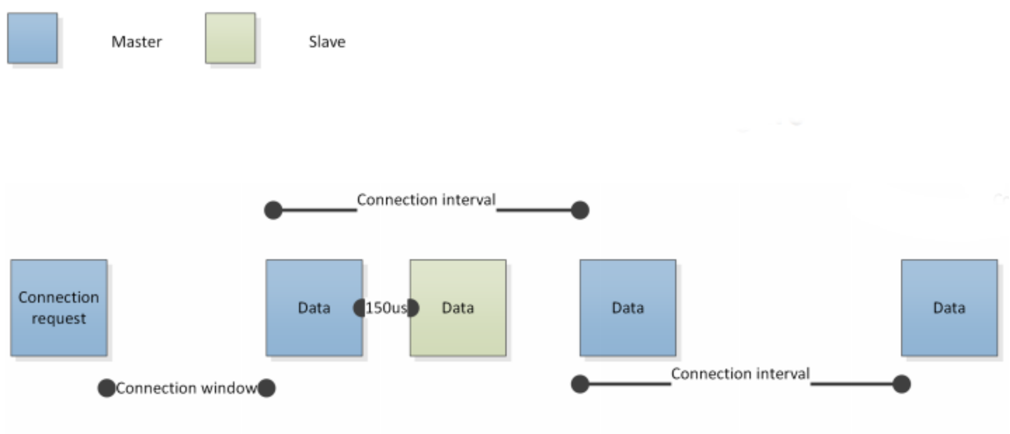


Figure 4.25: Connection flow with slave latency = 3 [27]

The **Attribute Protocol** defines two roles: the client and the server. A server holds resources, and those resources are represented by attributes. The ATT defines how those attributes can be accessed [113, 27, 114].

An attribute is composed of a Universally Unique Identifier (UUID), a 128-bit value, which defines the attribute type and what it is, as shown in Fig 4.26. It also contains a handle, is used to reference the attribute, which is stored in the server's database by increased (non-zero) handle value. The attribute permissions will dictate if a resource can be accessed, in the sense that it can be read or wrote, and will define the security level required. Finally, the attribute value contains the attribute's data [113, 27, 114].

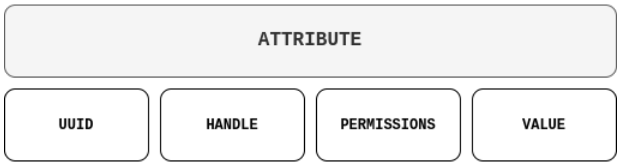


Figure 4.26: Attribute structure

How these attributes are accessed, is defined by six different methods. A client can send a **request** to a server, leading to a **response** from the server. A client may also send a **command** to the server, which doesn't trigger a response. An **indication** is sent by the server to the client, to inform that some value was changed. This requires a **confirmation** from the client, which acts as an acknowledgment. Lastly, there is the **notification** has the same goal as the indication but without the need for an acknowledgment [113, 27, 114].

To make use of those last three methods, the server must expose notifications or indications through Client Characteristic Configuration Descriptor (CCCD), setting the attribute to a specific value, and the client must enable either notifications or indications [113, 27, 114].

The **Generic Attribute Profile** lays on top of ATT, and its responsibility is to provide a common framework for the data exposed. A GATT server can accept ATT requests from the GATT client and can send ATT responses to the GATT client. The GATT client is also able to send indications and notifications to the GATT server [113, 27, 114].

The GATT structures hierarchically the data exposed by the GATT server and defines the procedures to interact with the attributes. The GATT also introduces some concepts to accomplish data structuring [113, 27, 114].

Starting with the highest level, the so-called **Profiles** are a general concept that intends to specify the composition and structure of data to accomplish a specific use-case [113, 27, 114].

Climbing down the hierarchy, the **Services** are a collection of attributes that satisfy a specific functionality in the GATT server [113, 27, 114].

Then, there are the **Characteristics**, which holds the information that the server wants to expose. The **Characteristic** has other attributes that help to define its behavior: *Properties*, which details how the **Characteristic** can be accessed and used (includes read, write, write without response, notify, indicate); *Value*, which contains the data itself; *Descriptor*, which holds information about the *Value*, that is a user description, extended properties, and more [113, 27, 114].

The following Figure gives an idea about the **Profile** structure.

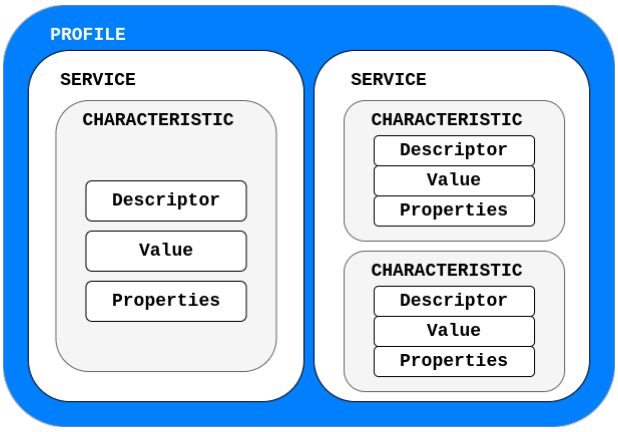


Figure 4.27: GATT data structure

4.3.1 System Design

Before the development of the technical particularities of the BLE, let's first define what are the system requirements and how can that be implemented on the BLE side. On a basic level, the system should:

- gather data from the force sensors.
- gather data from the MPU-6050.
- transmit that data to an external device.
- minimize power consumption.

For this use-case, the device should advertise and allow a connection from an external device. It has to expose the sensors' data, allowing the external device to read that information. It also needs to minimize the number of times that it exchanges data and the amount of data itself.

With that in mind, the GAP roles should be defined as the peripheral-central pair, in which the device acts as the peripheral. This is the case, being that we want to establish a connection with the external device. In the ATT and GATT side, the device will assume the role of a server, which makes sense because, in networking, the server holds the data or resources that can be accessed.

Then, the sensors' data is exposed through Services and Characteristics, which implies the development of customized Service, let's call it "Brux Service", composed of Characteristics that will represent the sensors' data. Three Characteristics are present to hold the value of the force sensors, accelerometer, and gyroscope.

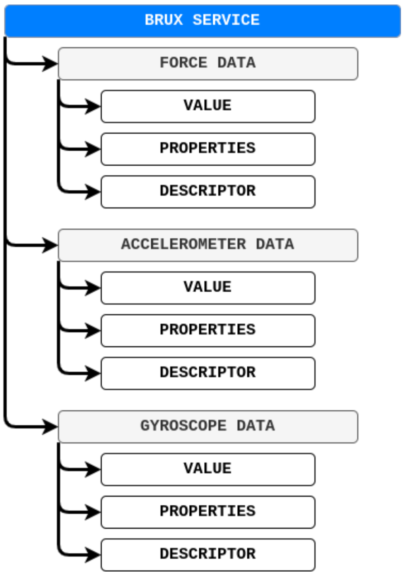


Figure 4.28: Structure of custom Service

In respect to the implementation, the first step is to define the UUIDs for the Services and Characteristics. The base UUID was set to the following:

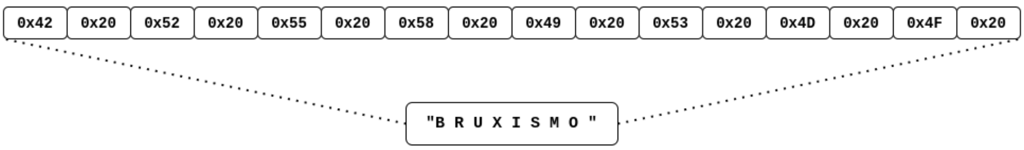


Figure 4.29: Base UUID meaning

Once the base UUID is assigned, the Service and Characteristics will derive from that. Usually, the 3rd and 4th most significant bytes of the base UUID is changed accordingly. The assigned UUIDs are presented in Fig 4.30.

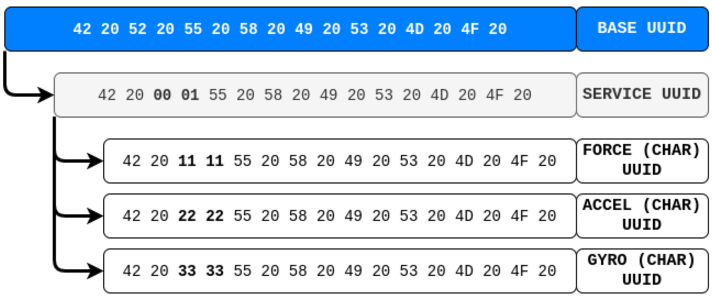


Figure 4.30: All UUIDs assigned

Let's start the service implementation. The idea is to establish a relation between code and the impact it has on the BLE structure.

1. The assigned UUIDs can be defined by the following macros. Note that the bytes are in reverse order. The reason is that these values are stored in little-endian, which means that the LSB is stored first.

```
#define BRUX_BASE_UUID    {0x20, 0x4F, 0x20, 0x4D, 0x20, 0x53, 0x20, 0x49, 0x20, 0x58, 0x20, 0x55, 0x20, 0x52, 0x20, 0x42}

#define BRUX_SERVICE_UUID   0x0001
#define BRUX_FORCE_UUID     0x1111
#define BRUX_ACCEL_UUID     0x2222
#define BRUX_GYRO_UUID      0x3333
```

2. Two data structures are defined to deal with events. One will gather all the possible events (1-9), and the other will store the event (11-14) The type of events are self-explanatory.

```
1  typedef enum
2  {
3      BLE_BRUX_ACCEL_NOTIFICATION_ENABLED,
4      BLE_BRUX_ACCEL_NOTIFICATION_DISABLED,
5      BLE_BRUX_GYRO_NOTIFICATION_ENABLED,
6      BLE_BRUX_GYRO_NOTIFICATION_DISABLED,
7      BLE_BRUX_FORCE_NOTIFICATION_ENABLED,
8      BLE_BRUX_FORCE_NOTIFICATION_DISABLED,
9  } ble_brux_evt_type_t;
10
11 typedef struct
12 {
13     ble_brux_evt_type_t evt_type;
14 } ble_brux_evt_t;
```

3. Define an event handler that deals with Service events. It also contains the security options needed for the Service initialization and can be utilized when a Characteristic contains a CCCD.

```
typedef struct
{
    ble_brux_evt_handler_t         evt_handler;
    ble_srv_cccd_security_mode_t   custom_value_char_attr_md;
} ble_brux_init_t;
```

4. Now, the definition of the Service data structure. It will contain all the handles (Service, Characteristics, and connection), UUID, etc.

```
struct  ble_brux_s
{
  ble_brux_evt_handler_t      evt_handler;
  uint16_t                    service_handler;
  ble_gatts_char_handles_t    brux_force_handles;
```

```
  ble_gatts_char_handles_t    brux_accel_handles;
  ble_gatts_char_handles_t    brux_gyro_handles;
  uint16_t                    conn_handle;
  uint8_t                     uuid_type;
};
```

5. Next, the implementation of connection (1-4) and disconnection (6-10) events. These functions will either set and store, or reset the connection handle, depending on the event type, that is, connection or disconnection, respectively.

```
1  static void on_connect(ble_brux_t * p_brux, ble_evt_t const * p_ble_evt)
2  {
3     p_brux->conn_handle = p_ble_evt->evt.gap_evt.conn_handle;
4  }
5
6  static void on_disconnect(ble_brux_t * p_brux, ble_evt_t const * p_ble_evt)
7  {
8     UNUSED_PARAMETER(p_ble_evt);
9     p_brux->conn_handle = BLE_CONN_HANDLE_INVALID;
10 }
```

6. The write handler, in this case, will handle the cases where notifications were either enabled or disabled by the client. It will check if the CCCD of a certain Characteristic is written and if it has the correct length (5). Then it will check if the notifications were enabled or disabled (12), and set the event accordingly, proceeding to call the application event handler. The following piece of code only shows the process of one Characteristic, but the same is repeated for the other ones.

```
1  static void on_write(ble_brux_t * p_brux, ble_evt_t const * p_ble_evt)
2  {
3    ble_gatts_evt_write_t const * p_evt_write = &p_ble_evt->evt.gatts_evt.params.write;
4
5    if ((p_evt_write->handle == p_brux->brux_accel_handles.cccd_handle) && (p_evt_write->len == 2) )
6    {
7
8       if (p_brux->evt_handler != NULL)
9          {
10             ble_brux_evt_t evt;
11
12             if (ble_srv_is_notification_enabled(p_evt_write->data))
13             {
14                 evt.evt_type = BLE_BRUX_ACCEL_NOTIFICATION_ENABLED;
```

```
            }
            else
            {
                evt.evt_type = BLE_BRUX_ACCEL_NOTIFICATION_DISABLED;
            }
            // Call the application event handler.
            p_brux->evt_handler(p_brux, &evt);
        }
    }
    /* ... */
}
```

7. For dealing with the BLE events, the following function is called by the SoftDevice. Its responsibility is to act upon certain events, calling the event handlers accordingly. These event handlers were defined previously like connection events (10-12), disconnection events (14-16), and write events (18-20).

```
1   void ble_brux_on_ble_evt( ble_evt_t const * p_ble_evt, void * p_context)
2   {
3    ble_brux_t * p_brux = (ble_brux_t *) p_context;
4
5    if (p_brux == NULL || p_ble_evt == NULL)
6       return;
7
8    switch (p_ble_evt->header.evt_id)
9    {
10      case BLE_GAP_EVT_CONNECTED:
11         on_connect(p_brux, p_ble_evt);
12         break;
13
14      case BLE_GAP_EVT_DISCONNECTED:
15         on_disconnect(p_brux, p_ble_evt);
16         break;
17
18      case BLE_GATTS_EVT_WRITE:
19         on_write(p_brux, p_ble_evt);
20         break;
21
22      default:
23         break;
24   }
25  }
```

8. Define the base UUID.

```
ble_uuid128_t base_uuid = {BRUX_BASE_UUID};
err_code = sd_ble_uuid_vs_add(&base_uuid, &p_brux->uuid_type);
```

9. Add the custom Service. Includes setting the Service UUID (4-5), and initializing some connection parameters (7).

```
1  p_brux->evt_handler = p_brux_init->evt_handler;
2  p_brux->conn_handle = BLE_CONN_HANDLE_INVALID;
3
4  ble_uuid.type = p_brux->uuid_type;
5  ble_uuid.uuid = BRUX_SERVICE_UUID;
6
7  err_code = sd_ble_gatts_service_add(BLE_GATTS_SRVC_TYPE_PRIMARY, &
    ble_uuid, &p_brux->service_handler);
```

10. To create and add a Characteristic:

    a) Set the UUID.

    b) CCCD configuration. Reads and writes are enabled.

    c) Characteristic configuration. Enable reads, writes without responses, and notifications (1-3). Also, assign the CCCD configuration to the corresponding data pointer (13), and set a user description (5-9).

```
1       char_md.char_props.read        = 1;
2       char_md.char_props.write_wo_resp  = 1;
3       char_md.char_props.notify         = 1;
4
5       static char user_desc[]         = "FORCE VALUE";
6
7       char_md.p_char_user_desc        = (uint8_t *) user_desc;
8       char_md.char_user_desc_size     = strlen(user_desc);
9       char_md.char_user_desc_max_size = strlen(user_desc);
10
11      char_md.p_char_pf       = NULL;
12      char_md.p_user_desc_md  = NULL;
13      char_md.p_cccd_md       = &cccd_md;
14      char_md.p_sccd_md       = NULL;
```

    d) Attribute configuration. Match the attribute configuration with characteristic's properties, that is, enabling reads and writes.

    e) Characteristic Value configuration. Set the data pointer to the assigned UUID (1), and the attribute configuration (2). Set the size of the value (3), and its actual value (the initial value is set to *0xFFFF FFFF*) (6-7).

```
1       attr_char_value.p_uuid      = &ble_uuid;
2       attr_char_value.p_attr_md   = &attr_md;
3       attr_char_value.init_len    = 2*sizeof(int16_t);
4       attr_char_value.init_offs   = 0;
5       attr_char_value.max_len     = 2*sizeof(int16_t);
6       int16_t value[2]            = {0xFFFF, 0xFFFF};
```

```
    attr_char_value.p_value     = (uint8_t *) value;
```

f) Add the Characterisitc.

```
    err_code = sd_ble_gatts_characteristic_add(p_brux->service_handler, &char_md, &attr_char_value, &p_brux->brux_force_handles);
```

11. The last steps, involve the Characteristic value update. It receives the custom Service pointer (*p_brux*), and a pointer to the new value of the Characteristic (*force_value*).

```
uint32_t ble_brux_force_update(ble_brux_t * p_brux, int16_t * force_value)
```

a) Initialize the value struct. The value is four bytes, two from each force sensor (in this case).

```
    ble_gatts_value_t gatts_value;

    memset(&gatts_value, 0, sizeof(gatts_value));

    gatts_value.len     = 2 * sizeof(int16_t);
    gatts_value.offset  = 0;
    gatts_value.p_value = (uint8_t *) force_value;
```

b) Update the server's database with the new value.

```
    err_code = sd_ble_gatts_value_set(p_brux->conn_handle,
                                      p_brux->brux_force_handles.value_handle,
                                      &gatts_value);
```

c) Then, check if a connection is established (1) and, if so, send a notification with the new value.

```
1       if ((p_brux->conn_handle != BLE_CONN_HANDLE_INVALID))
2       {
3           ble_gatts_hvx_params_t hvx_params;
4
5           memset(&hvx_params, 0, sizeof(hvx_params));
6
7           hvx_params.handle = p_brux->brux_force_handles.value_handle;
8           hvx_params.type   = BLE_GATT_HVX_NOTIFICATION;
9           hvx_params.offset = gatts_value.offset;
10          hvx_params.p_len  = &gatts_value.len;
11          hvx_params.p_data = gatts_value.p_value;
12
```

```
        err_code = sd_ble_gatts_hvx(p_brux->conn_handle, &
hvx_params);
    }
```

4.4 Application

So, let's start with SAADC sampling. The system requires a continuous sampling of the force sensors, which can be done by triggering an interrupt at a given rate, allowing the CPU to remain in idle state, or even to process another routine, instead of being busy-waiting. The Real-Time Counter (RTC) module provides a low-power timer on the low-frequency clock source (LFCLK), and it is used to generate the interrupt. As shown in Fig 4.31, every 140 ms an interrupt is generated and, consequently, leading to initialization of the SAADC peripheral, and a sample task is triggered.

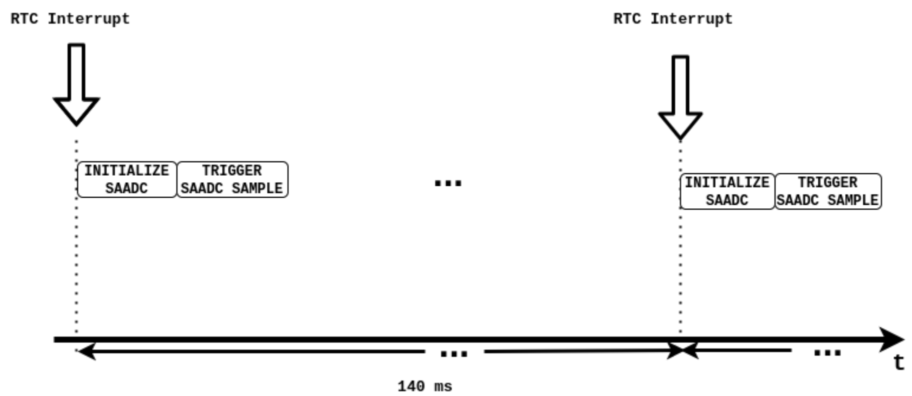


Figure 4.31: Flow of a RTC interrupt

Once the RTC interrupt routine is executed, the CPU should return to an idle state. Only when a SAADC event occurs will the CPU deal with that particular event. The Fig 4.32 demonstrates the flow of execution. If the event is due to the buffer was filled, meaning that the SAADC sample has finished, the data is processed and, then the SAADC peripheral is uninitialized, which will allow for less power consumption (mainly caused by EasyDMA).

To increase flexibility to the system, we've defined two modes of operation in the SAADC: low-resolution and high-resolution modes. While in the low-resolution mode, the sampling frequency corresponds to approximately 7 Hz, allowing to keep track of the force exerted between the teeth at an acceptable rate. The device enters in high-resolution mode when the force exceeds a predefined limit. At this point, the sampling frequency increases to 50 Hz, allowing an equilibrium between power consumption and resolution, as seen in Fig 4.33.

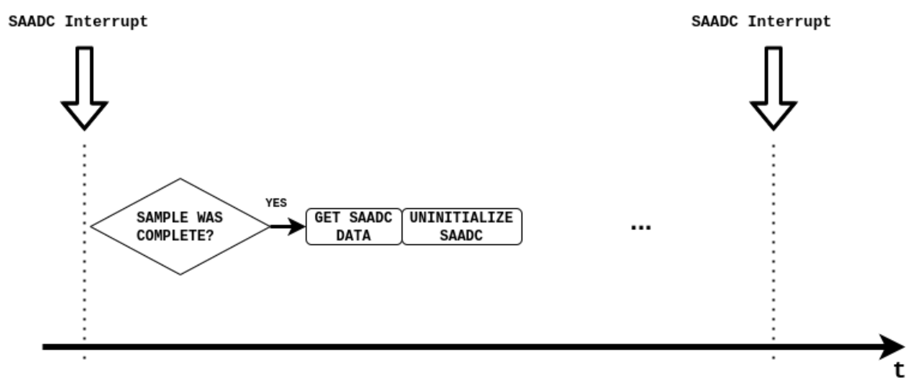


Figure 4.32: Flow of a SAADC interrupt

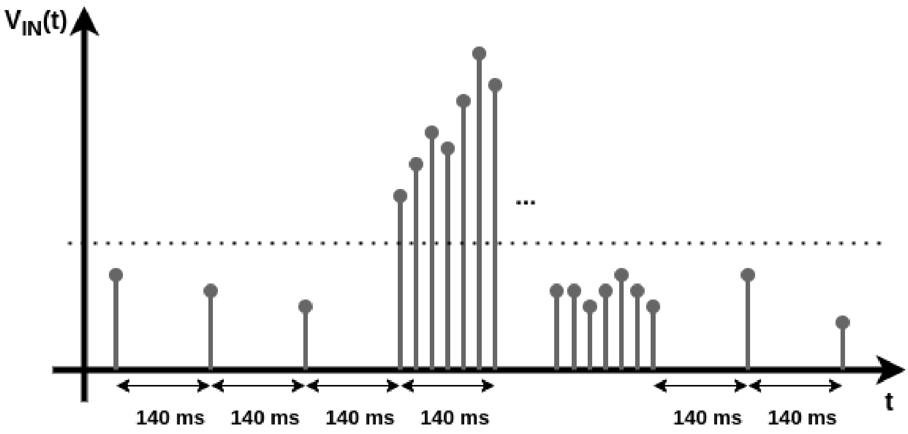


Figure 4.33: Dynamic sampling frequency of SAADC

The same principle applies to the IMU. In its normal state, the IMU is in sleep mode. Then, depending on the enabled notifications, the IMU is configured accordingly. If only one of the notifications is enabled, that is, either Accelerometer or Gyroscope, the sampling rate is 2 Hz. If both notifications are enabled, the device enters in high-resolution mode, increasing the sampling rate to 20 Hz, as shown in Fig 4.34.

For the BLE interface, there are a few things to configure. First, the BLE stack must be configured, such as the connection settings (peripheral and central links, ATT MTU size, etc.), and the application RAM's start address. Next, the Softdevice and, subsequently, the BLE stack is enabled. Lastly, a handler for BLE events is defined. The Fig 4.35 shows this flow of execution.

Furthermore, the GAP is initialized by configuring the connection security. For this use-case, the connection is set to open-link (no security needed). Then, the GAP device name is defined (*"BRUXISM"* in this case), which will be present in the advertising data. Finally, the connection parameters are configured. As

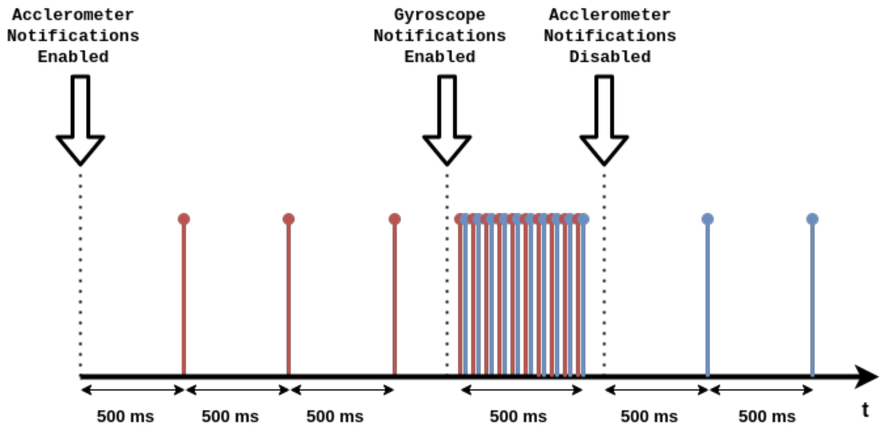


Figure 4.34: Dynamic sampling frequency of IMU

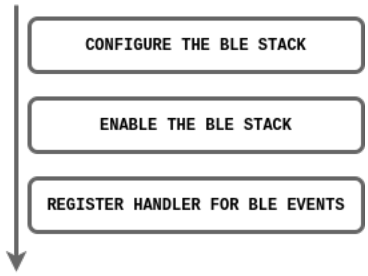


Figure 4.35: Flow of execution related to the BLE stack

explained before, the peripheral is not responsible to define the GAP parameters, but it can negotiate with the central device through the Peripheral Preferred Connection Parameters (PPCP), and are composed of four fields:

- The **Minimum Connection Interval** indicates the minimum interval between connection events. The **Maximum Connection Interval** indicates the maximum interval between connection events. These parameters will define the data throughput, which for this application is low, and in the "worst-case" scenario, the device will send data each 20 ms. So, these parameters were configured to 15 ms and 25 ms, respectively.

- The **Supervision Timeout**, as described previously, represents the time in which either device assumes that the connection is lost. In this system, this parameter will be set to the maximum possible, which is 32 seconds.

- The **Slave Latency** is the number of connection events that the slave can skip. As for the Supervision Timeout, the goal is to make this value as high as possible, since this will allow the slave to minimize power consumption.

The value is given by:

$$Slave \quad Latency = \frac{Supervision \quad Timeout}{Maximum \quad Connection \quad Interval * 2} \quad - \quad 1 \quad (4.40)$$

Resolving this equation with the values defined above, results in a Slave Latency of 799. However, this value exceeds the maximum allowed, which is 500, so we set the value to maximum.

The flow of execution is represented in Fig 4.36.

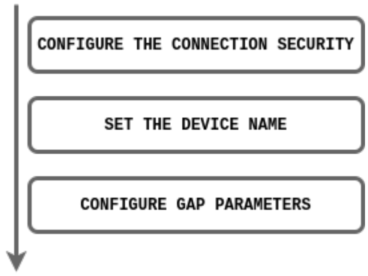


Figure 4.36: Flow of execution related to the GAP parameters

The next step is to initialize the GATT module and to define the ATT MTU size for the next connection, as shown in the following Figure.

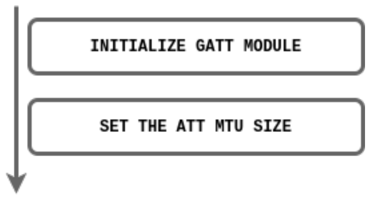


Figure 4.37: Flow of execution related to the GATT module initialization

Then, the services used by the application must be initialized, which includes our customized service, denominated by *Brux* service. First, an event handler is defined, and, afterward, the service initialization function is called. The Fig 4.38 depicts the process.

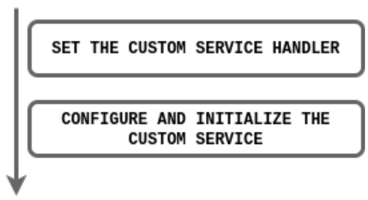


Figure 4.38: Flow of execution related to the service initialization

The advertising module also needs initialization. First, some advertising data fields are defined, such as the device name, if it should include appearance or not, the flags (LE General Discoverable Mode, Basic Rate (BR)/ Enhanced Data Rate (EDR) not supported), and the UUIDs. Next, the configuration of the advertising module, which configures the advertising mode (fast, in this situation), the advertising interval, and the advertising timeout.

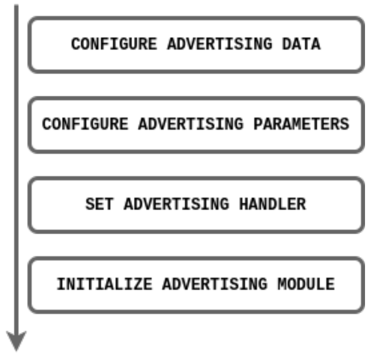


Figure 4.39: Flow of execution related to the advertising initialization

Finally, the connection parameters module is responsible for the negotiation with the central about the connection parameters. Here, the procedure of the negotiation of the "terms" is set through some fields. These fields are the time between an initiation event and the first parameter's negotiation, the interval between negotiations, and the number of negotiations.

4.4.1 Application Logic

Now, it's time to think about the application logic, that is, how will the application behave, and how can the system requirements be accomplished without losing the low-power characteristic. For that, let's define some device status.

In the **Idle Status**, the device is configured to take continuous SAADC samples. At this stage, the device is not advertising, and the IMU is in sleeping mode, which produces the lowest power consumption possible (in our case).

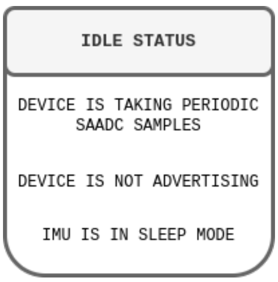


Figure 4.40: Idle Status definition

In the **Advertising Status**, the device starts advertising. The periodic sample of the SAADC continues, and the IMU remains in sleep mode. The device assumes the role of a peripheral and waits for a connection.

Figure 4.41: Advertising Status definition

In the **Connected Status**, the device has established a connection with an external device. At this time, the periodic sampling of the SAADC remains, and the IMU stays sleeping. As it's connected, the device stops advertising.

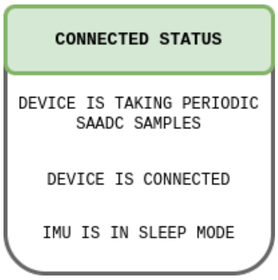


Figure 4.42: Connected Status definition

The device is in the **Notification Status** if the notifications were enabled by the client. Depending on the type of notifications, the system will have different behaviors, which means that the user is directly influencing the functioning of the system. All the different possibilities are described in Fig 4.43.

Lastly, the device is in the **Hold Status** if it was tracking either accelerometer or gyroscope data, and several consecutive SAADC samples didn't exceed the defined limits, meaning that the probability of a bruxism event occurring is low, which allows the device not to send new data during that time. So, to minimize power consumption, the IMU is put to sleep.

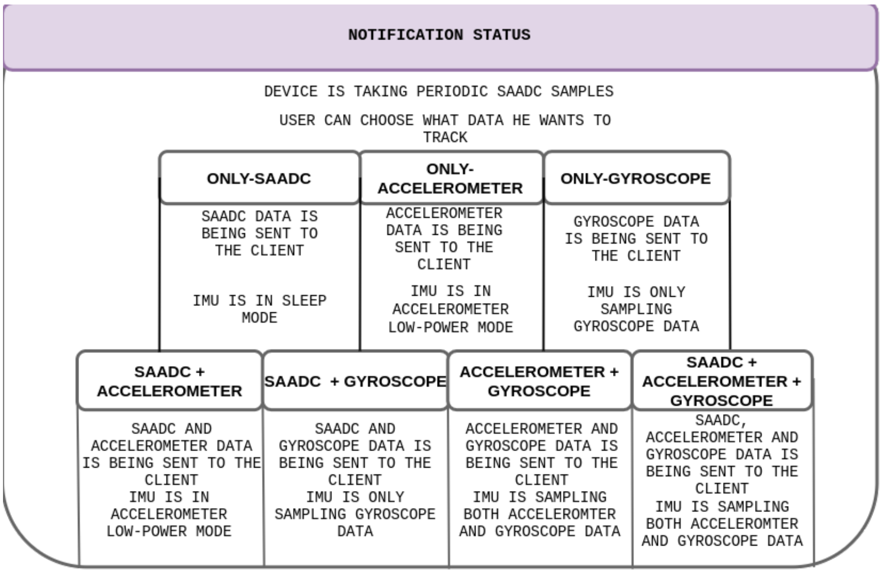


Figure 4.43: Notification Status definition

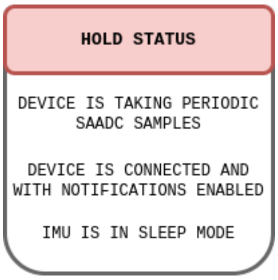


Figure 4.44: Hold Status definition

Once all the statuses are defined, the transitions between them need to be established. The Fig 4.45 graphically represents these transitions, but let's try to describe it.

The device starts on the **Idle Status**. The next "step" is to start advertising, but in what conditions should the device transition to the advertising status is the question. It should start advertising when it has some data to send. As this is the default status, it may serve as a mode to "activate" the device. So, once a SAADC sample exceeds the defined limits, the device transitions to the **Advertising Status**, and starts advertising.

Once advertising (peripheral role), the device waits for a connection. If a predefined number of samples of the SAADC don't exceed the limits defined (which acts like a threshold for the amount of force), and a connection wasn't established, then the device returns to the **Idle Status**. But, if a connection is established, the device transitions to the **Connected Status**.

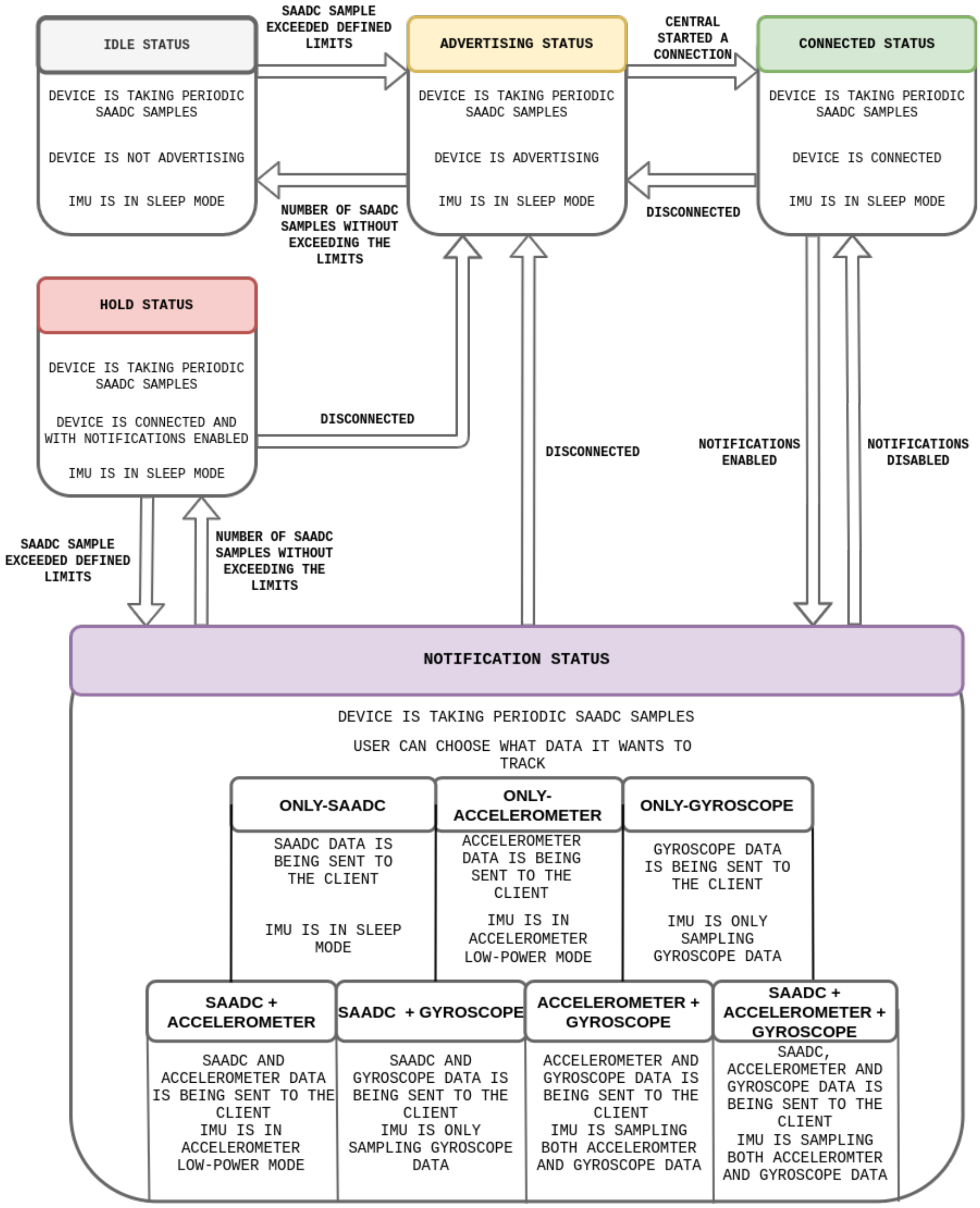


Figure 4.45: Relation between all status

Now that the device is connected, the user has the option to choose what data it wants to receive (Force, Accelerometer, Gyroscope). Depending on the choice, the user enables the notification associated with that data. If a notification is enabled, then the device transitions to the **Notification Status**. If the user terminates the connection for some reason, the device returns to the **Advertising Status**.

In the **Notification Status**, as said previously, the user has the freedom to track what type of data it wants. At this stage, the user may enable and disable the notifications associated with different Characteristics. If all the notifications are disabled, then no data is transmitted between the devices, and the device returns to the **Connected Status**. If the connection drops, then the device transitions to the **Advertising Status**.

The **Hold Status** is simply a way to minimize power consumption by stopping data being exchanged if the predefined number of samples don't exceed thethreshold. Imagine that the notifications associated with the Accelerometer Characteristic were enabled, then at this moment, the IMU would be put to sleep, returning to its previous state when a SAADC sample exceeds the limit. When this happens, the device transitions to the **Notification Status**. If the connection is lost, then the device returns to the **Advertising Status**.

4.4.2 Implementation

As for the implementation, we start by defining some data structures that will help track the device status. So the first data structure will store only one of the following properties:

- *DEVICE_IS_ADVERTISING*: this flag indicates that the device is in its Advertising Status.
- *DEVICE_IS_CONNECTED*: this flag indicates that there is an established connection between devices and no notification is enabled.
- *DEVICE_IS_IDLE*: this flag indicates that the device is in its Idle Status.
- *DEVICE_IS_NOTIFYING*: this flag is set when the device is Notifying.
- *DEVICE_IS_HOLDING*: this flag indicates that the device is in the Hold Status.

```
typedef enum {
    DEVICE_IS_ADVERTISING,
    DEVICE_IS_CONNECTED,
    DEVICE_IS_IDLE,
    DEVICE_IS_NOTIFYING,
```

```
    DEVICE_IS_HOLDING
} device_status_t;
```

For the SAADC, two data structures to manage the operation of the device. An *enum*, which contains a group of constants representing the resolution of SAADC, and a structure with the following members:

- *m_saadc_initialized*: indicates if the SAADC is initialized or not.
- *limit_exceeded*: indicates if a sample exceeded the defined limits.
- *samples_not_exceeded*: number of consecutive samples that don't exceed the defined limits.
- *m_adc_evt_counter*: incremented at each "DONE" event.
- *notification_enabled*: indicates if the notifications for the force values were enabled.
- *stop_sending*: flag set if notifications are enabled, but the device enters in the Hold Status.
- *resolution*: indicates the current resolution of the SAADC.

```
typedef struct saadc {                        |   typedef enum {
    bool m_saadc_initialized;                 |       LOW_RES,
    bool limit_exceeded;                      |       HIGH_RES
    uint8_t samples_not_exceeded;             |   } resolution_saadc;
    uint32_t m_adc_evt_counter;               |
    bool notification_enabled;                |
    bool stop_sending;                        |
    resolution_saadc resolution;              |
} saadc_status_t;                             |
```

A similar organization is done for IMU. The *mpu_stages* describes how the IMU is configured, and the *mpu_status_t* helps to manage the behavior of the MPU-6050.

For the *mpu_stages*:

- *MPU_SLEEP*: the MPU is in sleep mode.
- *MPU_ACCEL*: the MPU is in the Accelerometer Low-Power mode.
- *MPU_GYRO*: the MPU is configured to only track the gyroscope data.
- *MPU_ACCEL_GYRO*: the MPU is tracking both kinds of data.

For the *mpu_status_t*:

- *stage*: indicates in what stage is the MPU configured.
- *prev_stage*: this is used when the Hold Status is used. It'll hold the previous stage before it was put to sleep.
- *status_changed*: indicates that the MPU must be configured to another stage.
- *updating*: acts like a blocking variable, in which it doesn't allow the MPU to be configured while it is being accessed.
- *hold*: indicates if it's in the Hold Status.

```
typedef enum {                  |       typedef struct {
    MPU_SLEEP,                  |           mpu_stages stage;
    MPU_ACCEL,                  |           mpu_stages prev_stage;
    MPU_GYRO,                   |           bool status_changed;
    MPU_ACCEL_GYRO              |           bool updating;
} mpu_stages;                   |           bool hold;
                                |       } mpu_status_t;
```

The behavior of the application will change through a set of events. The type of events will determine the device status. To better understanding, let's try to describe what are these events, and how they will influence the system behavior.

Let's start with the *RTC handler*. This routine, as mentioned before, is executed each 140 ms. The basic functionality is to trigger a SAADC sample task, as seen in Fig 4.46.

The *SAADC callback* routine is called every time that a SAADC event occurs. There are two types of events that will be dealt with: one referring to the end of the sample task, and another relative to and another related to event monitoring. Only one is attended each time, as shown in Fig 4.47.

If the event corresponds to the end of the sample task ("DONE" event), then three types of checks are performed. The first one, checks if it's time to perform offset calibration of the SAADC, then it checks if the limits were exceeded during the present sample task, and, lastly, it will check what is the status of both the notification related to the Force Characteristic and the IMU, as demonstrated in Fig 4.48.

Concerning the offset calibration check, shown in Fig 4.49, it tracks the number of SAADC samples elapsed, and, if it corresponds to a predefined number of samples, it will perform the calibration of the SAADC waiting until the *EVENTS_CALIBRATEDONE* bit is set. Otherwise, the event counter is incremented, and the data conversion is issued, resulting in the SAADC sample acquisition.

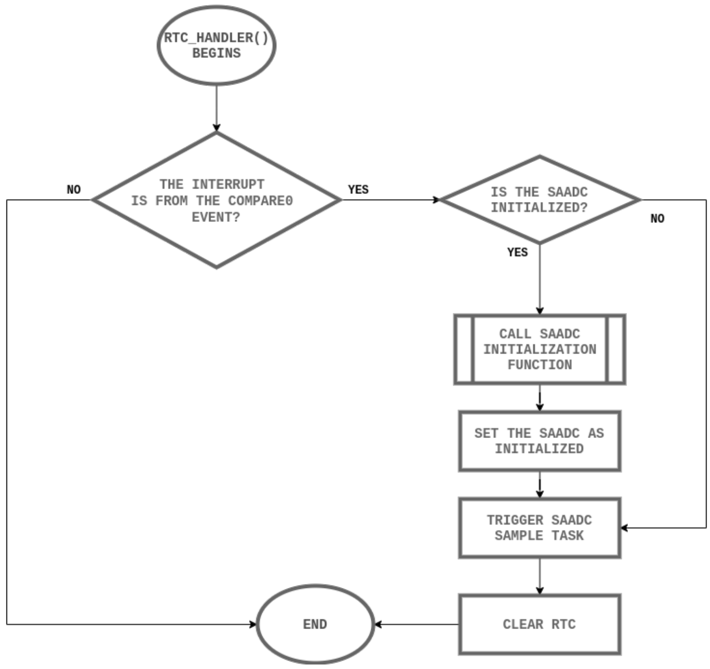


Figure 4.46: RTC handler

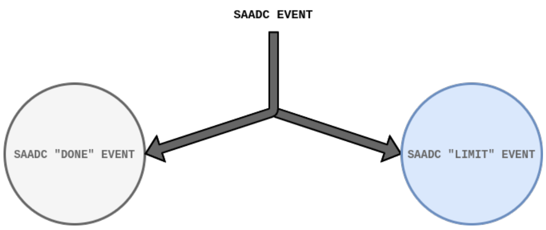


Figure 4.47: SAADC callback events

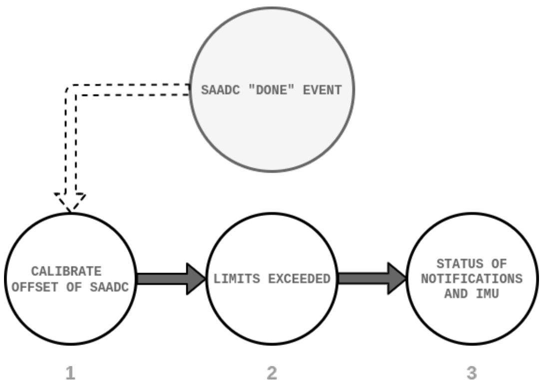


Figure 4.48: SAADC "DONE" event steps

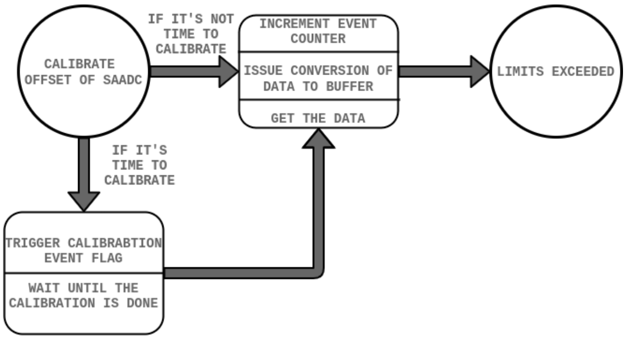


Figure 4.49: Offset calibration flow

The application needs to check if the SAADC samples exceeded the predefined values, meaning that some force is being exerted between the teeth. If the values were exceeded, then we need to check if the device is in low-resolution mode, that is, with a sampling frequency of 7 Hz, and if that's the case, the device is set to high-resolution mode, assuming a sampling frequency of 50 Hz. Furthermore, it will check if the device is in its Idle Status, and it will start advertising if it is. Otherwise, it will set the device to the appropriate status (either Connected or Notification Status). Then, it resets the flag relative to the limits exceeded and the number of samples necessary for the device to be set to low-resolution mode. The process is represented in Fig 4.50.

In case that the limits were not exceeded, we check if the predefined number of samples was reached, and the device is not on its Idle Status. If it isn't, it decrements the number of samples, but if it is, it will stop advertising if the device is not connected or put the device on the Hold Status. Then, it checks if the device is in high-resolution mode, and if it is, it sets it to low-resolution mode. Finally, the SAADC peripheral is uninitialized, and if the notifications of the Force Characteristic are enabled, then the value it's updated. The process is shown in Fig 4.50.

In case of a "LIMIT" event, Fig 4.51, the procedure is to check if the device is in the Hold Status, and if it is, set the IMU to its previous state. A flag indicating that the limits were exceeded is also set.

The BLE event handler is called whenever an event occurs. The most relevant events are relative to connection, disconnection, and timeouts. In case of a connection event, the connection handle is updated, and the device is set to the Connected Status. In case of a disconnection event, the connection handle is updated, set the device to the Advertising Status, and set the MPU-6050 to sleep mode. In case of a timeout, we simply disconnect. The procedure is defined in Fig 4.52.

Another type of event is those of the "Brux" Service. These events are dealt

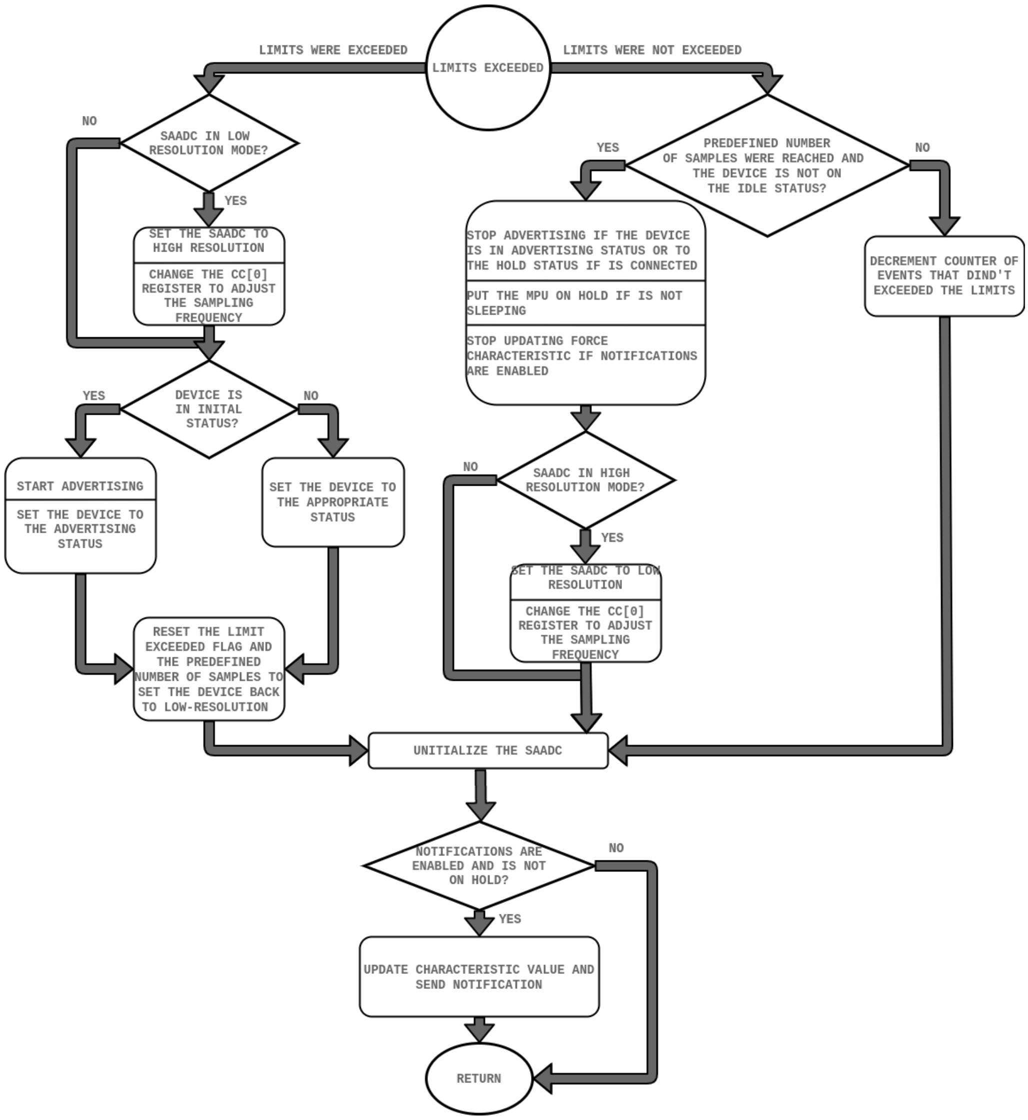


Figure 4.50: Limits exceeded check flow

with on the application side, according to the Fig 4.53. Every Characteristic has two events associated: notifications enabled and disabled.

For the Accelerometer Characteristic, when notifications are enabled, we set a flag indicating there was a change, setting the IMU status either low-power mode (only sampling the accelerometer) or to the high-resolution mode, in which the accelerometer and gyroscope are being sampled with a frequency of 20 Hz. If notifications were disabled, the flag indicating there was a change is set, and the timer is stopped. The IMU is set to sleep mode if it was in low-power mode. Otherwise, it sets the device to low-resolution mode, with the gyroscope only being sampled two times a second. The process is similar to the Gyroscope

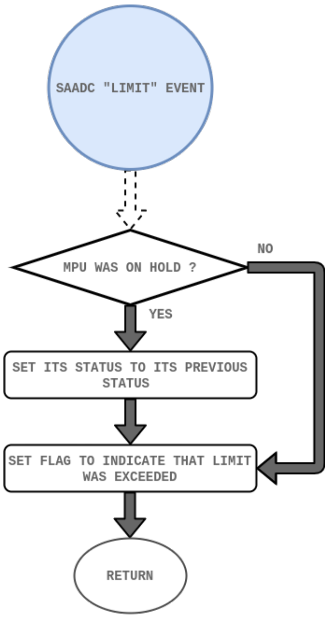


Figure 4.51: SAADC "LIMIT" event flow

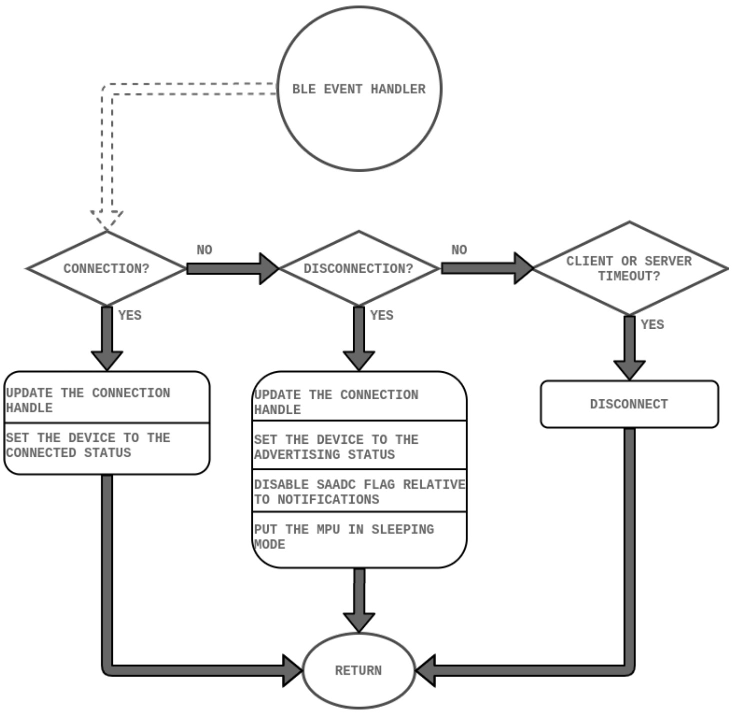


Figure 4.52: BLE event handler

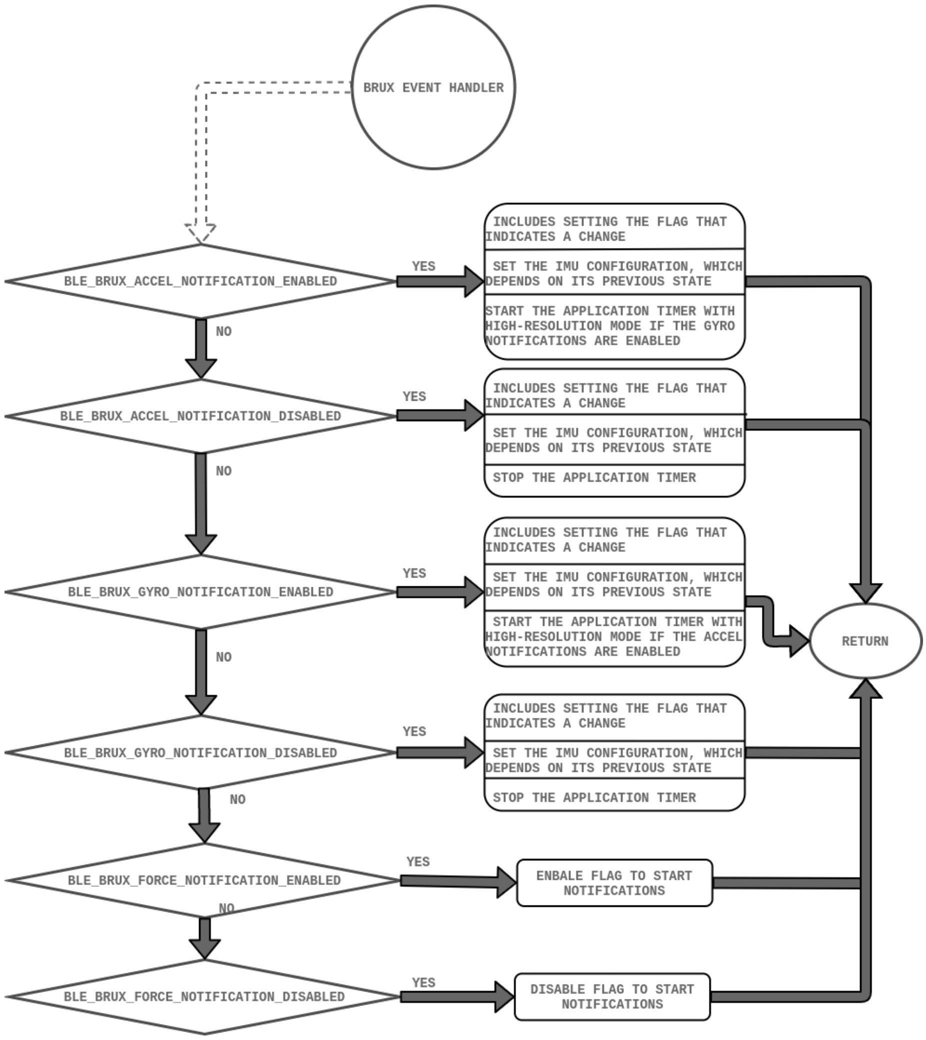


Figure 4.53: Service event handler

Characteristic.

If the Force Characteristic was enabled, then a flag is set to start updating the Characteristic value. If it was disabled, then the flag is reset, as shown in Fig 4.53.

But there is another particularity about the application which involves the accelerometer and gyroscope notifications. As they are independent of each other, they require an independent updating timing. First, the rate at which the values are updated was defined (500 ms, in this case). Then, two application timers are created. Here, a callback function is attributed, which is called whenever a timeout is generated. Here, the IMU is read, and the values are updated (notification sent). Finally, the timers will only start if the notifications are enabled. The flow is demonstrated in Fig 4.54.

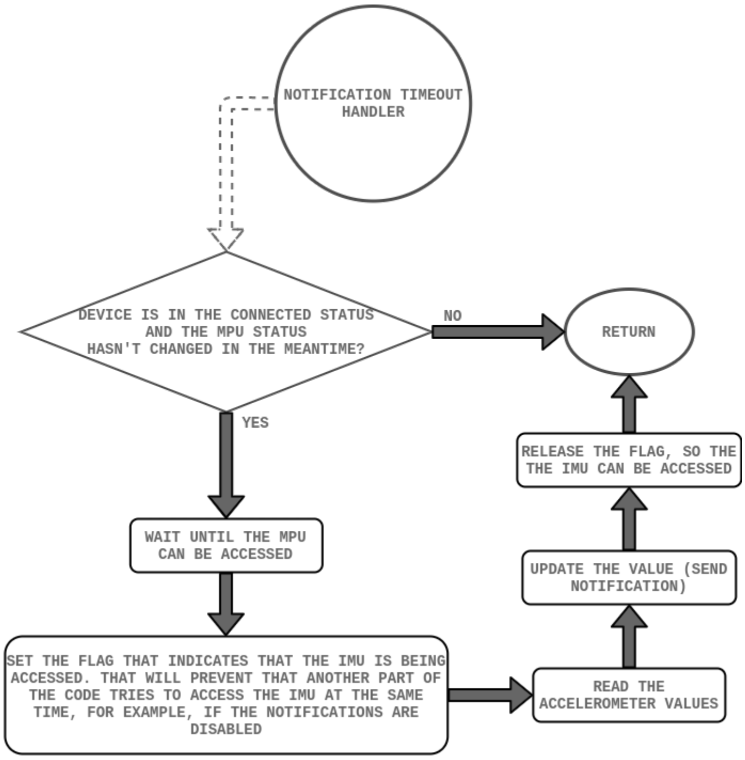


Figure 4.54: Application timer timeout handler

Lastly, in the main loop, there are a few verifications to be done, which will impact the overall behavior of the system. Those verifications are if the device should stop advertising, if the IMU status has changed, and if the device is set to Hold status. After those checks, the device is put in an idle state.

The purpose of the advertising check is to not waste power in advertising if the device has no data to send. So it will check if the device is set to stop advertising and, if it is, it will stop, putting the device in the **Idle Status**. Otherwise, it will pass to the next verification, as seen in Fig 4.55.

The objective of the IMU status changed check is to configure the MPU-6050 according to the notifications enabled by the user. In this step, we check if the IMU is set to be changed or not. If that's the case, it will check to what mode it should change, including the sleep, accelerometer-only, gyroscope-only, and accelerometer and gyroscope combined. At either of these stages, the MPU-6050 is configured, starting and stopping the application timer accordingly, as we can see in Fig 4.56.

And, finally, the hold status check. The goal of this verification is to put the IMU in sleep if the device is the **Hold Status**. Here, we verify if the device is in the **Hold Status** and if the previous stage was the sleep mode (guarantee that it's executed only once). If that's the case, we set the IMU's previous stage to its

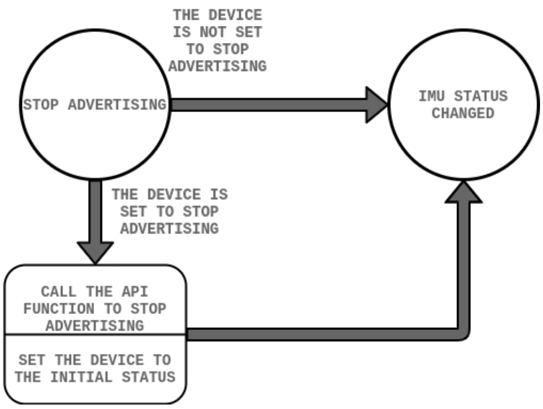


Figure 4.55: Stop advertising check

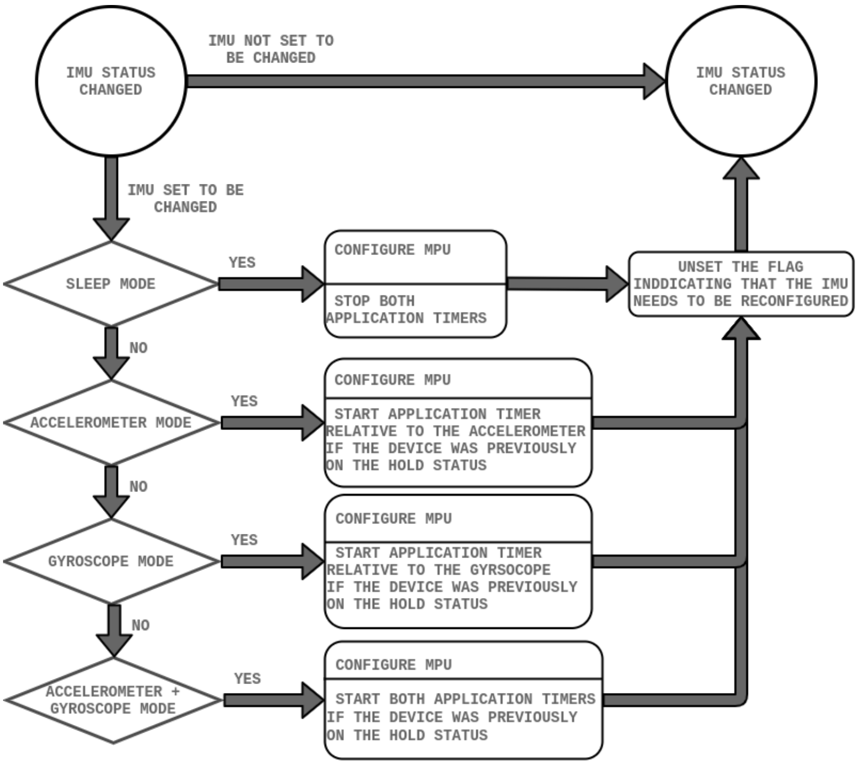


Figure 4.56: IMU status changed check

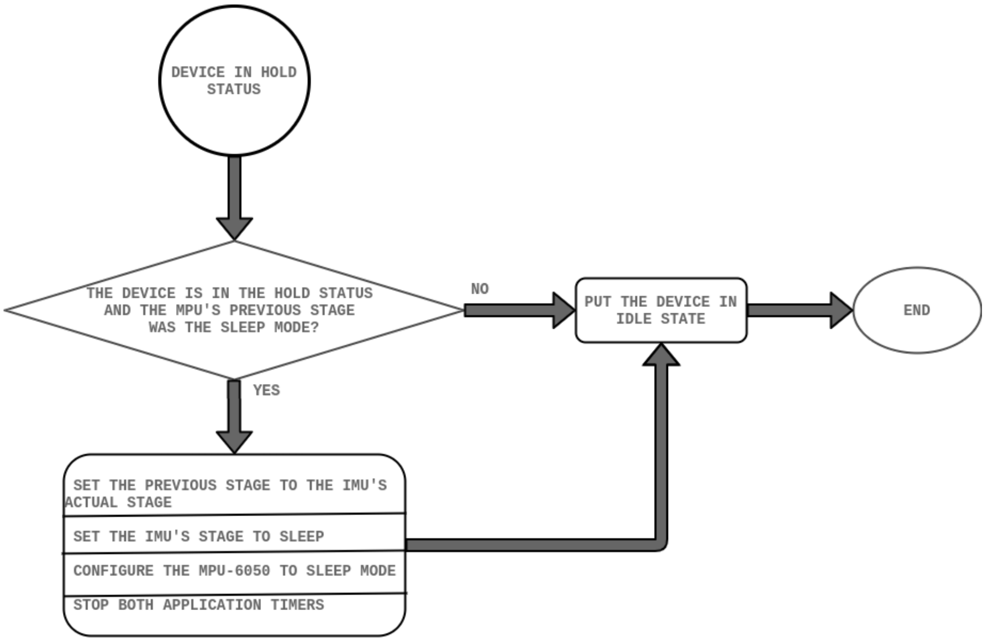


Figure 4.57: Hold status check

actual stage, proceeding to set the IMU to sleep mode, and, consequently, stop the application timers. Finally, we put the device in an idle state. The whole process is shown in Fig 4.57.

4.5 Prototype

The system consists of a power section, including the battery charging circuit, and the voltage regulator, the SoC section, which is the nRF52832, the IMU section, consisting of the MPU-6050, and the force sensors section, including the sensor connectors and the signal conditioning associated with each sensor. The overview of system is shown in Fig 4.68

For the construction of the prototype, all of the components need to be aglomerated to produce the desired output. Since the system needs to be autonomous, and as small as possible, the battery chosen was a coin cell, namely a LIR/CR 2032, which has 20 mm of diameter, and 3.2 mm of height. The device can work with either a CR 2032, which is a non-rechargeable coin cell, or an LIR 2032, which is a rechargeable lithium coin cell. If the CR 2032 is used, then the USB charging circuit must not be used.

4.5.1 nRF52832

For the nRF52832, the general design guidelines from Nordic Semiconductor will be followed. First, we need to define the package used, which in this

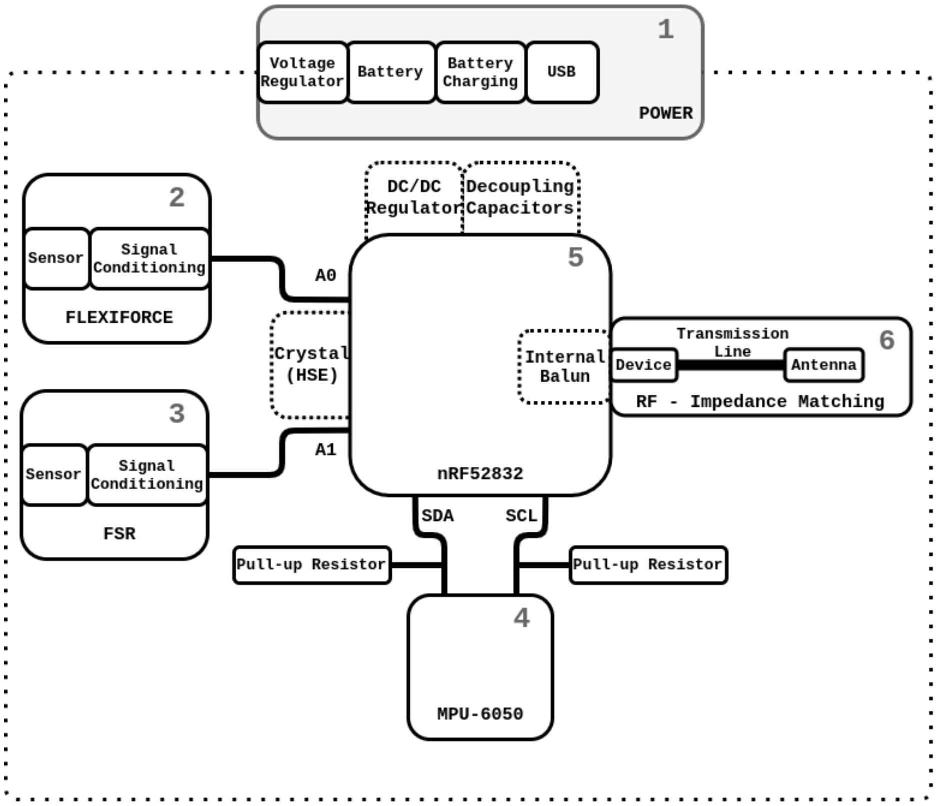


Figure 4.58: Prototype

case is the QFAA package, and what type of internal regulator will be used. The nRF52832 has two internal regulators: DC/DC and LDO. The default regulator is the internal LDO, however, the DC/DC will reduce current consumption, even though it requires an external LC filter. This regulator is enabled through the DCDCEN register.

Additionally, some decoupling capacitors for internal regulators are used. For example, DEC1 serves for a digital supply 0.9 V regulator, DEC2 is for a 1.3 V regulator (is left unconnected for QFN packages), DEC3 is for the power supply decoupling, and DEC4 is the input from the DC/DC regulator.

The main power supply also needs decoupling capacitors. Each V_{DD} pin, three in total, should have a capacitor, one with higher capacitance (4.7 µF), and the others with lower capacitance (100 nF).

The nRF52832 also requires a 32 MHz external crystal. At this point, we have to consider the capacitive loading of the crystal, which may influence the crystal oscillator frequency, causing problems to the operation of the device. To ensure that the crystal oscillates at the specified frequency, we need to make sure that the passive crystal load circuitry provides the C_L that is specified for the crystal. Furthermore, the choice should tend to crystals with lower C_L values since it gives a lower current consumption in the on-chip crystal oscillator. The type of crystal used is a 'parallel load compensated crystal', so we need to calculate the value of the crystal load capacitors, which is given by:

$$C_1 = C_2 = 2 * C_L - C_{PIN} - C_{PCB} \tag{4.41}$$

where C_L is the load capacitance of the crystal, C_{PIN} is the capacitance of the input pins of the nRF device, and C_{PCB} represents the parasitic capacitances associated with the PCB. Assuming the C_L as 8 pF (as for the crystal datasheet), and C_{PIN} + C_{PCB} around 4 pF (as for the application note), then it results in crystal load capacitors of about 12 pF.

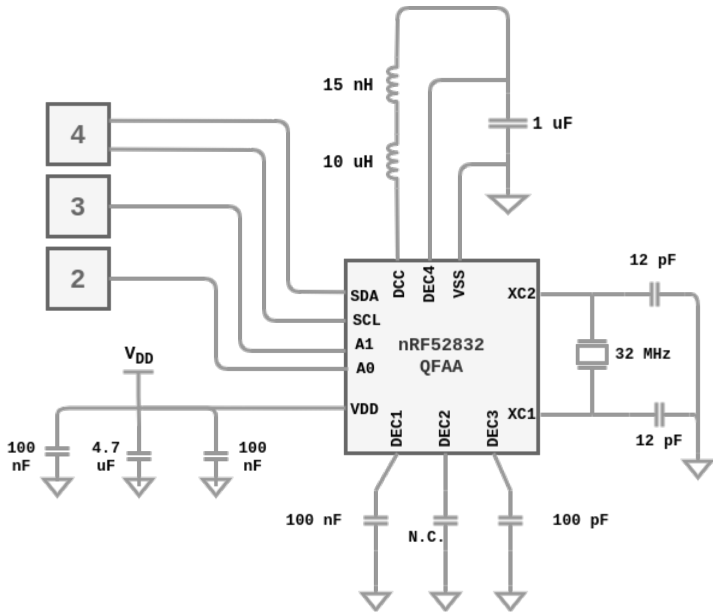


Figure 4.59: nRF52832 schematic

4.5.2 MPU-6050

The integration of the MPU-6050 also requires some external components to be present, which can be analyzed in Fig 4.60. Note that the I^2C protocol is used to communicate with this IMU. Let's analyze them:

- *CLKIN*: this is an optional external reference clock input. If it's unused, which is, in this case, the pin should be connected to the ground.

- *VDDIO*: this pin defines the logic levels of the device. It can either be 1.8 V ± 5 % or VDD. In this situation, VDD is used. A bypass capacitor is introduced of around 10 nF.

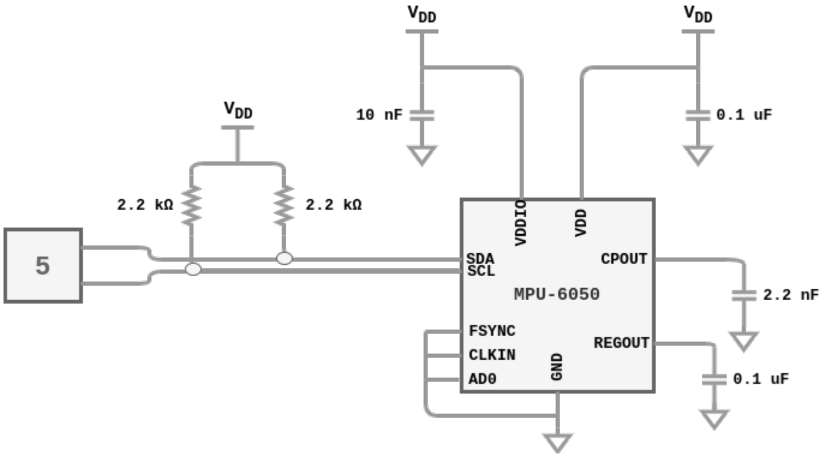


Figure 4.60: MPU-6050 design

- *VDD*: Represents the power supply voltage and the digital I/O supply voltage.
- *GND*: Power supply ground.
- *SDA*: I²C serial data (SDA). It needs a pull-up resistor.
- *SCL*: I²C serial clock (SCL). It needs a pull-up resistor.
- *FSYNC*: Frame synchronization digital input. If it is not used it should be connected to the ground.
- *AD0*: It defines the slave address for the I²C protocol. It is useful because it allows more than one MPU-6050 on the same I²C bus.
- *CPOUT*: Charge pump capacitor connection. The on-board charge pump generates the high-voltage required by the MEMS oscillator.
- *REGOUT*: Regulator filter capacitor connection.

4.5.3 Battery Charging

To charge the battery present in the circuit, a charge management controller, such as MCP73831, is used. This device provides a constant-current/constant-voltage charge algorithm. The charging current is set through an external resistor and can be adjusted as required by the application. Important to note that the charging is achieved through the USB connector, namely a micro B. The schematic is shown in Fig 4.61.

The V_{DD} and V_{SS} pins correspond to the power supply. A pull-down resistor is introduced to set the V_{DD} to the ground when the USB is unconnected. It

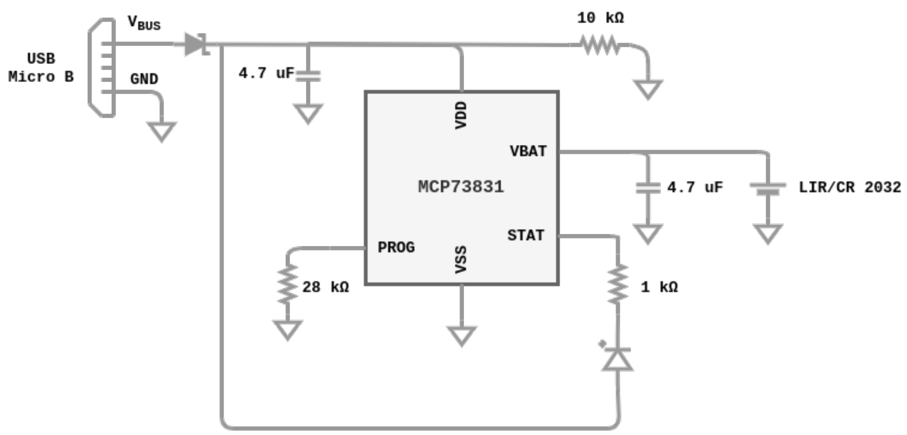


Figure 4.61: Battery charging design

allows the device to enter a Power-Down mode and prevents draining from the battery. The STAT pin represents the charge status output and is used to indicate the different stages of the charging process. In this case, a LED is turned on when the battery is charging. The V_{BAT} is the battery charge control output, and it's connected to the positive terminal of the battery providing a constant current and voltage regulation. The PROG pin is connected to an external resistor, which will set the charging current of the battery, and it's determined by:

$$I_{REG} = \frac{1000V}{R_{PROG}} \tag{4.42}$$

where I_{REG} is the charging current in mA, and R_{PROG} is the programming resistor in kΩ. For a LIR2032 battery, the common charging current is about 35 mA, which results in a R_{PROG} of 28 kΩ.

4.5.4 Voltage Regulation

An LIR 2032 battery, as it discharges, its output voltage will vary. Furthermore, a fully-charged battery reaches 3.7V, which is not acceptable for our circuit. There is a need to regulate this voltage to a fixed voltage, which can be used for all the other devices. The TPS61220 is a boost converter with an operating input voltage ranging from 0.7 V to 5.5 V. The output voltage, can also be adjusted from 1.8 V to 6 V. There is also a switch that can turn off the system, as observed in Fig 4.62.

The V_{IN} represents the unregulated input voltage. An inductor must be connected between V_{IN} and L pins to ensure the proper device operation. An inductance of 4.7 µH is a typical value. Input and output capacitors of 10 µF are recommended. Though the FB pin, the output voltage is adjusted according to the following formula:

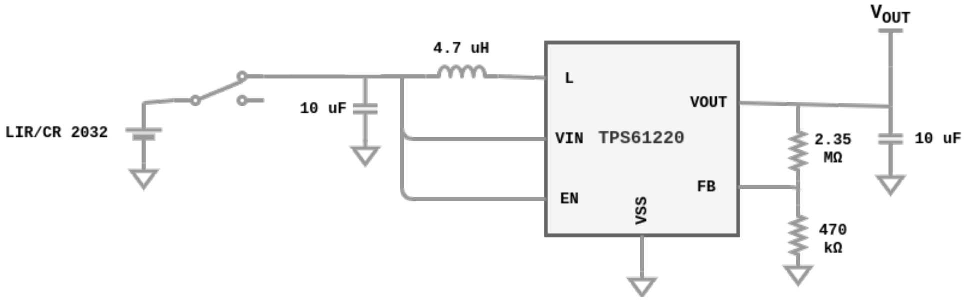


Figure 4.62: Voltage regulator design

where R_1 and R_2 form a voltage divider, V_{OUT} is the desired output voltage, and V_{FB} is the FB pin voltage. For this device, V_{FB} is around 500 mV. R_2 must be lower than 500 kΩ, ensuring that I_{FB} does not exceed 0.01 µA. If V_{OUT} is 3 V than R_1 and R_2 can be set to 2.35 MΩ and 470 kΩ, respectively.

4.5.5 Antenna

Since the device will be using BLE for data transmission, an antenna needs to be designed. As the device needs to be as compact as possible, the antenna used is a PCB antenna, instead of a chip antenna, for example. The problem is both the chip, antenna, and transmission line need to match to 50 Ω otherwise, there will be reflections of the signal in the transmission line, resulting in a loss.

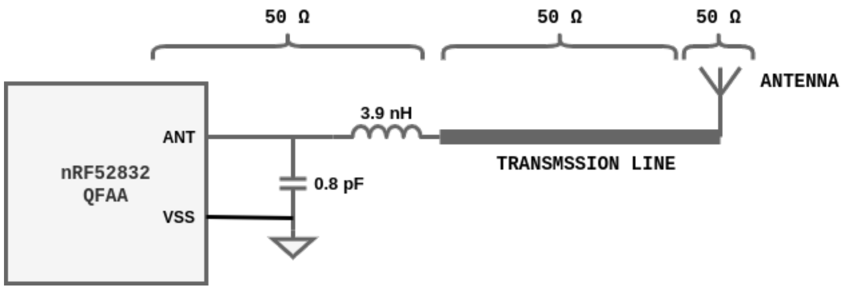


Figure 4.63: Antenna design

Starting with the nRF52, it is not matched to 50 Ω by default, but it has an internal balun[1] which facilitates this task, using only two components. The inductor and capacitor will also form a low-pass filter, suppressing harmonics. Note that these are typical values that may need to be changed to maximize performance

[1] **Balun** - is a device that functions as a transformer, converting a balanced line to an unbalanced one, and vice-versa.

The transmission line impedance depends on the manufacture, mainly because variables like the dielectric constant and the height of the substrate affect its impedance. Normally, the manufacture makes available an impedance calculator, which serves to calculate the necessary track width. In this case, the result was 0.293 mm.

$$w = \frac{7.48 * h}{e^{z_0 \frac{\sqrt{\epsilon_r + 1.41}}{87}}} - 1.25 * t \tag{4.43}$$

where, Z_0 is the desired impedance, w is the track width, t corresponds to the track thickness, h to the dielectric thickness, and ϵ_r to the dielectric constant.

The antenna used for this system was the Texas SWRA117D, which is an inverted F antenna (IFA) or, more specifically, a meandered inverted F antenna (MIFA). A MIFA is a microstrip antenna developed to be small but to provide wideband performance. These antennas are folded to decrease the overall size of the resonant. This MIFA only occupies 15.2 x 5.7 mm of space and is designed to match the 50 Ω. This approach was followed since the construction of an antenna is a topic on its own. Even the antenna position in relation to the ground plane can have a major impact on the antenna performance. The dimensions of the antenna are shown in Fig 4.64.

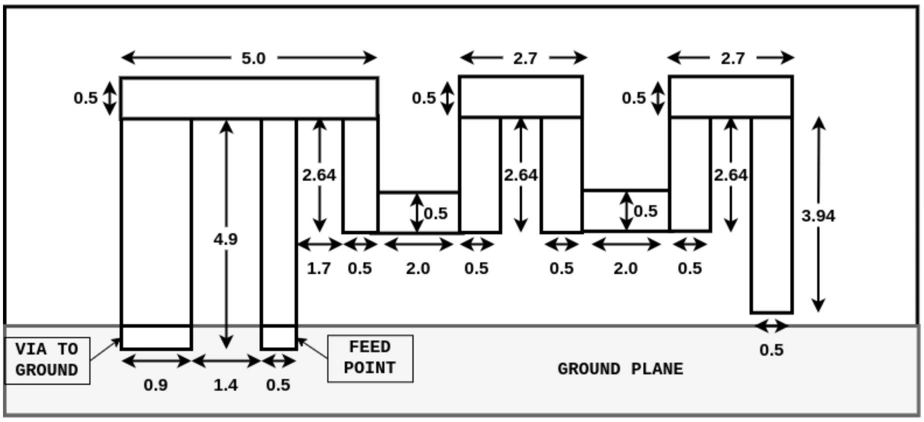


Figure 4.64: Inverted F Antenna (dimensions in mm)

Furthermore, we've tried to follow some of the guidelines for implementing a PCB antenna, which dictates the performance of the antenna. Some of these guidelines are related to manufacturing, like the board thickness or the antenna being covered with soldermask. Also, it is recommended the use of FR-4 substrate for this range of frequencies.

PCB antennas require the use of solid ground planes, namely on the first inner layer, and there should be a copper-free zone underneath the antenna.

Inverted F antennas have a feed arm, which is connected to the transmission line, and a shorting arm, which should be connected to the ground plane with ground vias. The transmission line must be matched to 50 ohms, as previously described. Additional stitching vias around the ground plane near the antenna provide a better ground reflection for the antenna.

4.5.6 PCB Layout

For this device, a 4-layer board is used, allowing the assignment of two of the layers for signal routing, the external layers, and the two inner layers to power, more specifically, ground and +3 V. One of the advantages of using this disposition is the ease of routing, making power traces unnecessary, and power distribution. Also, it is one of the recommended for PCB antennas.

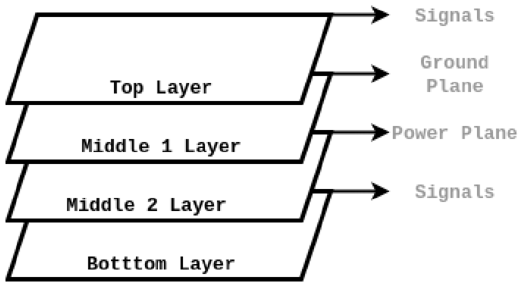


Figure 4.65: Layers of the PCB

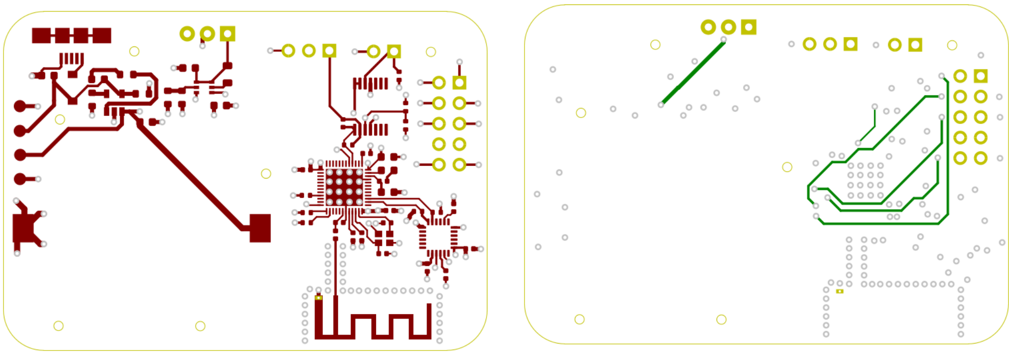
The prototype size is approximately 42 x 60 mm. The size is majorly impacted by the location of the battery holder, but it can be furtherly optimized. For the PCB layout, some cautions were taken, such as the placement of the bypass and decoupling capacitors near the respective pins, the crystal placement, among others.

The Top Layer is presented in Fig 4.66a. The majority of the signals are routed on this layer. The bottom layer, Fig 4.66a, is used for routing the signals that can't be routed in the top layer. Most of the signals are from the Serial Wire Debug (SWD) connector, which is a low-frequency signal.

The first inner layer consists of the ground plane. The ground plane is quite large and robust, making it less susceptible to interference, as seen in Fig 4.67a. The second inner layer, in Fig 4.67b, consists of the power plane, more specifically, the 3 V. This allows for good power distribution.

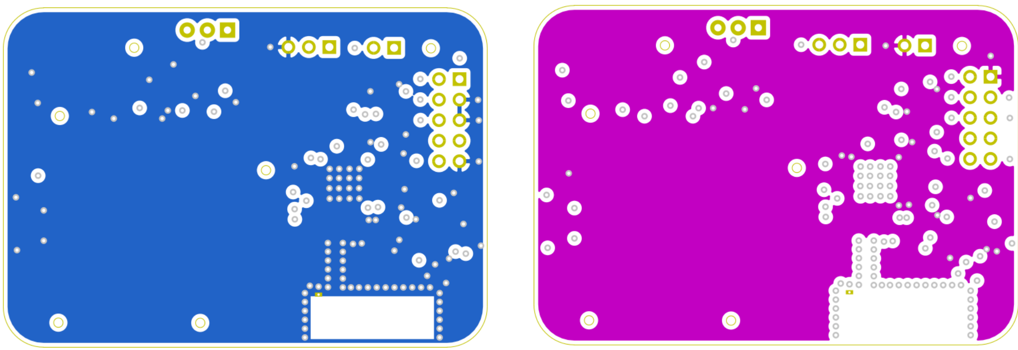
The result of the process is shown in Fig 4.68, as it demonstrates that the prototype was designed successfully.

The prototype with the force sensors is shown in Figure 4.69. The Flexiforce is the sensor on the left, and the FSR is the sensor on the right.



(a) Top layer (b) Bottom layer

Figure 4.66: External layers



(a) Inner 1 layer (b) Inner 2 layer

Figure 4.67: Internal layers

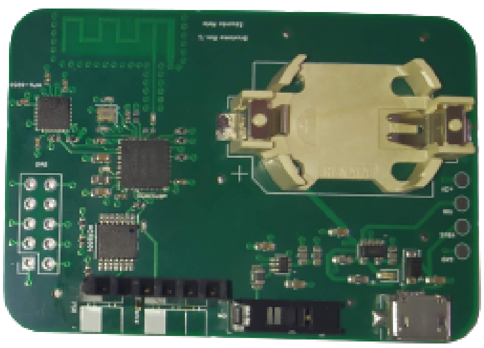


Figure 4.68: Prototype

Figure 4.69: Prototype with force sensors

Final Considerations

In this chapter, we've shown how the system was developed. It included how could we estimate the patient's sleeping position, how could we read the force sensors, how the firmware works, how the custom Service for the BLE was constructed, and the construction of the prototype.

The system is flexible, mainly because it allows the user to change its behavior. Three Characteristics were implemented, one for the force sensor values, one for the accelerometer values, and another for the gyroscope. If notifications are enabled only on the accelerometer, the system is in low-resolution mode, only sampling the accelerometer at 2 Hz. When both accelerometer and gyroscope's notifications are enabled, the system enters in high-resolution mode, sampling them at 20 Hz.

A similar process is applied to the force sensors. In normal operation, the device is in low-resolution mode, sampling at 7 Hz. When a peak is detected, meaning that it surpassed a defined threshold, the device enters in high-resolution mode, sampling at 50 Hz. When a number of consecutive samples do not exceed the threshold, the device returns to low-resolution mode, and, if it's connected, it will put the device in Hold Status, stopping the sampling of the accelerometer and/or the gyroscope.

Chapter 5

System Validation

Starting with the BLE, we can observe how the configurations previously defined influence the system. The following figures are from nRF Connect, which is one of the applications made available by Nordic.

If the device is advertising, or, more specifically, in the peripheral role, the device may be discovered by other devices. The Fig 5.1 shows some device parameters, such as the device name and its MAC address. The name is the same configured on the application, "*BRUXISMO*", and the MAC address is *E1:4A:57:BF:8A:43*, which is represented by a unique 48-bit value, and it serves to identify a BLE device. There are several types of MAC addresses, mainly due to security reasons, but in this case, the MAC address is a random static address. The address is randomized and stored by the chip manufacture and every time BLE is enabled, the address is used. Alternatively, we could set our random private address.

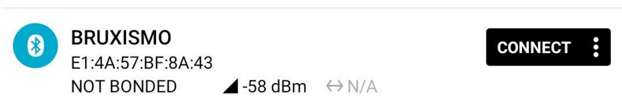


Figure 5.1: Device advertising

As for the advertising data, the Fig 5.2 shows some of the parameters. The most interesting to analyze are appearance settings, which in the application are defined to show appearance (the default is "Unknown"), and the flags *GeneralDiscoverable* and *BrEdrNotSupported*, which signifies LE General Discoverable and BR/EDR (Bluetooth Classic) is not supported, respectively. These flags are also configured in the application, namely on the advertising data initialization.

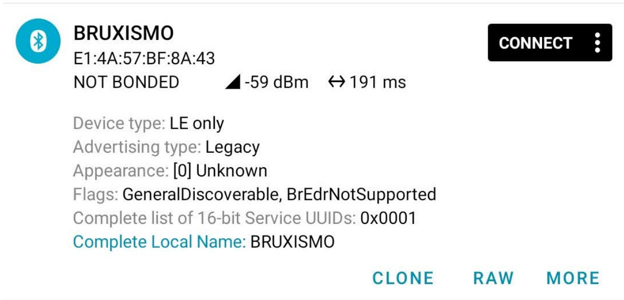


Figure 5.2: Device advertising info

By analyzing the raw data of the PDU, some pieces of information are retrieved.

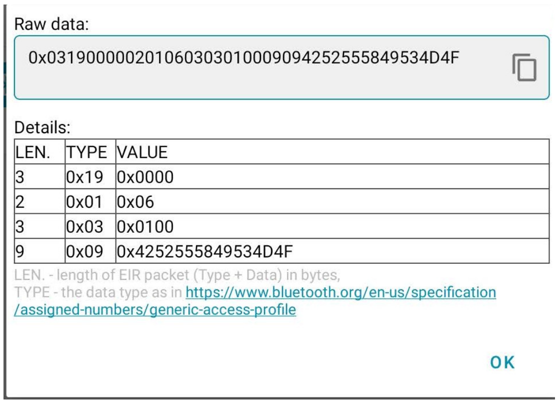


Figure 5.3: Advertising raw data

Let's start with the PDU header, which has four main fields. The PDU type defines the payload's structure, and each type has a unique function. The PDU type has a value of 0, indicating that it's an *ADV_IND* packet type. *ADV_IND* is a legacy advertising PDU and supports connectable, scannable, and undirected advertising. It's used when the peripheral first powers up and tries to connect to any device. The *ChSel* indicates if the advertiser supports LE Channel Selection Algorithm #2 feature if this field is set to 1. The *TxAdd* set to 1 indicates that the device's address is random, as we've seen0o previously. *RxAdd* is the same thing, but for the target's address, however it does not apply for a *ADV_IND* packet type.

The PDU payload consists of two main fields: *AdvA* and *AdvData*. The *AdvA* holds the device's address, and the *AdvData* holds the actual data, such as manufacture specific data, the device name, among others. For example, the last eight octets represent the device name.

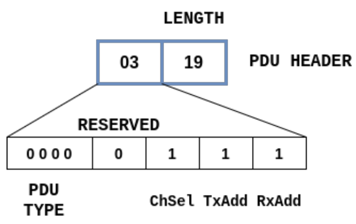


Figure 5.4: PDU Header

Next, we can see some advertising statistics, which enables us to calculate the advertising interval. As in Fig 5.5, the advertising interval is 189 ms, which is quite close to the advertising interval set in the application, corresponding to 187.5 ms. The advertising interval has a step of 0.625 ms, so to achieve the 187.5 ms, a value of 300 was set.

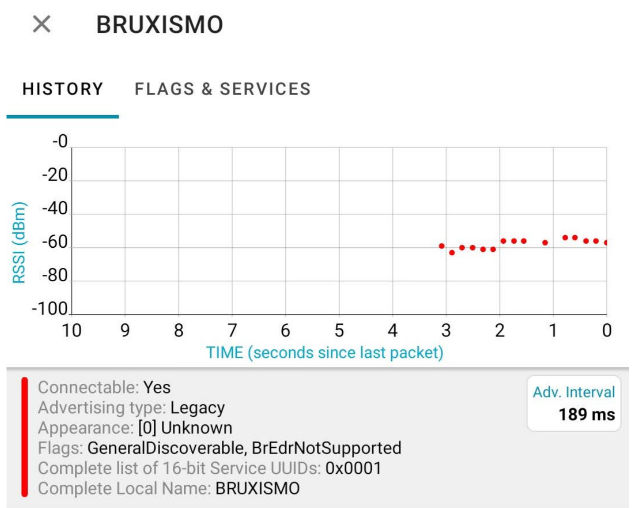


Figure 5.5: Advertising interval

After the service discovery, we can see that 3 services were encountered. The *Generic Access* service is related to the GAP and is mandatory by the BLE specification. Although the *Generic Attribute* service is not required by the BLE specification, the softdevice, which implements the BLE stack, it also implements this service. The *Unknown Service* is the service we've created. Note that both default services hold 16-bit UUIDs, meaning that they are part of the Assigned Numbers of the Bluetooth SIG.

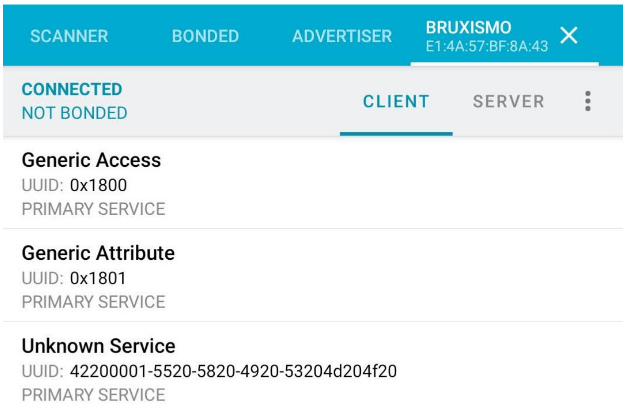


Figure 5.6: GATT Profile

The *Generic Access* service has four inherent characteristics. The **Device Name** and **Appearance** were already mentioned, and their value corresponds to what was analyzed. The **Peripheral Preferred Connection Parameters** holds the connection parameters defined in the application: the connection interval, defined between 15 and 25 ms; the slave latency, defined to 500; the supervision timeout multiplier, set to 3200. This value represents 10 ms steps, resulting in a supervision timeout of 32 s. Again, these values are only an indication to the central, not representing the definite values of the connection parameters. The **Central Address Resolution** defines if the device support privacy with address resolution, and stated in the value, it does.

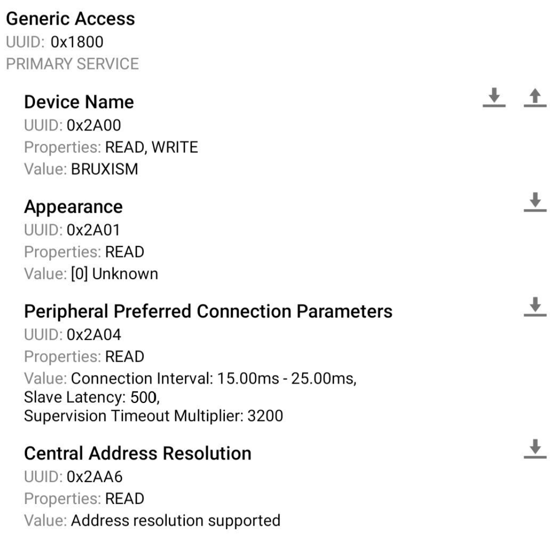


Figure 5.7: Generic Access

As for the Unknown Service or, more specifically, the Brux Service, we can see that it is the defined UUID, and it's composed of three characteristics. One thing to note is the are the properties, which are defined as specified in the application.

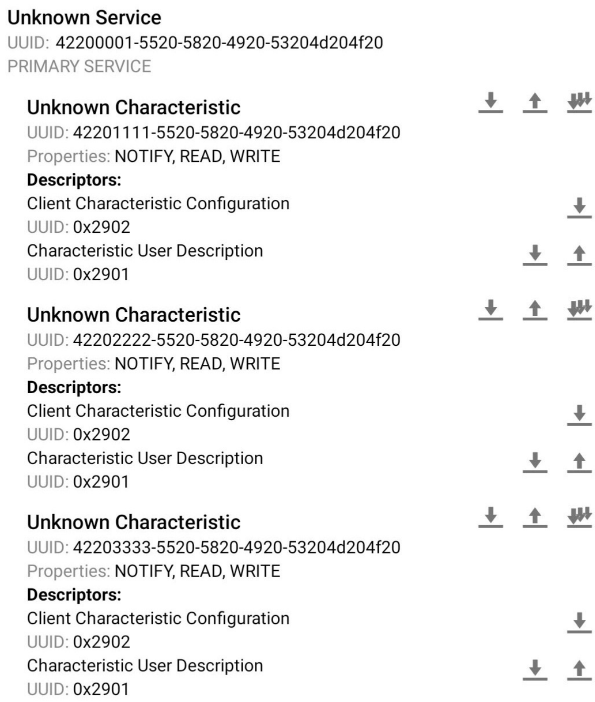


Figure 5.8: *Brux* Service structure

The first characteristic is relative to the force values, which is concluded by its UUID (0x1111 on the 3rd and 4th most significant bytes). The Value is 4 bytes of size, as defined in the application, and the CCCD has notifications enabled. The User Description is also present as "FORCE VALUE", which a good practice.

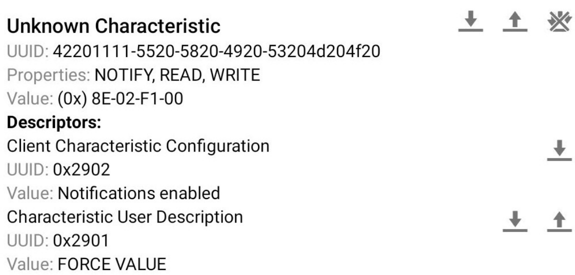


Figure 5.9: Force Characteristic

The next characteristic is relative to the accelerometer. The UUID validates that statement, the Value is 6 bytes long, and notifications are enabled. The User Description is set to "ACCEL VALUE".

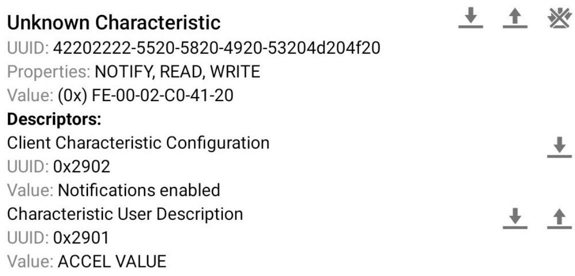


Figure 5.10: Accelerometer Characteristic

Lastly, there is a gyroscope characteristic. Similarly to the accelerometer characteristic, the Value is 6 bytes long, and notifications are enabled. The UUID corresponds to the expected value, and the User Description is set to "GYRO VALUE".

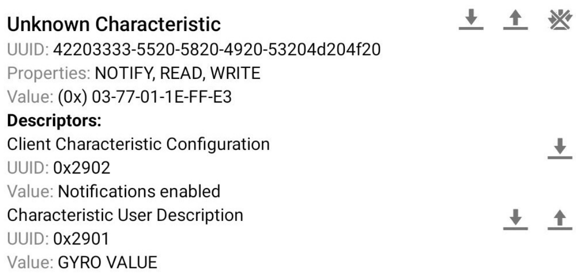


Figure 5.11: Gyroscope Characteristic

5.1 External Device Application

To gather the data sent from our device, a Python script was developed, which is responsible to establish a connection with it, show the data received and processing it. The Figure 5.12 shows the core functionality of the script.

The result of this process is demonstrated in Fig 5.13. The value read from the characteristic corresponds to the initial value set on the application-side. The first characteristic is relative to the Force values, the second to the Accelerometer values, and the third to the Gyroscope values.

We can enable notifications to those characteristics. The data received, from the force and accelerometer characteristics, is shown in Fig 5.14.

5.1.1 Estimation of the orientation

To balance the system between power consumption and accuracy, we've implemented two modes of operation: the low-resolution mode, with sampling

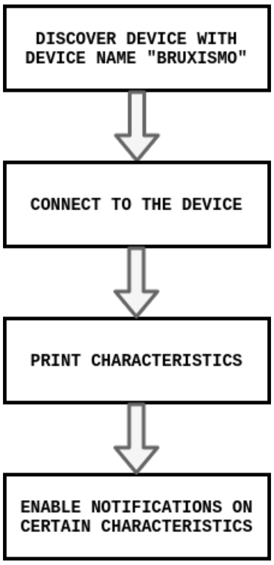


Figure 5.12: Main functionality of the script

```
    DEVICE FOUND  -> E1:4A:57:BF:8A:43

CHARACTERISTIC

    UUID  -> 42201111-5520-5820-4920-53204d204f20

    VALUE  -> b'ffffffff'

CHARACTERISTIC

    UUID  -> 42202222-5520-5820-4920-53204d204f20

    VALUE  -> b'000000000000'

CHARACTERISTIC

    UUID  -> 42203333-5520-5820-4920-53204d204f20

    VALUE  -> b'000000000000'
```

Figure 5.13: Characteristics found

```
Force -> 1: (341), 2: (255)
Force -> 1: (334), 2: (259)
Force -> 1: (338), 2: (247)
Force -> 1: (342), 2: (260)
Force -> 1: (341), 2: (247)
Force -> 1: (339), 2: (247)
Force -> 1: (339), 2: (254)
Force -> 1: (338), 2: (246)
Force -> 1: (337), 2: (251)
Accelerometer -> X: -0.004150390625, Y: 0.0556640625, Z: 1.02880859375

Force -> 1: (348), 2: (250)
Force -> 1: (406), 2: (258)
Force -> 1: (542), 2: (253)
Force -> 1: (672), 2: (256)
Force -> 1: (794), 2: (256)
Force -> 1: (866), 2: (246)
Force -> 1: (888), 2: (254)
Force -> 1: (875), 2: (253)
Accelerometer -> X: -0.00341796875, Y: 0.0576171875, Z: 1.017578125
```

Figure 5.14: Data received through notifcations

frequency (f_s) of 2 Hz, which is active when only one of the notifications are enabled, and the high-resolution mode, with sampling frequency of 20 Hz, which is active when both notifications are enabled. Furthermore, when using the accelerometer low-power mode of the MPU-6050, the device only draws 0.5 mA. When the gyroscope is enabled, the energy consumption rises to 3.6 mA (using only the gyroscope), which corresponds to an increase of more than seven times.

In low-resolution mode, the sampling rate is due to constraints to power consumption, but also because the system doesn't need an exact position at every point in time. For example, in a control system, this sampling rate would be unacceptable since it would "lose" a lot of information. In high-resolution mode, the sampling rate is more acceptable, and one technique that may be used to determine the angles is the complementary filter.

Low-Resolution Mode

Let's see how the system behaves in different situations. The first test is to see the values retrieves from the IMU, more specifically, from the accelerometer when the device is in low-resolution mode (f_s = 2 Hz) and at rest, meaning that the device is being subjected only to the gravitational force. As we observe from Fig 5.15. the accelerometer shows good stability in the long term, presenting no drift. The yellow line, representing the z-axis, shows that the device is laying flat, with the gravitational acceleration focusing majorly on that axis.

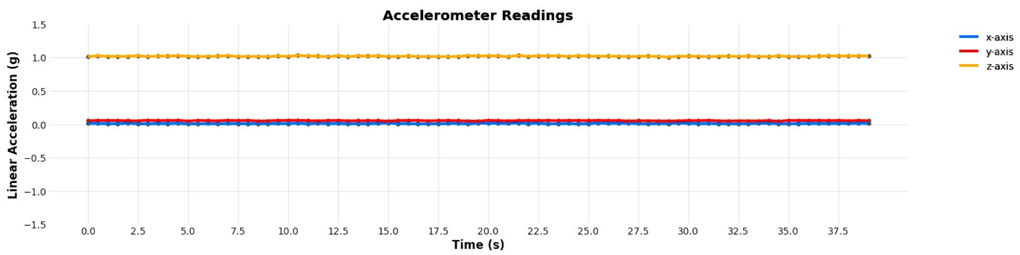


Figure 5.15: Accelerometer measurements

Now we execute the same test, with the same conditions, but for the gyroscope. The results presented in Fig 5.16 show that the values oscillate considerably, which makes this particular case not reliable.

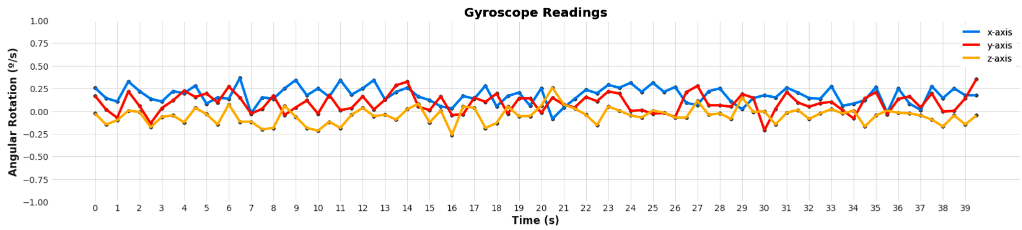


Figure 5.16: Gyroscope measurements

Now we can try to implement the complementary filter, while using the low-resolution mode. Let's try different values for τ and analyze how the system behaves. The conditions are the same for all tests: the device is at rest, with the gravitational acceleration pointing in the z-axis, for three minutes. The results are presented in Table 5.1.

Table 5.1: Complementary Filter with different values (at rest)

τ	**Roll**	**Pitch**	**Yaw**
1	11.90 º	11.37 º	-0.65 º
0.95	5.95 º	4.43 º	-1.03 º
0.7	-0.73 º	3.14 º	-4.60 º
0.5	-1.05 º	3.09 º	-6.04 º
0.2	-1.12 º	3.35 º	-3.43 º
0	-1.34 º	2.89 º	-5.13 º

For τ **= 1**, the accelerometer data does not count on the various angle calculations, and the results are what we expected, as roll and pitch angles drift from their actual position due to some bias in the readings and also potential noise. The yaw remains, surprisingly, constant. Note that the accelerometer does not affect the yaw angle in any test.

For τ **= 0.95**, the accelerometer measurements start to influence the overall estimate, and the results show immediate results. Both the roll and pitch angles show an improvement in the amount of drift, more than two times less. The yaw angle increases, but not as much as expected.

For τ **= 0.7**, the roll angle shows an acceptable value, given that the surface is not perfectly aligned and some offset on the axis of both the accelerometer and gyroscope. The pitch angle also shows an improvement, but not as much as the roll. The yaw, as expected, shows a significant variation.

For τ **= 0.5**, the accelerometer and gyroscope measurements are equally considered on the angle estimation. The roll and pitch angles continue to drift to a more accurate estimate. The yaw angle shows the contrary behavior.

For τ **= 0.2**, the accelerometer measurement becomes the main component of the calculation. The roll value continues to improve, although, at a significantly lower rate, the pitch shows a worse result, but not that significant, and the yaw also improves (not related to τ).

For τ **= 0**, the accelerometer measurement becomes the only component on the roll and pitch angle estimate. The results show that is the best configuration so far. The yaw angle remains because the gyroscope is responsible for its estimation.

The next experiment is to see how the system will respond when there is movement. The idea is to start with the device at rest, move the device, and return to approximately the same position. After the device recovers, we check the angle estimates. Note that the start and end position won't be precisely the same for every iteration and that the yaw angle is determined only with the gyroscope. The results are presented in Table 5.2.

Table 5.2: Complementary Filter with different values (moving)

τ	**Roll**		**Pitch**		**Yaw**	
	Start Angle	**Estimated Angle**	**Start Angle**	**Estimated Angle**	**Start Angle**	**Estimated Angle**
1	0.64 º	136.74 º	2.96 º	105.04 º	-1.04 º	-41.38 º
0.95	0.76 º	-113.90 º	3.14 º	5.83 º	-0.66 º	27.07 º
0.7	1.13 º	0.22 º	3.65 º	4.93 º	-0.54 º	-89.24 º
0.5	0.68 º	0.45 º	3.01 º	3.10 º	-1.13 º	24.77 º
0.2	0.74 º	-0.35 º	3.05 º	3.23 º	-0.78 º	-182.74 º
0	0.95 º	0.13 º	3.36 º	3.34 º	-0.98 º	13.64 º

Analyzing the results, we conclude that using the gyroscope measurements as the main component of the complementary filter will lead to unprecise results. From τ = 0.7 and lower, the roll and pitch angles start to show some improvements because, even with movement, the estimated angles approximately correspond to the start values. So, for the low-resolution mode, the accelerometer values are enough to estimate the patient's sleep position.

High-Resolution Mode

Next, we test the high-resolution mode. The device enters in this mode when notifications of the accelerometer and gyroscope are enabled, changing the sampling frequency to 20 Hz. Fig 5.17 shows the roll and pitch angles using only the accelerometer values, only the gyroscope values, and using the complementary filter with $\tau = 0.95$. The device is at rest in this case. What we observe is that the accelerometer angles are noisy, but they don't drift. The gyroscope angles show the opposite behavior since they show good stability, but they start to drift. Using the complementary filter, we can have the best of both. For example, the roll angle calculated with the complementary filter (purple line) clearly shows the benefits of using this approach by preserving the stability of the gyroscope angle (green line) and the linearity of the accelerometer angle (red line). Note that the gyroscope angle already shows the drift.

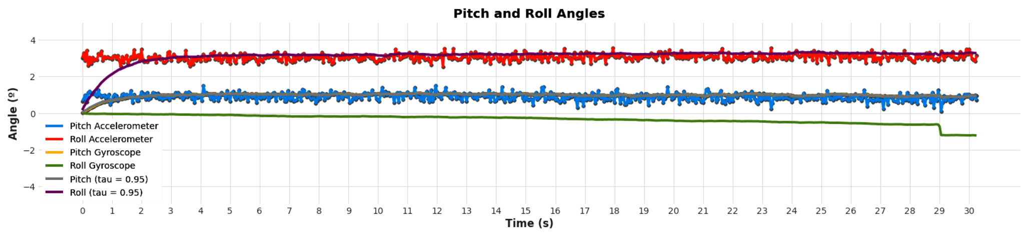


Figure 5.17: Complementary Filter measurements

The next step is to see how the system performs when a given rotation is applied. For this, we use the high-resolution mode and try to apply a rotation around an axis. The device first starts at rest, then the rotation is applied, and the device goes back to the initial position. In this test, the rotation is applied to the x-axis, resulting in a variation in the pitch angle. The results presented in Fig 5.18, show that the device starts at rest, and there is the rotation around the x-axis, which increases or decreases the pitch angle (blue line). Furthermore, when the device returns to the initial position, we can see that the angle values tend to the initial values.

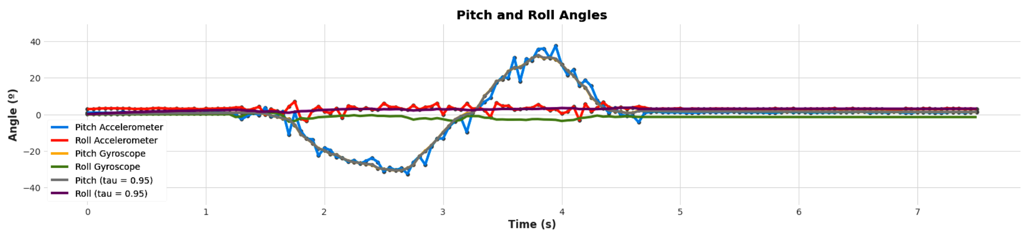


Figure 5.18: Complementary Filter measurements - Pitch Angle

We then apply the same process on the y-axis. In this situation, we will observe a change in the roll angle. Although some "spikes" exist from the accelerometer values (red line), we can see that the resulting values are smooth due to the gyroscope stability. When the device returns to its initial position, the roll and pitch angles also return to their initial values.

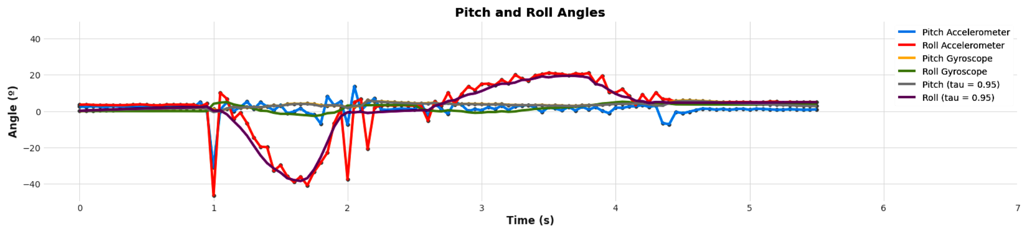


Figure 5.19: Complementary Filter measurements - Roll Angle

The following test tries to simulate a situation where the patient starts at a given position and rotates to another. Then, the system transmits no data because there was no force exerted between the teeth to trigger the notifications. Meanwhile, the patient returns to its initial position. The goal of this test is to see if the system can "recover" from this lack of information and determine the patient's sleep position. Figure 5.20 shows the process. Focusing on the pitch angle (grey line), we can see that it starts at 0º and it stabilizes at around -55 º. Then, no data is received for a while, and, when we receive new data, we can observe that the accelerometer (blue line) indicates that the pitch is back at 0 º. We then observe that the pitch angle calculated by the complementary filter has a transition period, but, after a while, it will converge to 0 º.

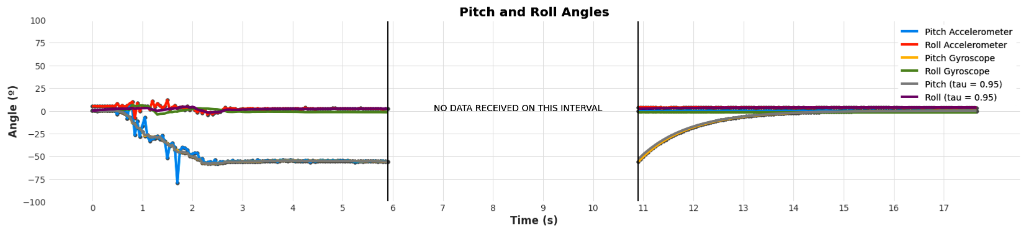


Figure 5.20: Complementary Filter - No data received

Through the pitch and roll angles, we can roughly know the sleeping position of the patient, as shown in Fig 5.21.

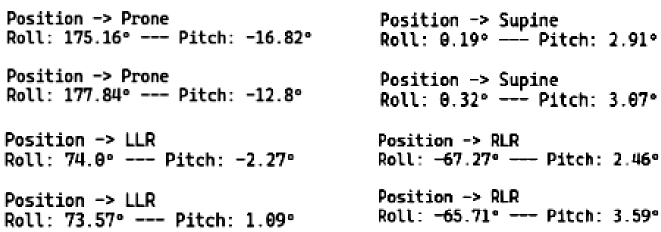


Figure 5.21: Different positions estimation

In the Figure 5.22, we present the regions for every position. In the x-axis, we have the pitch angle, which oscillates between -90 and 90 °, and, on the y-axis, we have the roll angle, oscillating between -180 and 180 °. One thing to note is that these regions were define based on our understanding of the device's functioning and can be changed. For this graph, 13000 samples were used.

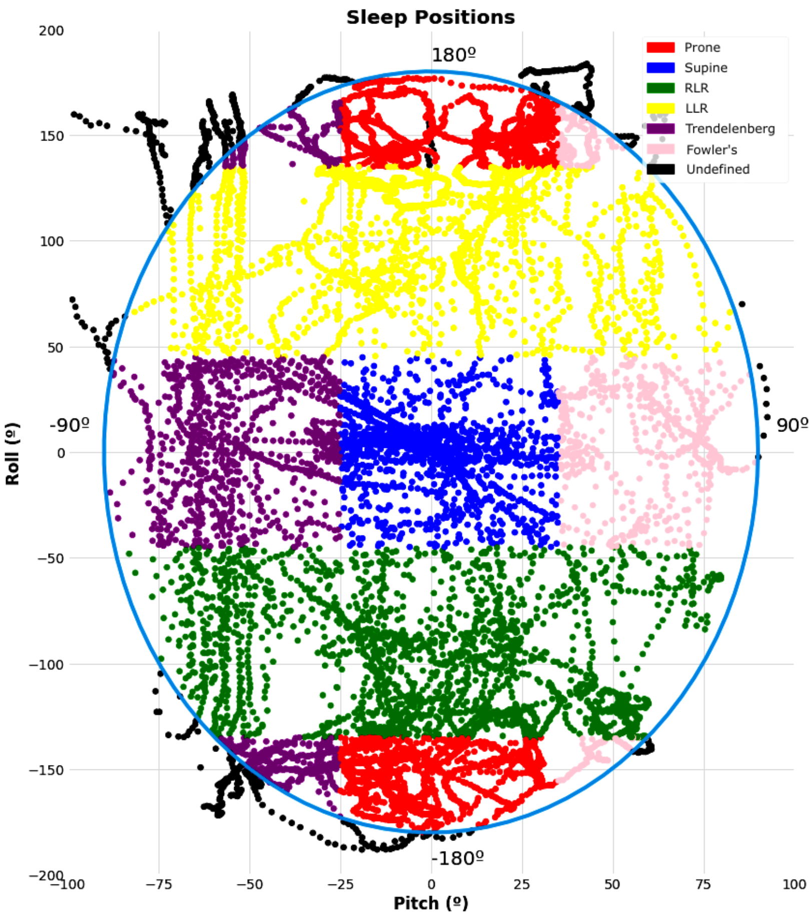


Figure 5.22: Different positions

Furthermore, we can observe that there are opposite regions. For example, the prone (red zone) and supine (blue zone) positions are on the opposite sides of the y-axis, which makes sense because, in the supine position, a person is facing upwards, and, in the prone position, a person is facing downwards. Other opposites are the LLR (green zone) and RLR (yellow zone), which also makes sense because, in the LLR position, a person is laying on its left, and, in the RLR position, a person is laying on its right, so we observe that those values are on the opposite side of the x-axis. We also have the Trendelenberg (purple area) position, which represents the state where the patient's head is facing downwards, and the Fowler's (pink area) position represents the state where the patient's head is facing upwards. The "Undefined" (black points) data corresponds to unreliable data and represents only 0.05% of the overall samples.

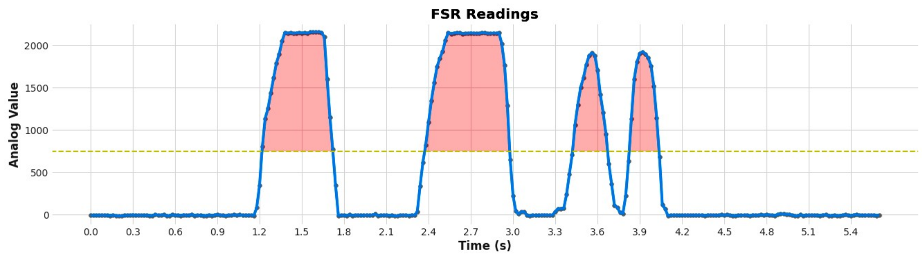


Figure 5.23: FSR measurements

5.1.2 Force Data Acquisition

As for the force readings, we have two sensors. When in low-resolution mode, they present a sampling rate of 7 Hz, which increases to 50 Hz when the device enters in high-resolution. The condition necessary is to detect a peak in the analog channels, surpassing the thresholds defined in the application. On the external device, we only have access to the data when the device is in the high-resolution mode because the device will only send data to the client when we suspect that a bruxism event is occurring, and, subsequently, the threshold of the force sensors is exceeded.

For the FSR, the values are shown in Fig 5.23. We can observe that, if no force is applied, the analog values are very close to 0, which is expected. Then, when a force is applied, we can see the values rising, even past the threshold (shown by the yellow line). The colored part corresponds to the values that surpass that threshold. This sensor has two problems: it doesn't have a linear response, and it has a low dynamic range. That's why we see the saturation when the force gets close to 2300.

The Flexiforce presents an offset of 0.5 V, which corresponds to the V_{REF} used in the signal conditioning of the sensor. As we observe in Fig 5.24, all the values are offset by that value, however, if we consider that our baseline, then we can see that, when no force is applied, the readings are stable. When force is applied, the analog values start to rise. The threshold value is represented by the yellow line, and the colored part corresponds to the values that surpass that threshold.

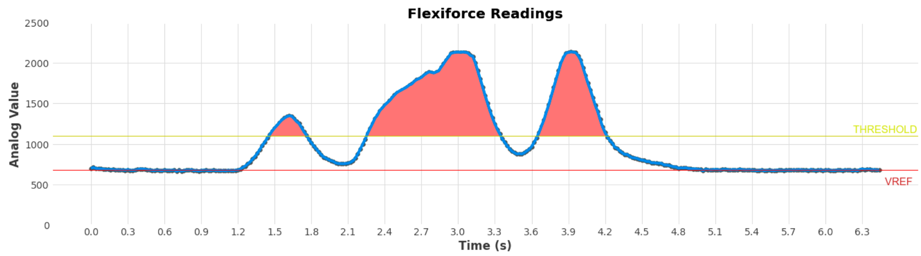


Figure 5.24: Flexiforce measurements

5.1.3 Practical Simulations

In this section, we'll try to simulate some of the situations that would exist in a sleeping environment. For that, we use the system as it would be implemented and simulate various positions and transitions in the bed.

The following Figure presents what we would consider a bruxism event. We can observe the force being exerted between the teeth and some movement on the roll angle, meaning that the patient has moved.

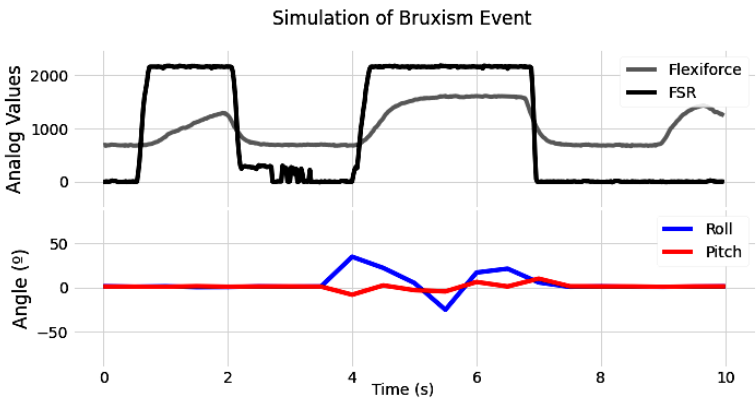


Figure 5.25: Simulation of Bruxism Event

Figure 5.26 shows a transition from supine to LLR. We can observe this by the change in the roll angle. We can see that, in the initial position (T = 0 s), both roll and pitch angles were close to 0 º, which indicates that the patient was in the supine position. Then, the roll angle starts to increase to values around 100º, and the pitch maintains the same value indicating that the patient moved to LLR.

In Fig 5.27, we observe the same process, but this time, the patient moves from a supine position to RLR. In this case, we see a variation in the roll angle, but in the opposite direction as the previous test, corresponding to a roll angle of -100 º.

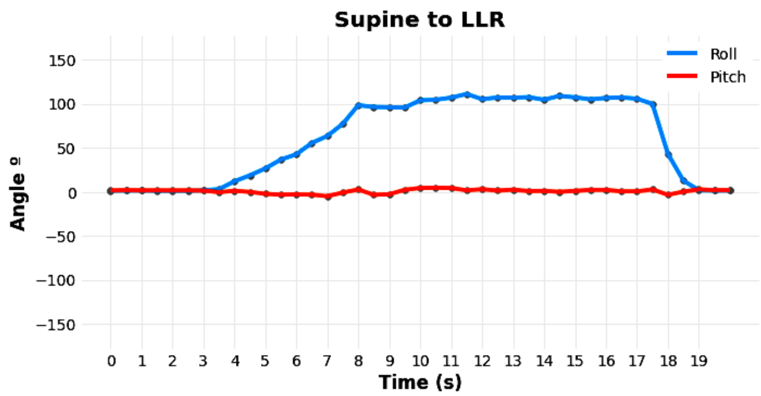


Figure 5.26: Transition from Supine to LLR

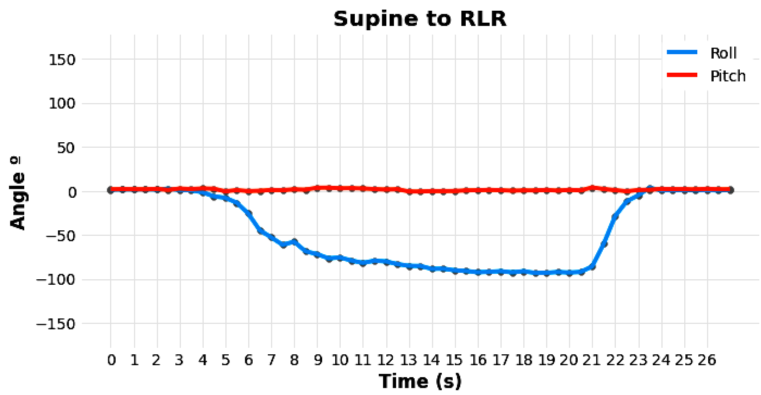


Figure 5.27: Transition from Supine to RLR

In the next test, the start position (T = 0 s) is different from the previous tests. As shown in Fig 5.28, the roll angle is around 100 º, and the pitch angle is about 0 º, so we can assume that the patient is in LLR. Then, we observe a transition from LLR to RLR, and that is reflected in the roll angle, going from 100 º to -100 º. The pitch suffers a bit of variation, which is normal when people transition from one position to another. After 2 s, there is another transition, this time back to LLR. Again, this transition is reflected in the roll angle, which goes from -100 º to 100 º.

Next, we observe a transition from supine to Fowler's position, which may have two interpretations: either the patient is sitting on the bed or is standing up. Unlike the other tests, in this case, the pitch suffers the most variation, going from 0 º to 90 º, as shown in Fig 5.29.

To further demonstrate the system's functioning, we've decided to simulate what would look like the data over some time, and the results are shown in Fig 5.30, in which the upper graph demonstrates the force sensors readings, and the lower graph represents the roll and pitch angles. There are 7 events in total,

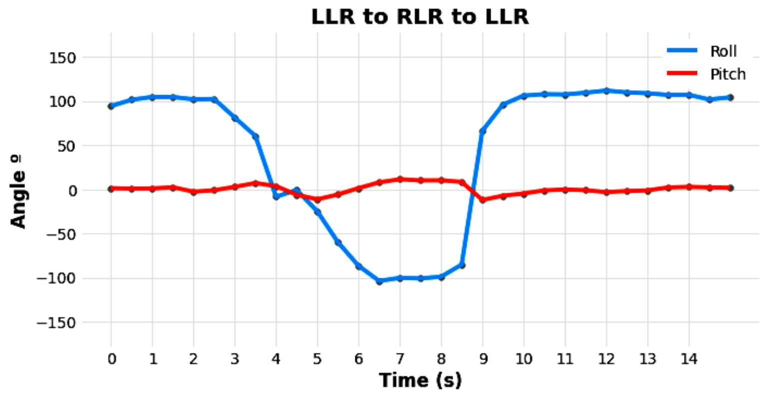


Figure 5.28: Transition from LLR to RLR and back to LLR

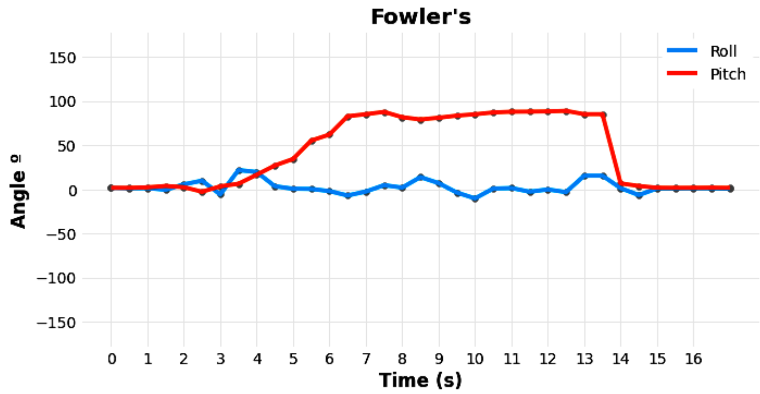


Figure 5.29: Transition from Supine to Fowler's

and only on those events data is received. This overview of the data enables the doctor or the patient to have an understanding of the patient's behavior throughout the night. With the information presented previously, we are also capable to decipher what sleeping position each event occurred.

Furthermore, we can "zoom" into a particular event to see the event with more resolution. Figure 5.31 shows two events separated by 20 seconds, in which we can observe the variation on the force sensors and the corresponding sleeping position. Between events, the patient has changed from supine to RLR.

Final Considerations

The results presented show that the system can track both the force exerted and the sleeping position of the patient. Although we can't quantify with certainty the amount of force exerted, mainly due to limitations of the sensors, we can still observe if a force is being applied or not.

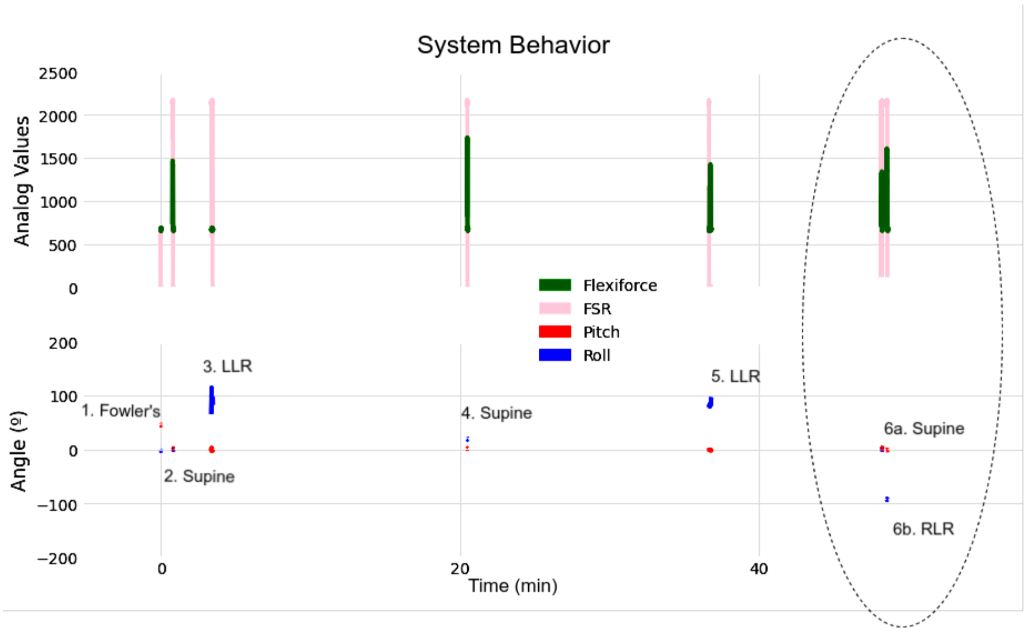


Figure 5.30: System behavior

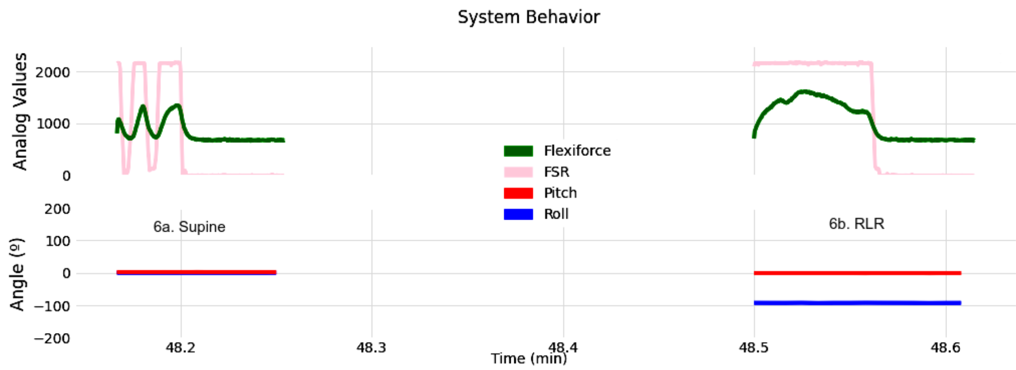


Figure 5.31: Close-up on events

We also can estimate, with acceptable accuracy, the sleeping position of the patients. The calibration of these positions was achieved through tests, simulating these various positions. The comparison between the estimation only using only the accelerometer and both the accelerometer and gyroscope was presented. We can say that the first is accurate enough, especially because of the application, meaning that we don't need to know the position of the patient at every point in time. To have a higher resolution, we implemented a complementary filter that merges the accelerometer and gyroscope readings, making the system more flexible and more accurate.

Chapter 6

Conclusion

In this thesis, the bruxism disorder was explored, which affects a significant percentage of the adult population, and with the tendency of increase in the younger population. Most of the time overlooked, we conclude that it can affect people both physically and mentally both in the short and long-term.

The development of this project arises from the need to find a system capable of monitoring physiological signals that enable us to study, understand, and detect events of bruxism. The construction of an intraoral system has advantages, in the sense that the device is close to the "source" of a bruxism event, but, it also has restrictions since the device has to be as less intrusive as possible.

One parameter to be measured is the force exerted between the teeth, leading to the introduction of a force sensor. The other parameter that we included in this system is the estimation of the patient's sleep position, which is achieved through an IMU. Another need of the system is to transmit the collected data to an external device. The protocol used for this purpose is the BLE, which is a protocol used for portable devices, with sufficient data throughput and low power consumption.

The force sensors tested for this system are the Flexiforce and the FSR 401. These are piezoresistive sensors, which means that the resistance will change as force is applied. These sensors will determine how the system will behave because, in a bruxism event, there is a force being exerted, so we can consider that a necessary condition. Furthermore, for the estimation of the patient's sleep position, we will use an IMU and, more specifically, the MPU-6050, which comprises an accelerometer, gyroscope, and temperature sensor in the same package.

The nRF52832 is an SoC developed by Nordic Semiconductors, which includes an ARM Cortex-M4, peripherals, and a built-in RF transceiver, allowing us to implement the BLE protocol without external ICs. The device is responsible for collecting data from all the sensors, implement the BLE stack, and transmit the collected data.

A personalized Service was developed to allow the transmission of the data. A Service is a concept introduced in BLE and, more specifically, on the Generic Attribute, to organize and access data. The service has three Characteristics, each one related to one piece of data, that is, the force sensors data, accelerometer, and gyroscope. These Characteristics also support notifications, which allows for maximizing the efficiency of the connection. Through the enabling/disabling of the Characteristics, the device will behave differently.

The estimation of the sleeping position is achieved through the roll and pitch angles. These values are determined by the accelerometer data. The gyroscope can be used to estimate the yaw angle, but, due to limitations on the sampling rate, its data is not entirely reliable.

The developed system presents satisfactory results and provides a solid basis for the development of a more robust solution and closer to a final product. In terms of future perspective, the system should integrate an EEPROM, to store data in case there is no active connection. Also, the force sensing may be improved, either by developing a custom sensor, or test different housings for these sensors. The measurement of other parameters such as sound or heart rate, may be included. Furthermore, inductive charging may be a better choice to charge the battery. Other factors like the occlusal splint, flexibility of the PCB, should be developed.

On a personal level, it was a project that made me grow since I had to learn a new architecture (ARM), a new communication protocol (BLE), and the development of a prototype from scratch.

Bibliography

[1] "Life expectancy - our world in data." https://ourworldindata.org/life-expectancy#all-charts-preview. (Accessed on 04/17/2020). [Quoted on p. iii, 8, 9]

[2] "Chapter 2 - principles of magnetic resonance imaging." https://users.fmrib.ox.ac.uk/~stuart/thesis/chapter_2/section2_6.html. (Accessed on 04/22/2020). [Quoted on p. iii, 10]

[3] "A family guide to ecmo | university of iowa hospitals & clinics." https://uihc.org/health-topics/family-guide-ecmo. (Accessed on 04/20/2020). [Quoted on p. iii, 11]

[4] "Wearable medical devices market size, share, growth | report 2026." https://www.fortunebusinessinsights.com/industry-reports/wearable-medical-devices-market-101070. (Accessed on 04/22/2020). [Quoted on p. iii, 12, 13]

[5] H. Cao, V. Leung, C. Chow, and H. Chan, "Enabling technologies for wireless body area networks: A survey and outlook," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84–93, 2009. [Quoted on p. iii, 14]

[6] "Wearable health devices—vital sign monitoring, systems and technologies." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6111409/. (Accessed on 04/21/2020). [Quoted on p. iii, 12, 17]

[7] T. Shochat, A. Gavish, E. Arons, N. Hadas, A. Molotsky, P. Lavie, and A. Oksenberg, "Validation of the bitestrip screener for sleep bruxism," *Oral surgery, oral medicine, oral pathology, oral radiology, and endodontics*, vol. 104, pp. e32–9, 10 2007. [Quoted on p. iii, 21, 22]

[8] "Fight stress & end teeth grinding disorder w/ smart wearable by beta hacktory — kickstarter."

https://www.kickstarter.com/projects/betahacktory/bruxrelief-fight-stress-and-bruxism-with-one-devic/description. (Accessed on 04/28/2020). [Quoted on p. iii, 25]

[9] J. H. Kim, P. McAuliffe, B. O'Connel, D. Diamond, and K. T. Lau, "Development of Bite Guard for Wireless Monitoring of Bruxism using Pressure-Sensitive Polymer," in *2010 International Conference on Body Sensor Networks*, pp. 109–116, 2010. [Quoted on p. iii, 26]

[10] J. Gao, L. Liu, P. Gao, Y. Zheng, W. Hou, and J. Wang, "Intelligent Occlusion Stabilization Splint with Stress-Sensor System for Bruxism Diagnosis and Treatment," *Sensors (Basel)*, vol. 20, Dec 2019. [Quoted on p. iii, 26, 27]

[11] "https://upload.wikimedia.org/wikipedia/commons/5/53/strain_gauge.svg." https://upload.wikimedia.org/wikipedia/commons/5/53/Strain_gauge.svg. (Accessed on 05/25/2020). [Quoted on p. iii, 29]

[12] K. Wong, *Electrical Engineering - Volume I*. EOLSS Publications, 2009. [Quoted on p. iii, 31]

[13] "instrumentation-amplifier-circuits.jpg (480×327)." https://www.allaboutcircuits.com/uploads/articles/instrumentation-amplifier-circuits.jpg. (Accessed on 05/25/2020). [Quoted on p. iii, 33]

[14] "Decision tree algorithm — explained - towards data science." https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4. (Accessed on 05/21/2020). [Quoted on p. iii, 38]

[15] "Logreg_1.png (571×306)." http://juangabrielgomila.com/wp-content/uploads/2015/04/LogReg_1.png. (Accessed on 05/25/2020). [Quoted on p. iii, 39]

[16] P. Valente Klaine, M. Imran, O. Onireti, and R. Souza, "A survey of machine learning techniques applied to self organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. PP, pp. 1–1, 07 2017. [Quoted on p. iii, 40]

[17] "Interlink electronics: Buy multi-platform hybrid sensors." https://www.interlinkelectronics.com/. (Accessed on 07/06/2020). [Quoted on p. iv, 47]

[18] "Gleie.png (523×442)." https://i.stack.imgur.com/GleIe.png. (Accessed on 09/15/2020). [Quoted on p. iv, 48]

[19] "Force sensors (flexiforce) | tekscan." https://www.tekscan.com/store/category/force-sensors-flexiforce. (Accessed on 11/27/2020). [Quoted on p. iv, 47, 48, 49]

[20] "Embedded force sensors | tekscan." https://www.tekscan.com/products-solutions/embedded-force-sensors. (Accessed on 06/30/2020). [Quoted on p. iv, 49]

[21] "Accelerometer - code: Robotics." https://docs.idew.org/code-robotics/references/physical-inputs/accelerometer. (Accessed on 09/15/2020). [Quoted on p. iv, 50]

[22] "nrf52832 - versatile bluetooth 5.2 soc - nordicsemi.com." https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52832. (Accessed on 07/06/2020). [Quoted on p. iv, 52]

[23] "Architecture of bluetooth low energy - iot projects with bluetooth low energy." https://subscription.packtpub.com/book/hardware_and_creative/9781788399449/1/01lvl1sec11/architecture-of-bluetooth-low-energy. (Accessed on 07/06/2020). [Quoted on p. iv, 54]

[24] "Bm71 ble connection states - developer help." https://microchipdeveloper.com/ble:bm71-ble-connection-states. (Accessed on 07/06/2020). [Quoted on p. iv, 55]

[25] InvenSense Inc., *MPU-6000 and MPU-6050 Register Map and Descriptions*, 8 2013. Rev. 4.2. [Quoted on p. iv, 58, 59, 60]

[26] Interlink Electronics, *FSR® Integration Guide Evaluation Parts Catalog*. Rev. D. [Quoted on p. iv, 77]

[27] Silicon Labs, *Bluetooth® LE Fundamentals*. Rev. 0.6. [Quoted on p. iv, 78, 79, 80]

[28] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 1–12, 2010. [Quoted on p. vii, 15]

[29] T. Verma, K. Kumathalli, V. Jain, and R. Kumar, "Bite force recording devices - a review," *Journal of Clinical and Diagnostic Research*, vol. 11, pp. ZE01–ZE05, 09 2017. [Quoted on p. vii, 28, 29, 30, 31]

[30] "Force sensors (flexiforce) | tekscan." https://www.tekscan.com/store/category/force-sensors-flexiforce. (Accessed on 06/30/2020). [Quoted on p. vii, 49]

[31] D. Manfredini, E. Winocur, L. Guarda-Nardini, D. Paesani, and F. Lobbezoo, "Epidemiology of bruxism in adults: a systematic review of the literature," *J Orofac Pain*, vol. 27, no. 2, pp. 99–110, 2013. [Quoted on p. 1]

[32] "Bruxism: A literature review." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3081266/`. (Accessed on 04/24/2020). [Quoted on p. 5]

[33] "Bruxism: A literature review." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3081266/`. (Accessed on 04/24/2020). [Quoted on p. 6]

[34] "Prevalence of sleep bruxism in ibd patients and its correlation to other dental disorders and quality of life." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5867589/`. (Accessed on 04/24/2020). [Quoted on p. 6]

[35] N. HUYNH, T. KATO, P. H. ROMPRÉ, K. OKURA, M. SABER, P. A. LANFRANCHI, J. Y. MONTPLAISIR, and G. J. LAVIGNE, "Sleep bruxism is associated to micro-arousals and an increase in cardiac sympathetic activity," *Journal of Sleep Research*, vol. 15, no. 3, pp. 339–346, 2006. [Quoted on p. 6]

[36] "Sleep bruxism: Current knowledge and contemporary management." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5026093/`. (Accessed on 04/24/2020). [Quoted on p. 6]

[37] "The bruxoff device as a screening method for sleep bruxism in dental practice." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6678144/`. (Accessed on 04/24/2020). [Quoted on p. 6]

[38] T. Kato, C. Dal-Fabbro, and G. Lavigne, "Current knowledge on awake and sleep bruxism: Overview," *The Alpha omegan*, vol. 96, pp. 24–32, 08 2003. [Quoted on p. 7]

[39] "Sleep bruxism: Current knowledge and contemporary management." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5026093/`. (Accessed on 04/27/2020). [Quoted on p. 7]

[40] "International consensus on the assessment of bruxism: Report of a work in progress." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6287494/`. (Accessed on 04/27/2020). [Quoted on p. 7]

[41] "Who | definitions." `https://www.who.int/medical_devices/definitions/en/`. (Accessed on 04/20/2020). [Quoted on p. 8]

[42] G. I. Barbash and S. Glied, "New technology and health care costs–the case of robot-assisted surgery.," *The New England journal of medicine*, vol. 363 8, pp. 701–4, 2010. [Quoted on p. 9]

[43] "Types of minimally invasive surgery (robotic, endoscopic, laparoscopic): Johns hopkins medicine in baltimore, md." https://www.hopkinsmedicine.org/minimally_invasive_robotic_surgery/types.html. (Accessed on 04/20/2020). [Quoted on p. 9]

[44] "What is laser surgery? | stanford health care." https://stanfordhealthcare.org/medical-treatments/l/laser/types/laser-surgery.html. (Accessed on 04/20/2020). [Quoted on p. 9]

[45] "Who | medical imaging." https://www.who.int/diagnostic_imaging/en/. (Accessed on 04/20/2020). [Quoted on p. 10]

[46] "Differences betweeen x-rays, ct scans & mri's | envision radiology." https://www.envrad.com/difference-between-x-ray-ct-scan-and-mri/. (Accessed on 04/20/2020). [Quoted on p. 10]

[47] "Magnetic resonance imaging (mri)." https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri. (Accessed on 04/20/2020). [Quoted on p. 10]

[48] "Extra corporeal oxygenation (ecmo) learning package." https://www.aci.health.nsw.gov.au/__data/assets/pdf_file/0007/306583/ECMO_Learning_package.pdf. (Accessed on 04/20/2020). [Quoted on p. 10]

[49] "What is an electronic health record (ehr)? | healthit.gov." https://www.healthit.gov/faq/what-electronic-health-record-ehr. (Accessed on 04/21/2020). [Quoted on p. 11]

[50] "Using machine learning to estimate risk of cardiovascular death | mit news." http://news.mit.edu/2019/using-machine-learning-estimate-risk-cardiovascular-death-0912. (Accessed on 04/21/2020). [Quoted on p. 11]

[51] "Machine learning in medicine." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5831252/. (Accessed on 04/21/2020). [Quoted on p. 11]

[52] C. D. Fjell, H. Jenssen, K. Hilpert, W. A. Cheung, N. Panté, R. E. W. Hancock, and A. Cherkasov, "Identification of novel antibacterial peptides by chemoinformatics and machine learning," *Journal of Medicinal Chemistry*, vol. 52, no. 7, pp. 2006–2015, 2009. PMID: 19296598. [Quoted on p. 11]

[53] "The surgeon who operates from 400km away - bbc future." https://www.bbc.com/future/article/

20140516-i-operate-on-people-400km-away. (Accessed on 04/21/2020). [Quoted on p. 11]

[54] M. Chan, D. Estève, J.-Y. Fourniols, C. Escriba, and E. Campo, "Smart wearable systems: Current status and future challenges," *Artificial Intelligence in Medicine*, vol. 56, no. 3, pp. 137 – 156, 2012. [Quoted on p. 12]

[55] "Wearable technology applications in healthcare: A literature review." https://www.himss.org/resources/wearable-technology-applications-healthcare-literature-review. (Accessed on 04/21/2020). [Quoted on p. 12]

[56] M. N. Islam and M. R. Yuce, "Review of medical implant communication system (mics) band and network," *ICT Express*, vol. 2, no. 4, pp. 188 – 194, 2016. Special Issue on Emerging Technologies for Medical Diagnostics. [Quoted on p. 15]

[57] "272-284 jcse-2013-0068.fm." http://jcse.kiise.org/files/V7N4-07.pdf. (Accessed on 04/23/2020). [Quoted on p. 17]

[58] "Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3892855/. (Accessed on 04/23/2020). [Quoted on p. 17]

[59] "Pacemaker mais pequeno do mundo – sns." https://www.sns.gov.pt/noticias/2016/07/25/pacemaker-mais-pequeno-do-mundo/. (Accessed on 04/21/2020). [Quoted on p. 19]

[60] "Pacemakers | national heart, lung, and blood institute (nhlbi)." https://www.nhlbi.nih.gov/health-topics/pacemakers. (Accessed on 04/21/2020). [Quoted on p. 19]

[61] "8 october 1958, d day for the implantable pacemaker." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2572009/. (Accessed on 04/21/2020). [Quoted on p. 19]

[62] "Insulin delivery devices: Syringes, pens, pumps, and more." https://www.webmd.com/diabetes/features/insulin-delivery-system#2. (Accessed on 04/21/2020). [Quoted on p. 19]

[63] "Insulin pump types." https://www.diabetes.co.uk/insulin-pumps/insulin-pump-types.html. (Accessed on 04/21/2020). [Quoted on p. 19]

[64] "Fda approves new closed-loop insulin delivery system for people with type 1 diabetes - the verge." https://www.theverge.com/2019/12/13/21020811/

fda-closed-loop-insulin-system-software-diabetes-tandem-control-iq. (Accessed on 04/21/2020). [Quoted on p. 19]

[65] "Patch pumps for insulin." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6313281/. (Accessed on 04/21/2020). [Quoted on p. 19]

[66] "The implantable insulin pump | what is the implantable insulin pump?." https://theiipump.com/index.php/what-is-the-implantable-insulin-pump/. (Accessed on 04/21/2020). [Quoted on p. 19]

[67] "What is bac? | office of alcohol policy and education." https://alcohol.stanford.edu/alcohol-drug-info/buzz-buzz/what-bac. (Accessed on 04/22/2020). [Quoted on p. 19]

[68] "Three types of bac testing." https://www.bactrack.com/blogs/expert-center/35043461-three-types-of-bac-testing. (Accessed on 04/22/2020). [Quoted on p. 19]

[69] "This wearable will tell you when you're drunk." https://www.forbes.com/sites/johngreathouse/2017/07/08/this-wearable-will-tell-you-when-youre-drunk/#770561403beb. (Accessed on 04/22/2020). [Quoted on p. 20]

[70] "Mental health - our world in data." https://ourworldindata.org/mental-health. (Accessed on 04/22/2020). [Quoted on p. 20]

[71] "Otsuka and proteus® announce the first u.s. fda approval of a digital medicine system: Abilify mycite® (aripiprazole tablets with sensor) | discover otsuka | otsuka in the u.s.." https://www.otsuka-us.com/discover/articles-1075. (Accessed on 04/22/2020). [Quoted on p. 20]

[72] "Fda approves pill with sensor that digitally tracks if patients have ingested their medication | fda." https://www.fda.gov/news-events/press-announcements/fda-approves-pill-sensor-digitally-tracks-if-patients-have-ingested-their-medica (Accessed on 04/22/2020). [Quoted on p. 20]

[73] "Touchpoints for calm - thetouchpoint solution™." https://thetouchpointsolution.com/collections/kids-sleep-focus/products/touchpoints-for-calm. (Accessed on 04/22/2020). [Quoted on p. 20]

[74] "Ageing | united nations." https://www.un.org/en/sections/issues-depth/ageing/. (Accessed on 04/22/2020). [Quoted on p. 20]

[75] "A survey on recent advances in wearable fall detection systems." https://www.hindawi.com/journals/bmri/2020/2167160/. (Accessed on 04/22/2020). [Quoted on p. 20]

[76] "Hypertension." https://www.who.int/news-room/fact-sheets/detail/hypertension. (Accessed on 04/22/2020). [Quoted on p. 20]

[77] "Detecting vital signs with wearable wireless sensors." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231103/. (Accessed on 04/22/2020). [Quoted on p. 21]

[78] "Distar - an international company - bitestrip." http://www.distar.com/products_bitefresh.html. (Accessed on 04/27/2020). [Quoted on p. 22]

[79] "Bruxoff." http://www.bruxoff.com/en/bruxoff/. (Accessed on 04/27/2020). [Quoted on p. 23]

[80] K. Saczuk, B. Łapińska, P. Wilmont, Pawlak, and M. Łukomska Szymańska, "The bruxoff device as a screening method for sleep bruxism in dental practice," *Journal of Clinical Medicine*, vol. 8, p. 930, 06 2019. [Quoted on p. 23]

[81] T. Castroflorio, A. Deregibus, A. Bargellini, C. Debernardi, and D. Manfredini, "Detection of sleep bruxism: comparison between an electromyographic and electrocardiographic portable holter and polysomnography," *Journal of Oral Rehabilitation*, vol. 41, no. 3, pp. 163–169, 2014. [Quoted on p. 23]

[82] "User manual - tecno-gaz | manualzz." https://manualzz.com/doc/7059922/user-manual---tecno-gaz. (Accessed on 04/27/2020). [Quoted on p. 23, 24]

[83] R. Needham and S. Davies, "Summary of: Use of the grindcare ® device in the management of nocturnal bruxism: A pilot study," *British dental journal*, vol. 215, p. E1, 07 2013. [Quoted on p. 23, 24]

[84] "Nox t3 portable sleep monitor | resmed." https://www.resmed.com/epn/en/healthcare-professional/products/diagnostics/nox-t3.html. (Accessed on 04/27/2020). [Quoted on p. 24, 25]

[85] N. Medical, *Nox T3*. Nox Medical, IS - 105 Reykjavik, 1 ed. [Quoted on p. 24, 25]

[86] K. S. I.M. Abdel-Motaleb, K. Ravanasa, "Design of a pressure sensor to monitor teeth grinding," 2012. [Quoted on p. 30]

[87] P. Eswaran and M. Subramani, "Mems capacitive pressure sensors: A review on recent development and prospective," *International Journal of Engineering and Technology*, vol. 5, pp. 2734–2746, 06 2013. [Quoted on p. 30]

[88] "Wheatstone_englisch." http://eln.teilam.gr/sites/default/files/Wheatstone%20bridge.pdf. (Accessed on 05/25/2020). [Quoted on p. 31, 32]

[89] "Operational amplifier summary, op-amp basics." https://www.electronics-tutorials.ws/opamp/opamp_8.html. (Accessed on 05/25/2020). [Quoted on p. 32]

[90] "Microsoft word - edch 8 filter.doc." https://www.analog.com/media/en/training-seminars/design-handbooks/Basic-Linear-Design/Chapter8.pdf. (Accessed on 05/25/2020). [Quoted on p. 33, 34]

[91] "1920px-bandform_template.svg.png (1920×812)." https://upload.wikimedia.org/wikipedia/en/thumb/e/ec/Bandform_template.svg/1920px-Bandform_template.svg.png. (Accessed on 05/25/2020). [Quoted on p. 33]

[92] "Fast fourier transform - towards data science." https://towardsdatascience.com/fast-fourier-transform-937926e591cb. (Accessed on 05/25/2020). [Quoted on p. 34]

[93] "Finite impulse response - an overview | sciencedirect topics." https://www.sciencedirect.com/topics/engineering/finite-impulse-response. (Accessed on 05/25/2020). [Quoted on p. 34]

[94] "Infinite impulse response - an overview | sciencedirect topics." https://www.sciencedirect.com/topics/computer-science/infinite-impulse-response. (Accessed on 05/25/2020). [Quoted on p. 35]

[95] "nrf52 series socs - nordic | mouser." https://eu.mouser.com/new/nordic-semiconductor/nrf52-series-soc/. (Accessed on 05/25/2020). [Quoted on p. 35]

[96] "nrf52832-product-brief.pdf." https://www.nordicsemi.com/-/media/Software-and-other-downloads/Product-Briefs/nRF52832-product-brief.pdf?la=en&hash=2F9D995F754BA2F2EA944A2C4351E682AB7CB0B9. (Accessed on 05/25/2020). [Quoted on p. 36]

[97] "nrf24 series - nordic semiconductor - nordicsemi.com." https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF24-series. (Accessed on 05/25/2020). [Quoted on p. 36]

[98] "Productspec duchess.fm." https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf. (Accessed on 05/25/2020). [Quoted on p. 36]

[99] "esp32_technical_reference_manual_en.pdf." https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. (Accessed on 05/25/2020). [Quoted on p. 36, 37]

[100] "Logistic regression — detailed overview - towards data science." https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc. (Accessed on 05/25/2020). [Quoted on p. 39]

[101] "Machine learning basics with the k-nearest neighbors algorithm." https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. (Accessed on 05/25/2020). [Quoted on p. 39, 40]

[102] H. Omar, M. Alhajrasi, N. Felemban, and A. Hassan, "Dental arch dimensions, form and tooth size ratio among a saudi sample," *Saudi medical journal*, vol. 39, pp. 86–91, Jan 2018. 29332114[pmid]. [Quoted on p. 44]

[103] "nrf52832 product specification." https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf. (Accessed on 07/06/2020). [Quoted on p. 51, 52]

[104] "1901_feature_overview_brief_final.pdf." https://www.bluetooth.com/wp-content/uploads/2019/03/1901_Feature_Overview_Brief_FINAL.pdf. (Accessed on 07/06/2020). [Quoted on p. 53, 54, 55]

[105] D. Guerra, "Relatório - ble.pdf." http://web.tecnico.ulisboa.pt/~ist168247/rs/download/Relat%C3%B3rio%20-%20BLE.pdf, 2015. (Accessed on 07/06/2020). [Quoted on p. 53, 54, 55]

[106] "Nordic semiconductor infocenter." https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct_sdk%2Fstruct%2Fsdk_nrf5_latest.html&cp=7_1. (Accessed on 07/06/2020). [Quoted on p. 56]

[107] "Nordic semiconductor infocenter." https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_gsg_ses%2FUG%2Fgsg%2Fsoftdevices.html. (Accessed on 07/06/2020). [Quoted on p. 56]

[108] InvenSense Inc., *MPU-6000 and MPU-6050 Product Specification*, 8 2013. Rev. 3.4. [Quoted on p. 57]

[109] M. Pedley, "Tilt sensing using a three-axis accelerometer," Mar 2013. [Quoted on p. 62]

[110] "Guide to understanding successive approximation registers (sar) and flash adcs | maxim integrated." `https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1080.html`. (Accessed on 10/26/2020). [Quoted on p. 71]

[111] "Nordic semiconductor infocenter." `https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52832.ps.v1.1%2Fsaadc.html`. (Accessed on 10/26/2020). [Quoted on p. 72, 73, 74]

[112] Tekscan, *Best Practices in Electrical Integration of the FlexiForce™ Sensor*. Rev. B. [Quoted on p. 74, 75]

[113] "The ultimate bluetooth low energy (ble) guide - novel bits." `https://www.novelbits.io/basics-bluetooth-low-energy/`. (Accessed on 10/26/2020). [Quoted on p. 78, 79, 80]

[114] Bluetooth, *Bluetooth Core Specification*, 12 2019. Rev. 5.2. [Quoted on p. 78, 79, 80]