1 2 9 0

UNIVERSIDADE D
COIMBRA

Bernardo Henriques Plácido Fernandes

SMART BED IOT-BASED WIRELESS DATA ACQUISITION FOR UNTETHERED PATIENTS

Dissertation submitted to the Faculty of Sciences and Technology of the University of Coimbra for the Master's Degree in Biomedical Engineering with the specialization in Clinical Informatics and Bioinformatics, supervised by Dr. David Portugal and Prof. Dr. Mahmoud Tavakoli.

February 2022

• U C •

Smart Bed IoT-based Wireless Data Acquisition for Untethered Patients

Supervisor:

Doutor David Portugal

Co-Supervisor:

Professor Doutor Mahmoud Tavakoli

Jury:

Professor Doutor Jorge Manuel Oliveira Henriques

Doutor David Bina Siassipour Portugal

Professor Doutor Jorge Miguel Sá Silva

Dissertation submitted in partial fulfillment for the degree of Master of Science in Biomedical Engineering.

Coimbra, February 2022

Acknowledgements

Having reached the end of this great challenge that was the completion of this dissertation and Biomedical Engineering Degree, I would like to thank multiple people for their essential contribution. Due to these people, I never felt alone in this journey.

Firstly, I deeply thank Dr. David Portugal for his crucial guidance and for granting me the opportunity to work in a project with such potential and innovation that allowed me to grow academically and personally. His great feedback skills were essential to add value to the work and dissertation and his commitment, high standards and knowledge are pivotal for the success of the WoW project.

To Prof. Dr. Mahmoud Tavakoli, I thank his feedback ideas for the design of the GUI.

For all the help that the WoW team members Miguel, Francisca and specially Fernanda and José provided, I am very grateful. I take this opportunity to wish success to their careers and to the careers of the recently added members Daniel and Marco.

To Centro Hospitalar e Universitário de Coimbra, I thank the opportunity to perform tests in a real scenario and to Cármen, Inês and Patrícia I thank them their sympathy during the pilot. I thank the project leader Glintt Healthcare solutions, S.A. for their contribution to the project.

I thank Dr. Gonçalo Martins for his feedback to this dissertation and his important and candid point of view.

I thank my cousin Diogo for his availability and help in the starting stages of the work and I thank Maria for her help and support in the latter stages.

To my dearest friends Daniel Filipe, Gonçalo Filipe, João Miguel, João Pedro, Luis António and Pedro Miguel, I thank them for all their friendship, support and mutual help in the completion of my degree and for all the study sessions, music jams and great memories.

To my Erasmus friends, specially to Daniel, Joana and Patrícia, I thank the amazing trips and memories in this great adventure.

I thank my friends from BEST Coimbra and IEEE UC Student Branch for the opportunities to learn soft skills that I will certainly use all my life.

I thank all my teachers and educational establishments for the importance and value of the knowledge transmitted to me and for inspiring me to learn since I was a child.

To my childhood friends I thank their support, friendship, incentive and enthusiasm in this important moment of my life.

At last, I thank all my family for their unconditional love, specially to my mother, father and sister for supporting my decisions and encouraging me in the pursuit of my goals. I am certain they will be by my side for the next milestones.

Resumo

Internet das Coisas (IoT, sigla proveniente do termo inglês Internet of Things), que consiste no desenvolvimento de redes de dispositivos conectados que comunicam dados através de Internet, é um campo de investigação relativamente recente, com um grande potencial na área da saúde. O uso de IoT permite aos pacientes controlarem mais efetivamente a sua própria saúde a qualquer momento e, no caso de idosos, permite-lhes ser independentes em casa por mais tempo.

Tendo em conta ambientes hospitalares, o uso de comunicações sem fios permite a monitorização sem confinar o paciente à cama. Sem o uso de fios ligados ao paciente, a desconexão indesejada dos instrumentos de medição devido aos seus movimentos é anulada, diminuindo o número de falsos alarmes que necessitam de atenção imediata e assim aliviando os recursos humanos e financeiros do sistema de saúde. Para além disso, IoT permite antecipar a alta dos pacientes sem comprometer a sua saúde, já que permite monitorizá-los remotamente fora do ambiente hospitalar.

O principal objetivo desta dissertação é propor um sistema sem fios de acquisição de dados para ser usado num nó de IoT que adquire dados de sensores colocados no corpo do paciente. O sistema é responsável por adquirir vários tipos de dados de sinais vitais e seguinte preprocessamento. Os dados são depois armazenados numa base de dados e transmitidos para outros nós da arquitetura. Para visualizar os sinais vitais, uma interface gráfica foi desenvolvida.

A solução apresentada foi validada em experiências conduzidas num cenário real (hospital) usando dados de voluntários obtidos em tempo real.

Palavras-Chave: Internet das Coisas; Sistema de Assistência Médica; Aquisição de Dados; Transmissão de Dados; Visualização de Dados

Abstract

Internet of Things (IoT), which consists in the development of networks of devices interacting with each other via Internet, is a relatively new field of research with great potential in healthcare. The use of IoT offers patients the opportunity to control more effectively their own health at all times, with the added effect of enabling elderly people to live independently at home for longer.

Regarding hospital environments, the use of IoT wireless communications allows monitorization without confining the patient. Without the use of wires attached to the patient, the undesired detachment of the measuring electrodes due to their movement is nullified, decreasing the number of false alarms that require immediate attention and relieving human and financial resources of the healthcare system. Furthermore, IoT allows for earlier discharge without compromising the patients' health outcomes since it allows the patients' remote monitorization outside the hospital environment.

The main goal of this dissertation is to propose a wireless data acquisition system to be used in an IoT node that acquires data from sensors placed in the patient's body. The system is responsible for acquiring various types of vital signs data, followed by their preprocessing. The data is then stored in a database and transmitted to other nodes of the architecture. In order to visualize the vital signs of the patient, a graphical user interface was developed.

This solution was validated through experiments conducted in a real scenario (hospital) using real-time data from volunteers.

Keywords: Internet of Things; Healthcare System; Data Acquisition; Data Transmission; Data Visualization

Contents

List of Acronyms

AAL	Ambient Assisted Living
ADR	Adverse Drug Reaction
ADSL	Asymmetric Digital Subscriber Line
AI	Artificial Inteligence
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
BLE	Bluetooth Low Energy
BLEAK	Bluetooth Low Energy platform Agnostic Klient
BPM	Blood Pressure Monitors
CHUC	Centro Hospitalar e Universitário de Coimbra
CSS	Cascading Style Sheets
DAS	Data Acquisition System
DPS	Degrees per Second
ECG	Eletrocardiogram
FHIR	Fast Healthcare Interoperability Resources
FFT	Fast Fourier Transform
FPS	Frames Per Second
GUI	Graphical User Interface
HTML	Hypertext Markup Language

HTTP	Hypertext Transfer Protocol
IMU	Inertial Measurement Unit
IoT	Internet of Things
IT	Information technology
IP	Internet Protocol
JSON	JavaScript Object Notation
LPWAN	Low-Power Wide-Area Network
M2M	Machine to Machine
MiTM	Man-in-the-Middle
MQTT	Message Queuing Telemetry Transport
OS	Operating System
PC	Personal Computer
PHP	Hypertext Preprocessor
PPG	Photoplethysmography
PSD	Power Spectral Density
RFID	Radio Frequency Identification
SIG	Special Interest Group
SQL	Structured Query Language
URL	Uniform Resource Locators
UTP	Unshielded Twisted Pairs
WBAN	Wireless Body Area Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

WoW **W**ireless bi**O**monitoring stickers and smart bed architecture: to**W**ards Untethered Patients

XML Extensible Markup Language

List of Figures

List of Tables

1 Introduction

1.1 Context and Motivation

Hospitalized patients are often tethered to many measurement instruments in order to achieve continuous biomonitoring. The confinement of the patient to the bed limits his mobility. Furthermore, patients' movements often result in detachment of the measuring electrodes. These situations lead to false alarms, which require immediate attention, straining human and financial resources of the healthcare system.

Internet of things (IoT) is described as network of devices interacting with each other via machine to machine (M2M) communications, enabling collection and exchange of data [6]. According to previous works [36, 61, 86, 3], the emerging technology breakthrough of the IoT is expected to offer promising solutions in healthcare with the potential to minimize the problems described above.

From the point of view of the patient, the use of IoT in Healthcare allows for a better control over their own health at all times. Since the patient can be monitored at home, it allows an early discharge without compromising their health outcomes and possible costs of traveling to meet healthcare professionals can be reduced. Regarding elderly people, IoT enables them to live independently at home for longer [6].

In this dissertation, a Data Acquisition System (DAS) is proposed to acquire data from sensors connected to the patients. Data acquisition is the process of sampling signals that measure real-world physical phenomena and converting them into a digital form that can be manipulated by a computer and software [69].

When allied with data collection and reporting, data acquisition allows the monitorization of non-critical patients in home as well as in the hospital, improving access to healthcare resources (specially in rural areas) and reducing strain on healthcare systems [6].

1.2 Objectives

The main goal of this dissertation is to propose a system to be used in an IoT node named Smart Box to wirelessly acquire vital signals data from sensors. The system is also responsible for temporary data storage in a database and data transmission of the biomonitoring parameters to a Gateway that interacts with a central patient management system.

Another goal is to visualize these signals through a Graphical User Interface (GUI) in a web interface.

The final goal is to test, adjust, and validate the developed system, in order to be used in experimental tests in a real scenario at Centro Hospitalar e Universitário de Coimbra (CHUC), as illustrated in Figure 1.1.

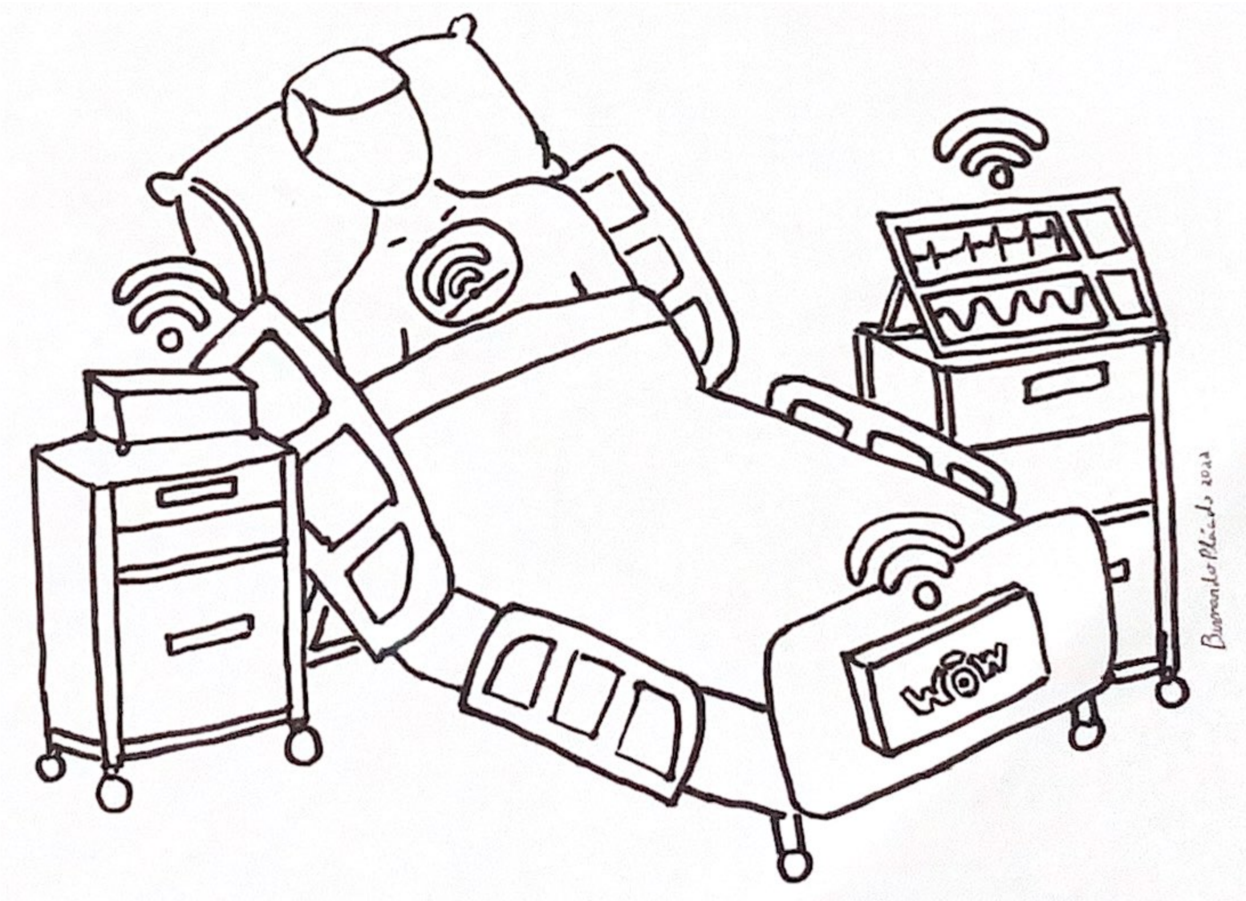


Figure 1.1: Illustration of a hospitalized, patient, Biosticker, Gateway and Smart Box and GUI.

1.3 Overview

This dissertation is organized in multiple chapters. Chapter 1 consists of an introduction to the dissertation, stating its context, motivation and objectives. In chapter 2, the state of art with the most relevant research work related to data acquisition, communications, healthcare interfaces and healthcare system architectures is explored. In this chapter, a comparative study about the works mentioned is present and the contributions of the dissertation are detailed. Chapter 3 provides background information about the WoW project, aiding in the understanding of the work developed. The system components, the communications used and the methodology of the work is presented in this chapter. Chapter 4 presents detailed information about the system modules implementation. In chapter 5, the results relevant to the system adequacy and reliability are presented and discussed. Chapter 6 is the conclusion of the dissertation, summing up the work and stating the future work to be implemented.

2 Literature Review

Digital health technologies have the goal of improving the health of populations or individuals, using computing platforms, connectivity, software, and sensors. These technologies can be a medical product, aid in diagnostics or improve the awareness of the patient's health [22].

IoT is an emerging global Internet-based information architecture facilitating the exchange of goods and services in global supply chain networks in a reliable manner [79].

It has been applied to many fields, such as smart parking [1], agriculture [31] and water usage management [67] and it is expected to be used on developing intelligent systems in multiple areas, such as: traffic congestion minimization [17], crash-avoiding cars [2], smart grids [73], smart homes [64], visually impaired people mobility and accessibility [66], natural calamities prediction [71], waste management [40] and security [36].

The use of IoT in the existing systems mentioned above has proved that its techniques are capable of monitoring remote objects, using data collection and reporting [6], considering that, in order to interconnect the multiple used devices, their transmission range are overlapping [11].

Since IoT is still a relatively new field of research, its use in healthcare has great potential [6]. This potential has attracted much attention that resulted in multiple studies [86, 36, 77, 72] that identify and confirm IoT as an solution to alleviate the pressures on healthcare systems [6].

Even thought IoT has many advantages when used in healthcare, it is also associated with some disadvantages. The sensors that provide the vital signs may need to be regularly calibrated if they are monitoring inaccurately and there is a risk of disconnection from the healthcare services if the user is out of the cellular range or if the device runs out of battery. The increase of IoT products and devices can lead to interoperability issues if the manufacturers do not follow standard rules [36, 72]. IoT devices have data storage limitations, resulting in short data lifespan. This can lead to losses of evidences caused by data being overwritten. A solution to this problem is transferring the data to another device such as a Gateway or cloud [13].

When applied to healthcare, the data exchanged in IoT can be obtained through sensors. So sensors are fundamental to most systems and applications, and without the unique input from them, no system would be useful or even functional [28]. Wearable sensor nodes are those that measure physiological conditions, such as the vital signs pulse, respiratory rate, and body temperature [6].

Data acquisition converts values from the sensors into a digital form [69] that is then communicated in the system architecture using many communication standards.

2.1 Data Acquisition

Data acquisition is the process of sampling signals that measure real-world physical phenomena and converting them into a digital form that can be manipulated by a computer and software [69]. An accurate Data Acquisition System (DAS) is essential to any IoT system, since reliable data is crucial to achieve the said system goal. In this subsection, various types of DAS's are explored.

In [21], a system with the aim of providing temperature and humidity measurements was presented. In this work, the data was acquired using the multiple GPIO ports of the Raspberry Pi connected to acquisition software application using RJ-45 connectors and Unshielded Twisted Pairs (UTP) cables.

Another system in which the sensor values are sent to a Raspberry Pi through wires is the one presented in [62]. The data from the sensors is accessed using the arduPi software library provided by the manufacturer that is responsible for simultaneously storing the received sensor data and sending it to standard output in a certain format.

The authors of [5] propose another method to acquire temperature and humidity measurements. In this system, the Message Queuing Telemetry Transport (MQTT) protocol was used to achieve real time data acquisition. In this system, a device named Wemos D1 Mini ESP 8266 Wi-Fi Module publishes data from DHT 11 sensor to a computer that functions as the MQTT broker. The MQTT broker facilitates exchange of data between the publisher and subscriber.

Different methods of acquisition can be used simultaneously in a system, such as in [70]. In this system, the architecture was developed in order to integrate sensors that used different networks protocols such as Wi-Fi, ZigBee and Bluetooth.

In [84], it is proposed an acquisition method that uses Bluetooth Low Energy (BLE) to collect Blood Pressure data from a Blood Pressure Monitors (BPM). In this proposed system, Generic ATTribute Profile (GATT) is used to determine how two devices interact. According to the authors, the first step is to understand the services and characteristics of the Koogeek sensor used, using tools that are pre-installed on a Ubuntu system, such as `hcitool`, `bluetoothctl` and `gatttool`. Using `hcitool`, the authors discovered the services and which of their characteristics had the property `Notify` or `Indicate`. The data is acquired when a characteristic with

Table 2.1: Comparative table of Data Acquisition Systems.

Works	Type of data acquired	Communication standard	Device	Data storage	Highlights
Ferencz and Domokos [21]	temperature, humidity	Serial Peripheral Interface	Raspberry Pi	multi-node Apache Cassandra cluster	Non-wireless acquisition
Pap et al. [62]	blood pressure, pulse oximeter, air flow, galvanic skin response, body temperature	not stated	Raspberry Pi	Node.js Web Server	Non-wireless acquisition, Independent Sofware Components for Sensor Access
Atmoko et al. [5]	temperature, humidity	Wi-Fi (MQTT)	Wemos D1 Mini ESP Wi-Fi Module	MySQL	This study proves that MQTT is better for a DAS than HTTP
Spanò et al. [70]	ECG	Wi-Fi ZigBee Bluetooth	Gateway connected to ADSL router	not stated	Architecture enables the usage of different networks protocols
Wu and Martin [84]	Blood Pressure	BLE (GATT)	Device with Ubuntu	not stated	Uses BLE, he most well suited communication standard

the property `Notify` changes its ASCII value. When this ASCII value is translated the data is considered acquired. The last step of the acquisition was to automate it.

In Table 2.1, a comparative study of data acquisition systems mentioned above is presented.

The Data Acquisition systems from [21, 62], even though well tested with satisfactory results, are not an appropriate alternative to be developed in this dissertation, since they rely on wires to function that would restrict the movements of the patient. Nevertheless, both systems showed that the Raspberry Pi is an adequate choice to be used as the device responsible for the data acquisition.

The DAS in [70] has the advantage of being able to incorporate sensor signals using different communication standards.

The method presented in [84] is similar to the one proposed in this dissertation. In accordance with the Communications Section 2.2 of this dissertation, BLE is extremely well suited to healthcare applications due to its appropriate range, low latency, low power consumption and robustness to interference. When allied with GATT, that works as an Application Programming Interface (API), it allows for a safer connection and a more organized acquisition.

2.2 Communications

Healthcare systems rely on two types of communication: Short Range Communications and Long Range Communications.

Short Range Communications are responsible for the communication between IoT nodes placed within the WLAN (Wireless Local Area Network) and the WBAN (Wireless Body Area Network). The standards explored in this section are Bluetooth, Bluetooth Low Energy, Zigbee, Wi-Fi and

RFID.

Long Range Communications connect the central node of the WLAN/WBAN to a base station. The standards explored in this section are LPWANs and NB-IoT.

In a healthcare environment, there are many factors to be considered in a IoT communication standard. Firstly, the used standard should have no side effects on the human body that lead to health concerns to the patient. In order to minimize the time in which medical help arrives to the patient in need, specially in life or death situations, it is important to utilize a communication standard with low latency. In order to assure that the information is always received independently of the patient's location, the communication standard should have high availability [6].

Bluetooth is a connection oriented communication technology, where terminals must establish their connection when they communicate with each other. It was created by Ericson in 1994 and it has a maximum rate of 3 Mpbs and a frequency band of 2.4 GHz [86] and the advantage of using low radiation which is less harmful to humans [86]. It has been used in IoT systems such has the ones present in [68] (a blood pressure monitoring system) and [12] (an home patient monitoring system for early detection of Alzheimer's disease).

According to [39], the terminals of the systems that use Bluetooth can either be "Discovering" or "Discoverable" and the connection must be established by a "Discovering" terminal and an adjacent "Discoverable" terminal.

Each terminal changes its status periodically by using a randomly chosen duration. This can increase the connection establishment latency if the terminals have the same status.

Bluetooth originated **Bluetooth Low Energy**, one of the two main standards used as short range communication in a IoT system. BLE is developed by Bluetooth Special Interest Group (SIG) with the goal of providing an energy-efficient standard capable of using coin-cell battery operated devices. BLE is very helpful in IoT, since it connects peripheral devices to processing devices like smartphones that contain Bluetooth related applications.

The topology used in BLE related systems is the star topology. The star topology is characterized by a central node connected to sensors. The data is obtained by the sensors, that send it to the central node, where it is collected and processed. The sensors do not communicate with each other.

In an open-field, BLE has a range of 150 meters [9] and in non-ideal conditions its range is around 10 meters [29], which is enough distance to reach the nodes. Its range, low latency and high data rate makes BLE suitable for emergency health. Due to operating in the 2.4GHz band, BLE is robust enough to noise to be used in a healthcare system [6].

BLE has many applications in different industries, such as healthcare [8, 55], smart energy and smart home domains [60, 16].

The connection mechanism in BLE is a major improvement in relation to the one used in Bluetooth. Instead of using “Discovering” and “Discoverable” terminals (active scan), BLE uses terminals that can be “Advertising” and “Scanning” simultaneously in order to find and be found by other terminals [39]. This mechanism allows very quick connections [60]. Another characteristic that proves that BLE is suitable for healthcare applications is its extremely low power consumption [6].

The other main standard for short range communication is the **ZigBee** standard, that was designed by the ZigBee Alliance with the goal of providing low-cost, low-powered networks for M2M communications. Even though it can also be used in a star topology, it is more common to find the Zigbee standard used in a mesh network. In a mesh network, there is no central node and all the nodes communicate with each other.

There are many ZigBee modules, such as XBee (the simplest), XBee Pro and XBee Pro 900 XSC. XBee is the preferable solution as a short distance communication since, although its range is inferior (30 meters), its power of transmission is the lowest (1 mW) and the data rate is the highest (250 kbps, same as the XBee Pro) [6].

ZigBee has been used in many IoT systems such as Escort, a safety monitor for people living with Alzheimer’s disease [75], and a Low-power wearable ECG monitoring system for multiple-patient remote monitoring [70].

Wireless Fidelity (Wi-Fi) was created in 1997 for non-IoT purposes. It is a very common standard in healthcare environments but, compared to other standards, it has a very high power consumption (over 1000 mW) [60]. Its range is 1 to 300 Mbps and its band is 2.4 G to 5GHz.

Radio Frequency Identification (RFID) is a communication standard that allows the identification of the devices from a distance using an electronic tag without the need of battery. This is useful in a healthcare environment, since the tag can contain the records of each patient and provide them easily to the medical professional [56]. RFID has a band of 13.56 MHz and its data rate is 106 to 424 Kbps [86].

Low-Power Wide-Area Networks (LPWANs) are a subset of long-range communications standards. Their range of several kilometers, low-power device design that minimizes the risk of patients being offline and ability to support short bursts of data infrequently make them highly suitable for IoT applications [6]. The most common LPWANs are Sigfox and LoRaWAN.

NB-IoT is a more recent standard. It is still being deployed and it is expected to become more prominent. Overall, NB-IoT is suitable for healthcare applications since it can support many devices, it is secure and has high energy efficiency [6].

In Table 2.2, a comparative study of communication standards in the literature is presented.

Table 2.2: Comparative table of Communication Standards.

Communication Standard	Maximum Data rate	Band of operation	Typical transmission distance	Energy consumption
Bluetooth	3 Mbps	2.4 GHz	20 m	High
Bluetooth Low Energy	2 Mbps	2.4 GHz	5 m	Very low
ZigBee	250 Kbps	2.4 GHz	30 m	Very low
Wi-Fi	300 Mbps	2.4 - 5 GHz	50 m	Very high
RFID	424 Kbps	13.56 MHz	20 cm	Very low
LPWAN	100 bps	868 MHz (EU) 915 MHz (US)	several Kms	Low
NB-IoT	250 Kbps	Various	15 Kms	Low

BLE is extremely well suited to healthcare applications due to its appropriate range, low latency, low power consumption and robustness to interference.

Compared to ZigBee, BLE is present in more devices such as smartphones. Furthermore, BLE is more secure, since the fact that the developer is the one to implement the key exchange of ZigBee, its security can be compromised. Even though this dissertation does not focus on security of the system, lack of security of one standard compared to another must weight on the choice of standard.

If the throughput required is of a higher value and power consumption and connection latency are not of great concern classic Bluetooth should be used [20].

Even though the long range communications are very helpful in IoT systems, the work of this dissertation does not require data transmission with a range greater than 5 meters, so this type of communication standards will not be further explored in this dissertation.

2.3 Healthcare Interfaces

Since IoT systems in healthcare generate large amounts of data, the IoT system needs to contain a component to represent it in a visual way that allows the healthcare professionals and patient to understand it [41]. It is then important to develop a interface to fulfil this goal. In this section, some examples of interfaces in healthcare IoT systems are presented.

The authors of [62] aim to present a system for local and remote patient monitoring using blood pressure monitor, pulse oximeter, air flow, galvanic skin response and body temperature. In order to analyse the data, this system requires a GUI that obtains it through Asynchronous JavaScript

And XML (AJAX) requests to the Node.js server. This interface contains a main page presented in Figure 2.1 that shows the latest value of each sensor and redirects to pages that contain charts with the history of values in real-time with functionalities like scaling, zooming and panning to aid data analysis. The interface can be accessed locally or through any device through the browser using the standard HTTP protocol on the default port 80.

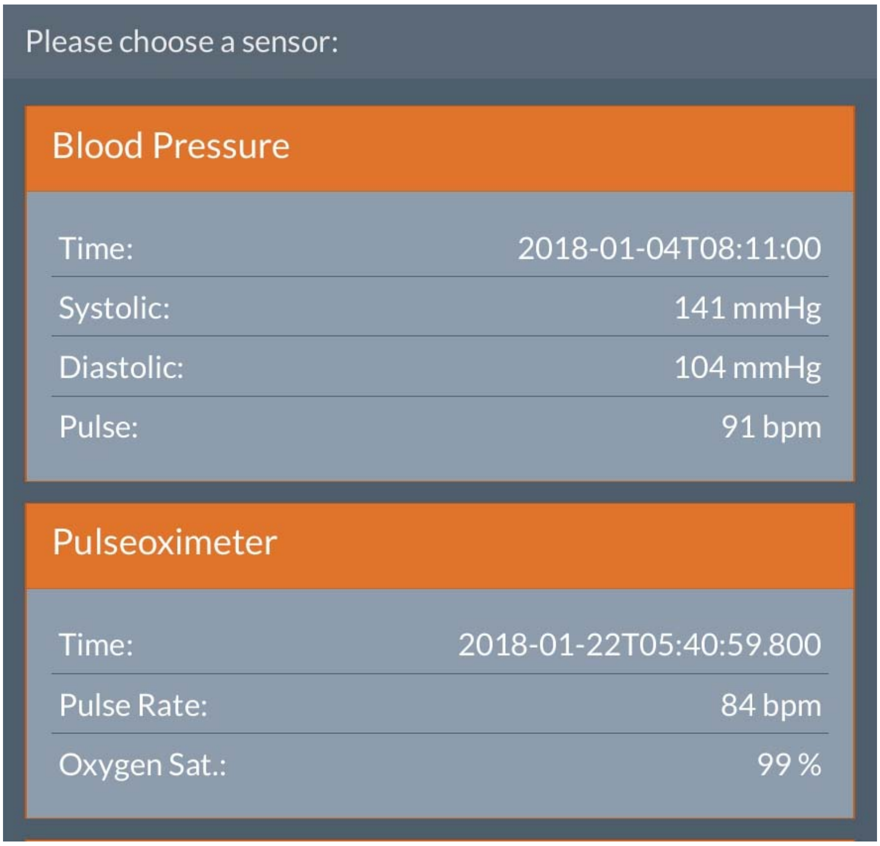


Figure 2.1: Main page of the interface of the system described in [62].

In order to monitor the physiological parameters of in-hospital patients, the authors of [42] opted for a simple interface, consisting of a table for each sensor data acquired, as shown in Figure 2.2. The intended users of the interface are the physicians, so they can analyse it and prescribe the appropriated medical management. The interface can be accessed through any device like a Personal Computer (PC), tablet or smartphone.

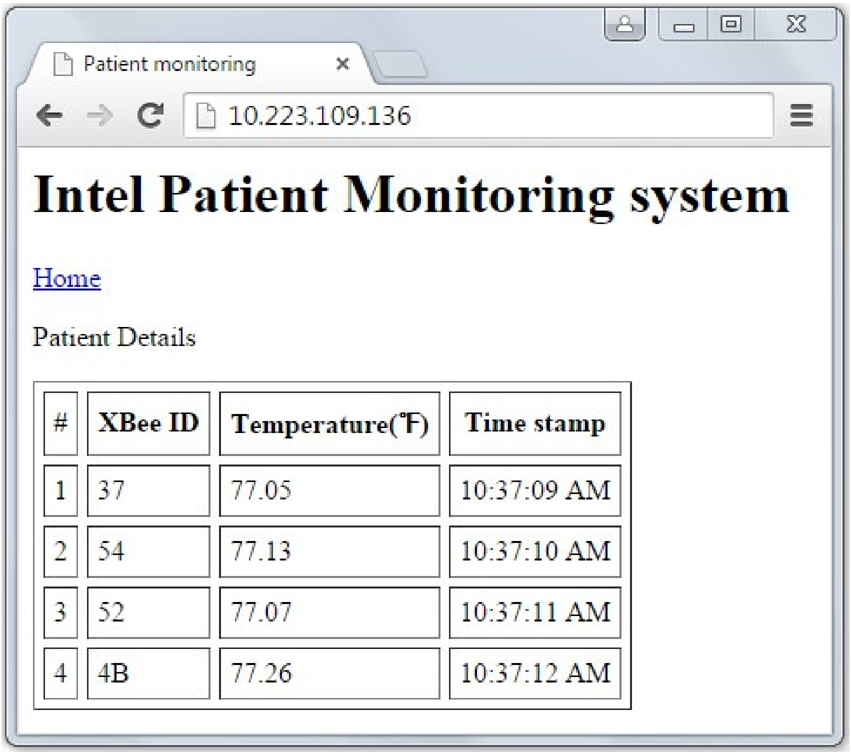


Figure 2.2: Patient Monitoring System in [42].

The goal of the system in [70] is to remotely monitor non-technical users in need of long-term health. In order to allow users and authorized clinicians to configure and control the entire system, a crucial interface was developed. This GUI can be accessed from any computer, smart phone or tablet connected to the internet. The ECG visualization page is shown in Figure 2.3.

Figure 2.3: ECG page from the Interface from the system described in [70].

In Table 2.3, a comparative study of healthcare interfaces in the literature is presented.

Table 2.3: Comparative table of Healthcare Interfaces.

Works	Data acquired	Language	Data Retrieval	Historical Data
Pap et al. [62]	blood pressure, pulse oximeter, air flow, galvanic skin response, body temperature	Node.js	AJAX Requests	yes
Khan and Khachane [42]	temperature	not stated	not stated	no
Spanò et al. [70]	ECG	not stated	not stated	yes

All the healthcare interfaces achieve the main goal of visualization of data in real-time, independently of the language in which the interface was developed or the method of retrieval of data. The functionalities of assessing the historical data through graphs in the interfaces from the systems [62, 70] and the ability to access the interface using any device from the interfaces [62, 42, 70] seem helpful and add value to the mentioned interfaces. The main drawback of the interfaces presented is their design, which is not appealing, specially the one from [42].

2.4 Healthcare System Architectures

Healthcare systems use a set of interconnected devices to create an IoT network devoted to healthcare assessment, including monitoring patients and automatically detecting situations where medical interventions are required [74]. A healthcare system can provide a variety of services, offering solutions to many healthcare problems, such as the lack of autonomy of older generations that can be helped with Ambient Assisted Living (AAL) systems [87] and injuries caused by taking medication that need an Adverse Drug Reaction (ADR) system [38].

In a healthcare system architecture, it must be represented the structure of the entire tested system, starting from the sensors and sensor circuitry, passing through the data acquisition circuit and software development until the design and implementation of data storage and visualization application. In this subsection, multiple healthcare systems architectures are explored.

The authors of [62] present a system based on a Raspberry Pi 3 that records blood pressure, pulse oximeter, air flow, galvanic skin response and body temperature data and presents it in a graphical interface that is explored in Section 2.3. The main component of the system is a Node.js server-side application running on the board, that is responsible for controlling the entire system, from the acquisition of the sensors data, explored in Section 2.1, to the web interface, explored in Section 2.3.

The authors of [59] propose the general architecture with a front end that is responsible for acquiring healthcare data from the patient and transmitting it to the backend securely and in a privacy-preserving fashion. For the acquisition of the data the authors propose the usage of the main communication standards Bluetooth Low Energy and ZigBee. The backend of the system is responsible for storing and processing the data securely.

The paper [75] presents the Escort System, a safety monitor for people living with Alzheimer's disease, that often exhibit wandering behavior, that increases the likelihood of accidents, serious injury, and even death. This system communicates the patient's location to a central server and can communicate alerts to caregivers. The communications standards used are ZigBee in a mesh

Table 2.4: Comparative table of Healthcare Architectures.

Works	Goal	Acquired Data	Communication Technologies	Devices	Highlights
Pap et al. [62]	Propose a data acquisition, storage and visualization solution	blood pressure, pulse oximeter, air flow, body temperature	not stated	Raspberry Pi 3	Modular approach on the acquisition, seamless integration of multiple types of sensor, node.js as central component of the architecture
Page et al. [59]	Propose a data acquisition, storage and visualization solution	not stated	BLE, ZigBee	not stated	Machine Learning in the Back end
Taub et al. [75]	Monitor Alzheimers patients	location	ZigBee	Night Light Location Beacon	Mesh network
Spanò et al. [70]	Remote ECG monitoring	ECG	BLE, ZigBee, Wi-Fi	Gateway connected to ADSL router	It can be integrated with other systems
Nikolaevskiy et al. [57]	Transfering mobile data securely to healthcare services	ECG, glucose sensors, RFID tags, insulin pumps, accelerometers	not stated	Portable medical terminal	One Gateway for each patient

network that can support at least 20 to 30 nodes and cellular networks. Instead of using vital signals sensors like in the others systems presented in this dissertation, this system uses Night Light / Location Beacon as the provider of data.

The system presented in [70] has the goal of monitoring multiple non-technical users in need of long-term health monitoring in residential environments remotely with a low power wearable ECG. According to the authors, the solution presented has the advantages, compared to the state of the art, of using ECG prototype sensors with record-low energy per effective number of quantized levels, providing low marginal cost per added sensor/user and offering the possibility of seamless integration with other smart home systems through a single IoT infrastructure.

The system proposed in [57] presents a WBAN for each patient that consists of wearable and implantable devices like ECG, glucose sensors, RFID tags, insulin pumps and accelerometers. The main different characteristic of this architecture is that there is one Gateway for each patient, instead of one Gateway for many patients. The Gateway is responsible for the aggregation, processing and transferring of the data to the healthcare system.

In Table 2.4, a comparative study of healthcare architectures in the literature is presented.

The system on the paper [62] has the benefit of showing that an interface can be developed using node.js and could be improved by turning the acquisition wireless.

The system from [59] could be more detailed in the types of devices that the authors intend to use. For example, the insertion of another layer with a data processing device in the architecture could be beneficial.

The Escort system present in [75] could benefit from the inclusion of a wider variety of sensors, instead of only using the position of the users, so that the approach to ensure the safety of the patients can be broader.

The system on [57] uses one Gateway for each patient. In order to make studies with the data from different users, the authors could consider using the same Gateway for different users.

2.5 Contributions

As stated before, this dissertation has the goal to propose a DAS to be used in a Smart Box that is responsible for data acquisition, storage, transmission and visualization. This IoT node is part of a system to wirelessly monitor patients as a step towards domiciliary hospitalization.

Regarding the fragilities of the state of the art and in order to achieve an innovative system architecture to wiressly monitor untethered patients, this dissertation proposes the development of a Smart Box firmware capable of:

- Reliable and continuous low-level data acquisition from a series of biomonitoring stickers for patients, including fully untethered, simple and very low-cost printed stickers, using BLE, inspired by the method described in [84]:
- Data transmission to the Gateway through MQTT, using specific topics for each type of sensor transmitted.
- Raw data storage in an internal MongoDB database to feed a GUI and to assure data redundancy;
- Deploying an interface using node.js, HTML and CSS that allows to visualize real-time and historical data using web a browser.

In addition to the firmware development, the dissertation has the contribution of experimental validation in real tests in an healthcare environment.

In Figure 2.4, an overview of the proposed system in this dissertation is presented. In the next chapters, the methodology and implementation is detailed.

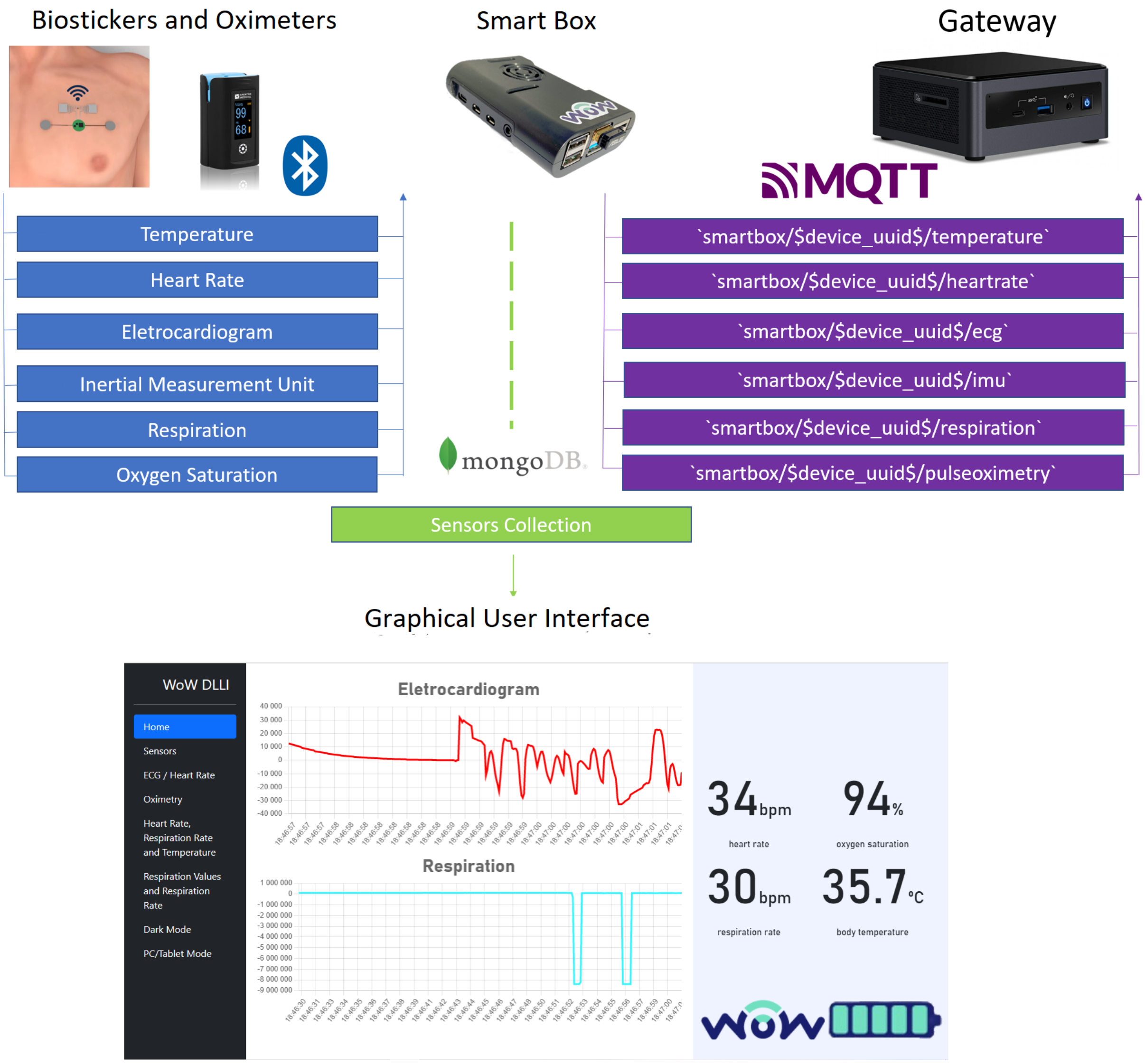


Figure 2.4: Overview of the system described in this dissertation.

3 System Architecture and Methodology

The work detailed in this dissertation was developed as part of the **W**ireless bi**O**monitoring stickers and Smart Bed architecture: to**W**ards Untethered Patients (WoW) project.

The overall goal of the WoW project is to demonstrate wireless patient biomonitoring and centralized data collection, processing and transmission, as a step towards domiciliary hospitalization. This goal is accomplished by developing a system architecture that contains thin-film biomonitoring stickers that obtain data that is then collected, processed and transmitted to the most utilized software in the Portuguese hospitals, Globalcare, a proprietary Hospital Information System developed by the project leader Glintt.

The architecture proposed in the WoW project is presented in Figure 3.1. The blue square in the image highlights the components and interfaces in focus within the work developed in this dissertation.

The architecture has a star topography where the Gateway is the central node and the Smart Boxes embedded in hospital beds are the nodes connected to it. One patient-Biosticker pair is assigned to each Smart Box.

To aid to a better understanding of the work that originated this dissertation, background concepts related to the components of the architecture.

3.1 System Components

3.1.1 Biosticker

The Biosticker, represented in Figure 3.2, is a wearable device that combines the sensors that register the vital signs Temperature, Heart Rate, Electrocardiogram (ECG), Inertial Measurement Unit (IMU) and Respiration. The Biosticker is comfortably attached to the patients skin without restricting his/her movements, collects the data and sends it fully wirelessly through BLE to an IoT node named Smart Box. According to multiple works [6, 65], the development of accurate sensors is essential to the success of the system developed, since they are responsible for the measurement

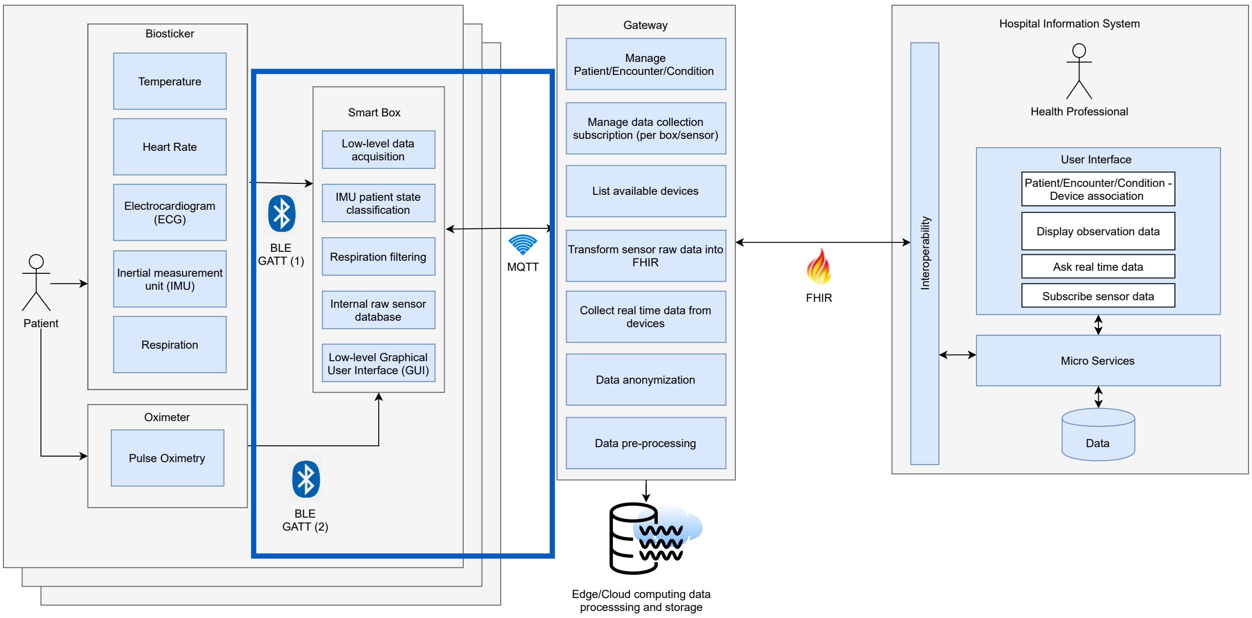
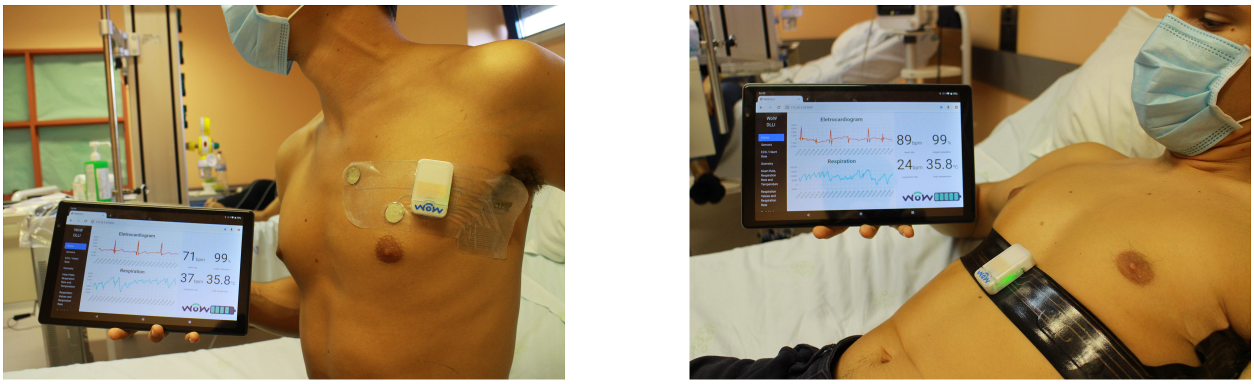


Figure 3.1: WoW system architecture [83].



(a) Biosticker composed by an adhesive material.

(b) Biosticker placed above the waist using a belt with several adhesives.

Figure 3.2: Biostickers placed on the patient's skin.

of the essential signs for determination of critical health such as pulse, respiratory rate, and body temperature.

Temperature

Body Temperature is one of the vital signs and its values provide to the wearable healthcare system the possibility of predicting hypothermia, heat stroke, fevers and other diseases.

Previous works have shown that it is possible to implement reliable temperature measurement for remote health monitoring [47, 58, 43].

The sensor model used is the Maxim Integrated MAX30205 human body temperature sensor [19, 18]. It is used for fitness and medical applications. Its main features are an accuracy of 0.1°C in the range 37°C to 39°C, a 16-Bit temperature resolution and a 600A (typ) Supply Voltage

Range.

Heart rate and Electrocardiogram

The Heart rate reflects the pacing of the heart (beats per minute), by counting the beats of the artery within 1 minute [14].

Electrocardiogram (ECG) is an non-invasive exam that evaluates the cardiac function, using an electrode to measure and record the electrical activity of heart muscle. It is one of the most widely used technique to determine heart beat response in real-time through the analysis of its five components named PQRS and T [65].

Multiple IoT systems [30, 45, 65, 85] have Heart Rate and ECG sensors integrated in their architecture.

The model used to collect both Heart Rate and ECG data is the MAX30003 [48], which is a complete, biopotential, analog front-end solution for wearable applications, consisting of a single biopotential channel providing ECG waveforms and heart rate detection. It can be used for clinical and fitness applications.

Inertial Measurement Unit

An inertial measurement unit (IMU) is an electronic device composed by accelerometers, gyroscopes and magnetometers that measures and reports a body's specific force, angular rate, and sometimes the orientation of the body.

Clinically, this type of sensor is very helpful in the detection of falls and it also has fitness applications, such as the evaluation of fatigue [63] and human step-counting [78, 76].

The IMU sensor used is the LSM6DS3 [46], that features a 3D digital accelerometer and a 3D digital gyroscope. This sensor performes at 1.25 mA (up to 1.6 kHz ODR) in high-performance mode and enables always-on low-power features. The LSM6DS3 is robust to mechanical shocks, has a full-scale acceleration range of ±16 g and an angular rate range of ± 2000 degrees per second (DPS).

Respiration

The measurement and monitorization of the respiratory rate can aid in the identification of conditions such as asthma attacks, hyperventilation due to panic attacks, apnea episodes, lung cancer, obstructions of the airway, tuberculosis and more [6].

The solution used to collect respiration data is a strain gauge that measures the changes in resistance that varies with the changes in strain [54] caused by the respiratory cycle.

3.1.2 Oximeter

Pulse oximetry measures the level of oxygen in the blood, serving as an indicator of respiratory function. This vital signal has been integrated in many IoT systems with various goals, such as monitoring cardiac activity and heart related diseases [53] and early detection of Covid-19 Symptoms [33].

Two models of oximeter are used: PC-60FW [50, 32] and Wellue FS20F [81]. These oximeters have OLED display, are resistant to water splashes, drops and mechanical shocks and are portable and durable. These instruments are capable of continuous measurements followed by transmission through BLE. The data can be visualized in the OLED display and in their respective mobile applications. In addition to pulse oximetry, both oximeters also provide pulse rate.

3.1.3 Smart Box

The Smart Box is the main component used in the work of the dissertation. The system architecture envisages a Smart Box for each patient. This Smart Box is responsible for various modules listed below:

- the low-level acquisition of data obtained from the Biostickers referenced in the previous sections, using BLE;
- raw data storage in a MongoDB database;
- data prepocessing such as calculating the respiration rate from the respiration values;
- low-level interface that allows the WoW project team and the healthcare professionals to check real-time and historical data using a browser.

As explored in Chapter 2, Raspberry Pi has been used in many systems, being commonly used in he IoT community obtaining positive reviews. Allied with its easy replication, scalability and low cost, the Raspberry Pi 4 Model B was chosen to be used as the Smart Box. The Smart box is shown in Figure 3.3 and its board specification is shown in Table 3.1. This device is compatible with Bluetooth 2.1/3.x/4.x.

Figure 3.3: Smart Box.

Table 3.1: Smart Box board specification.

Specifications	Raspberry Pi 4 Model B
Memory	8 GB RAM LPDDR4
CPU	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8)
DC Input Voltage Supported	5 V / 3 A
GPU	Broadcom VideoCore VI (100 Hz)
Mass Storage	32 GB SD Card
Operating System	Ubuntu Server 20.04.2 LTS
Bluetooth	BLE 5.0 (internal adapter)

3.1.4 Gateway

The gateway is responsible to collect real-time data from multiple Smart Boxes using the MQTT protocol. The data is acquired in JSON format so it is accessible to higher level modules, e.g. for data analysis.

Other responsibilities of the Gateway in the WoW system are:

- managing sensor subscriptions per Smart Box;
- providing a constantly updated list of all Smart Boxes and sensors available;
- converting the data to Fast Healthcare Interoperability Resources (FHIR) specification, the format defined for the connection with the Health Information System (HIS), ensuring interoperability;
- pseudo-anonymizating the data to protect the privacy of the patients;
- preprocessing the data to detect critical conditions of patients' state so it can be notified to the health professionals. This allows for a quicker reaction from the health professionals, since the delays caused by the transmission to higher level modules is eliminated.

- managing interactions between an healthcare provider and a patient, such as attaching and detaching devices to the patient and update encounters

The choice of Gateway was made with similar criteria as the choice of Smart Box. The Intel NUC 8i7BEH mini-PC [34] was chosen to host the Gateway services and its specifications are presented in Table 3.2.

Table 3.2: Gateway mini-PC specification.

Specifications	Intel NUC Kit NUC8i7BEH
Memory	16 GB RAM DDR4-2400 1.2V SO-DIMM
CPU	Intel Core i7-8559U Processor (8M Cache, up to 4.50 GHz)
DC Input Voltage Supported	12-19 VDC
GPU	Iris Plus Graphics 655
Storage	1 TB SSD
Operating System	Ubuntu Server 20.04.2 LTS

3.1.5 Tablet

In order to visualize the acquired data in a GUI, tablets, smartphones or any device with access to a web browser can be used. The tablet models used were YESTEL MODFEL T5 and VANKYO MATRIXPAD S20.

3.2 Communications

In this section, information about the communications between the components of the architecture is provided.

3.2.1 Bluetooth Low Energy

The real-time data acquisition was achieved using BLE with GATT. As stated in Chapter 2, BLE is extremely well suited to healthcare applications due to its appropriate range, low latency, low power consumption, robustness to interference and security. These characteristics make BLE an adequate communication standard to be used between the Biosticker and Smart Boxes.

GATT controls the connections and advertisements between devices, dividing it in multiple layers [84]:

- Profile: series of services;

- Services: collections of characteristics, helping to better organize the information;

- Characteristics: lowest level concept in GATT transactions, composed by the following attributes:
  - Handles: 16-bit identifiers that function as connection points where data can be read or written;
  - UUID: 16 bytes unique identifier;
  - Properties: access permissions that define how the data can be used. Its options are:
    * Read: the characteristic can be read by the client;
    * Write: the attribute can be written by the client;
    * Notify: the client asks the server to notify it every time the value changes;
    * Indicate: Similar to notification, with the addition of a confirmation in the server that the message has reached the client.
  - Value: value of the attribute.

3.2.2 Message Queuing Telemetry Transport

Wi-Fi is used together with the MQTT protocol to transmit the data from the multiple Smart Boxes to the Gateway. MQTT was chosen for the communication since it is considered an adequate protocol for low-power efficient data transmission with fast message delivery.

This protocol is specifically designed for M2M communications and uses a publish/subscribe model to transmit data in various forms, such as binary data, text, Extensible Markup Language (XML), or JavaScript Object Notation (JSON).

To establish a communication using the MQTT protocol, three entities are needed: the publisher, responsible for sending the collected data; the subscriber, that receives the data; the broker, that coordinates the exchange of data.

3.3 Methodology

3.3.1 Data Acquisition from Biostickers

The distinct phases of implementation are explored in this subsection.

To be able to connect to a device and acquire its data, the first step is to have a deep understanding of its services and characteristics.

The Biosticker GATT profile definition is presented in Table 3.3.

Table 3.3: Biosticker GATT Profile.

Service	Characteristic	Handle	UUID	Property
Device Information	-	41	0000180a-0000-1000-8000-00805f9b34fb	-
Vendor specific	-	27	0000a000-0000-1000-8000-00805f9b34fb	-
-	Vendor specific	39	0000a006-0000-1000-8000-00805f9b34fb	read, write
-	IMU	36	0000a004-0000-1000-8000-00805f9b34fb	read,notify
-	ECG RAW	33	0000a003-0000-1000-8000-00805f9b34fb	read, indicate
-	RR ADC	30	0000a002-0000-1000-8000-00805f9b34fb	read, notify
-	Vendor specific	28	0000a001-0000-1000-8000-00805f9b34fb	read,write
Health Thermometer	-	21	00001809-0000-1000-8000-00805f9b34fb	-
-	Temperature Type	25	00002a1d-0000-1000-8000-00805f9b34fb	read
-	Temperature Measurement	22	0000a004-0000-1000-8000-00805f9b34fb	read,indicate
Battery Service	-	17	0000180f-0000-10000-8000-00805f9b34fb	-
-	Battery Level	18	00002a19-0000-1000-8000-00805f9b34fb	read, notify
Heart Rate	-	11	0000180d-0000-1000-8000-00805f9b34fb	-
-	Body Sensor Location	15	00002a38-0000-1000-8000-00805f9b34fb	read
-	Heart Rate Measurement	12	00002a37-0000-1000-8000-00805f9b34fb	notify
Generic Attribute Profile	-	8	00001801-0000-1000-8000-00805f9b34fb	-
-	Service Changed	9	00002a05-0000-1000-8000-00805f9b34fb	indicate

There are multiple characteristics with properties `indicate` and `notify` in the Biosticker GATT profile. The measurements of the Vendor specific Characteristics can only be accessed after writing on the characteristic Vendor specific with the handle 28. The written value dictates which types of sensor can be read. The other characteristics with `indicate` or `notify` property can be subscribed right away.

In order to improve its energy consumption, the Biosticker requires the receival of a keep-alive flag in the characteristic with the handle 39 periodically to maintain the connection with the Smart Box.

Once the GATT profile is deeply understood, the second step is to automate the data acquisition and assuring that, in case of disconnections, the devices reconnect automatically.

The actual implementation of these steps is further detailed in Chapter 4.

3.3.2 Data Acquisition from Oximeters

The PC-60FW oximeer GATT primary services are presented in Table 3.4. The PC-60FW Nordic UART Service provides the `notify` characteristic that contains the data. In order to subscribe to it, the Smart Box first writes "0x0d" in the `read` characteristic.

For the second oximeter used in this work, the Wellue FS20F GATT profile is not as clear as the other devices GATT profiles, since it contains multiple services and characteristics marked as "Unknown" that are bespoke to the manufacturer. In its GATT profile, there is a `notify` characteristic that contains the data and can be subscribed without the need to write in another characteristic as in the previous devices.

Table 3.4: PC-60FW GATT Primary Services.

Service	UUID
Generic Access	0x1800
Generic Attribute	0x1801
Nordic UART Service	6e400001-b5a3-f393-e0a9-e50e24dcca9e

3.3.3 Data Preprocessing

Once the data is acquired by the Smart Box, it is important to understand the format of the values so they can be stored and transmitted in a proper format.

In Table 3.5, information about the data such as data size (number of bytes received in each value), data range (minimum and maximum interval of the value) and units of measure are presented.

Regarding Temperature, the data received is composed by 1 informative byte (Flags field) followed by 4 bytes that correspond to the value. Bit 0 of the Flags field informs the units of measure. If the bit 0 is set to 0, the Temperature is in Celsius. If the bit 0 is set to 1, the Temperature is in Fahrenheit. Bit 1 of the Flags field informs if a timestamp is supported (1 if it is supported and 0 if it is not). Bit 2 of the Flags field informs if the Temperature Type value is supported. If it is 1, it is supported and it includes the Timestamp field in the Intermediate Temperature characteristic. If it is 0, the Temperature Type value is not supported. The first byte is always "00000000".

Regarding the Electrocardiogram, each data received corresponds to 10 values. Each pair of consecutive bytes (0 and 1, 2 and 3, 4 and 5 and so on) correspond to a value of ECG.

Table 3.5: Acquired data information.

Signals	Data size	Data range	Unit of measure	Endianness	Data type
Heart rate	2 bytes	0 to 65535	beats per minute	bid endian	int
Temperature	5 bytes	-3.4×10^{38} to $+3.4 \times 10^{38}$	°C	little endian	float IEEE 11073
Respiration	4 bytes	0 to 4.294.967.295	none	big endian	int
ECG	20 bytes	-32768 to 32767	none	big endian	int
IMU	24 bytes	-3.4×10^{38} to $+3.4 \times 10^{38}$ for each value	g (1 g = $9.8m/s^2$) (Bytes 0:11) and degrees/sec (Bytes 12:23)	none	float
Battery	1 byte	0 to 255	percentage	big endian	int
Pulse Oximetry	it varies	it varies	percentage	none	int

Regarding IMU, the Bytes 0:3 correspond to the x axis value of the linear acceleration. Bytes 4:7 correspond to the y axis value of the linear acceleration. Bytes 8:11 correspond to the z axis value of the linear acceleration. Bytes 12:15 correspond to the x axis value of angular acceleration. Bytes 16:19 correspond to the y axis value of angular acceleration. Bytes 20:23 correspond to the z axis value of angular acceleration.

After understanding the values acquired, they are translated to be stored and transmitted as explored in the next sections. After translation, the Respiratory Values are used to calculate Respiratory Rate with an algorithm based on the Welch's method.

3.3.4 Data Transmission

All messages exchanged must be in JSON format, and must have the structure present in Figure 3.4. The field "client_id" is unique for each Smart Box, and its value can be found in Table 5.2 as the Smart Box UUID.

```
{
    "client_id": client_uuid,
    "timestamp": timestamp,
    "message_type": message_type,
    "payload": {
        //...
    },
}
```

Figure 3.4: MQTT messages JSON format.

The field ‘message_type’ defines the type of message sent, and must be one of the following: "measurementTEMPERATURE", "measurementECG", "measurementOXIMETRY", "measurementRESPIRATION", "measurementIMU", and "measurementHEARTRATE".

The field ‘payload’ must contain the message’s content and its format varies depending on he type of sensor data communicated:

- Temperature:

```
"payload": {
        "temperature" : 10.0, //temperature measurement
        "is_celsius" : true // indication if measurement
                              is in celsius or fahrenheit
}
```

- Inertial Measurement Unit (IMU):

```
"payload": {
    "imu": {
            "linear_acceleration": {"x": 0.00, "y": 0.00, "z": 0.00},
            // accelerometer measurement
            "angular_velocity": {"x": 0.00, "y": 0.00, "z": 0.00}
            // gyroscope measurement
        }
    "pose_description" : "SITTING"
}
```

- Electrocardiogram (ECG):

```
"payload": {
        "ecg" : 10 // ecg measurement
}
```

- Pulse Oximetry:

```
"payload": {
        "spo2" : 10.0 // spO2 measurement
}
```

- Heart Rate:

```
"payload": {
        "bpm" : 10.0 // bpm measurement
}
```

- Respiration Rate:

```
"payload": {
            "respiration" : 10.0 // respiration rate
                                          measurement in %
}
```

Each message must be communicated in a topic, a UTF-8 string that the broker uses to filter messages for each connected client. Each sensor of each Smart Box uses its specific topic. The device_UUID of each Smart Box is present in Table 5.2.

Table 3.6: MQTT Topics.

Sensor	MQTT topic
Temperature	'smartbox/*device_UUID*/temperature'
IMU	'smartbox/*device_UUID*/imu'
ECG	'smartbox/*device_UUID*/ecg'
Pulse Oximetry	'smartbox/*device_UUID*/pulseoximetry'
Heart Rate	'smartbox/*device_UUID*/heartrate'
Respiration	'smartbox/*device_UUID*/respiration'

As referred previously in Chapter 2, this dissertation does not focus on security of the system. The only type of security implemented in the data transmission is the assignment of a X.509 certificate to each Smart Box that allows the gateway to validate its identity before communicating any information, thus preventing unauthorized access to malicious actors through attacks such as Man-in-the-Middle (MiTM).

3.3.5 Data Storage

After being acquired and translated as explained in the previous sections, the data must be stored.

The software chosen to function as a storage database in the WoW project system is the MongoDB [52]. This software has the advantage, compared to relational databases, of allowing flexible document schemas and using a change-friendly design.

The data is saved in the MongoDB database using a JSON srtucture composed of sensor_name, value and timestamp. The sensor_name in the database can be "Heart Rate", "Temperature", "Battery Level", "Respiratory Values", "Respiratory Rate", "ECG", "IMU Accelerometer X", "IMU Accelerometer Y", "IMU Accelerometer Z", "IMU Gyroscope X", "IMU Gyroscope Y", "IMU Gyroscope Z", "Oxygen Saturation" or "Pulse Rate".

All the data stored is acquired from the Biostickers or the oximeters, except the Respiratory Rate that is calculated using the acquired Respiratory Values, as mentioned in Subsection 3.3.3.

Continuous data acquisition results in large volume of data in the database, which can result in delays in its retrieval. To prevent that, compound indexes must be created. In a compound index, a single index structure holds references to multiple fields within a collection's documents [51].

3.3.6 Data Visualization

In order to visualize the large amount of data acquired in real time, a GUI was developed. A web interface was used due o its easier access and use, since it is available in any device with a browser. The web interface is supported by the MongoDB database.

The graphical user interface of the WoW system is composed by a plethora of screens that can be accessed through a browser with internet connection using each Smart Box fixed Internet Protocol (IP) and the port 8081.

To facilitate the usage of the interface by the healthcare professionals, icons were placed in the desktop that connect directly to the intended Uniform Resource Locators (URL).

4 System Implementation

In this Chapter, the Smart Box functionalities implemented in this dissertation are detailed.

4.1 Data Acquisition from Biosticker

As referred in Section 3.3, the first step in data acquisition is to obtain the BLE device's GATT profile. Ubuntu offers many tools to study the BLE device's GATT profile, manipulate its attributes, configure connections and send special commands to such devices, such as `hcitool` [49], `bluetoothctl` [44] and `gatttool` [37].

From the referred above, the main tool used was `bluetoothctl`. Firstly, the bluetooth service of the Raspberry must be enabled and activated. `bluetoothctl` searches for Bluetooth devices and connects with the Biosticker, once it is properly working.

The peripheral device (Biosticker) sends advertisement packages and the central device (Smart Box) discovers them and may decide to initiate the connection. Once this is done, it sends a connection request o create the connection. After a period of time, called connection interval, the central device (that now is considered the master) sends a data packet to the peripheral device (now named slave). When the slave receives the data packet, it sends another one and the connection is established. This connection mechanism is represented in Figure 4.1.

Once the connection is established, the services and its characteristics can be listed. The user can select with `bluetoothctl` the desired `indicate` and `notify` characteristics referred in Section 3.3 using its UUID and request notifications when its value varies, starting the data acquisition.

Once there is a deep understanding of the GATT profile and the characteristics that provide the values of the data, the automation of data acquisition can be implemented. The programming language used in automation of the data acquisition is Python 3.0, using the `asyncio` and `Bleak` [7] libraries, among others.

Bluetooth Low Energy platform Agnostic Klient (`Bleak`) is a GATT client software, capable of connecting to BLE devices acting as GATT servers [7]. `Bleak` contains many crucial functions

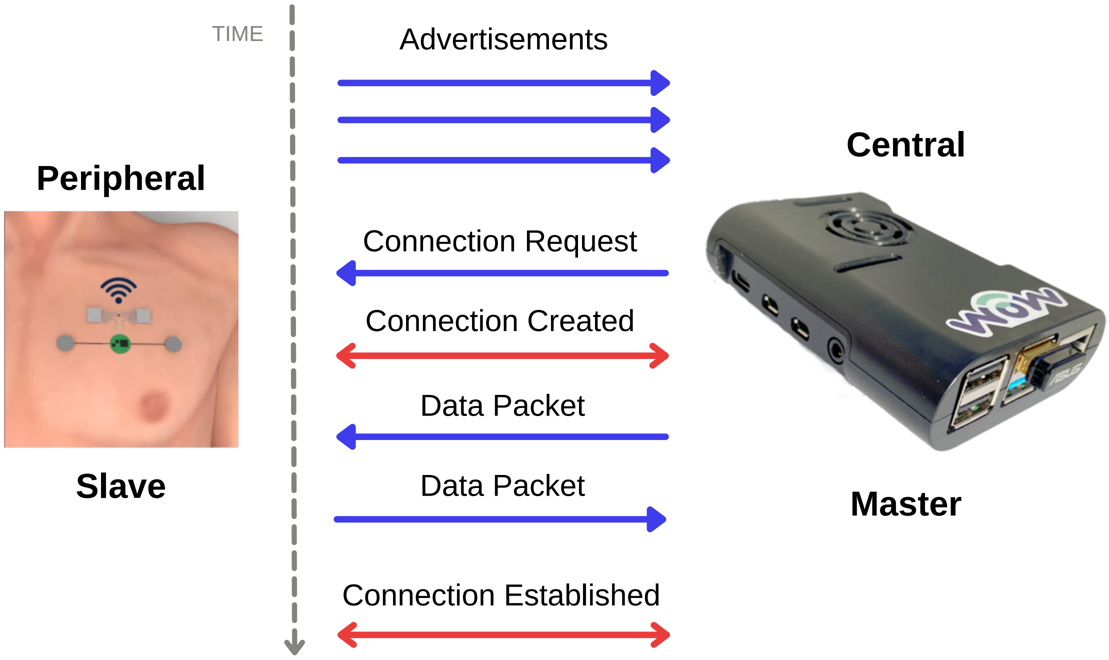


Figure 4.1: BLE connection mechanism.

in communications with BLE devices such as scanning, reading, writing and getting notifications from GATT servers.

The library `asyncio` is used to write concurrent code using the `async/await` syntax [24]. The core of every `asyncio` application is event loops. Event loops run asynchronous tasks and callbacks, perform network IO operations, and run subprocesses [25]. Considering that the acquisition of the multiple sensor types must be accomplished simultaneously, this is a very useful library.

The first step in the automation of the acquisition is to create a `Bleak` client using the MAC Address of the respective kit, ensuring that the connection is established with the desired device, and an external adapter. This external adapter is a Bluetooth dongle named Asus USB BT-500 Bluetooth 5.0 Adapter [4] that allows the acquisition of messages with more than 23 bytes. Since the ECG messages have 24 bytes, the external adapter must be assigned to the Biosticker acquisition and the internal is assigned to the oximeter acquisition.

Then, an event loop must be created that runs a function until it is completed or it raises an exception. This function is responsible for the connection with the device, writing in the characteristic with the handle 39 that provides access to the data, writing the keep-alive flag in the characteristic with handle 28 and subscription of the characteristics.

Finally, to ensure continuous acquisition for multiple hours without resetting the system, the previous steps are implemented inside a simple loop. When the connection fails for a period of time larger than 7 seconds, a new `Bleak` client is created and the acquisition process is restarted.

The implementation of data acquisition from Biostickers is summarized in Figure 4.2.

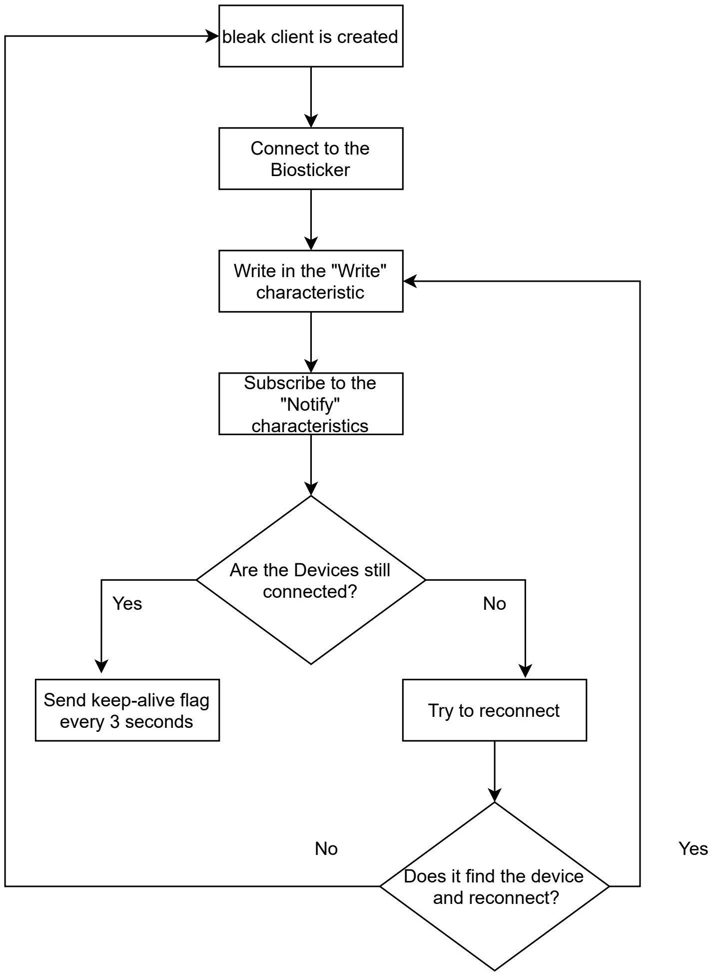


Figure 4.2: Flowchart of the implementation of data acquisition from Biostickers.

4.2 Data Acquisition from Oximeters

The `asyncio` and `Bleak` libraries in Python 3.0 was also used in the automation of the data acquisition from the oximeters. The main difference in the implementation is that in the creation of the `Bleak` client, the adapter used is the internal one of the Smart Box. The acquisitions from Biosticker and oximeters use different adapters (internal and external) to allow them to occur simultaneously.

Then, an event loop is created and runs the function that subscribes and writes to the respective characteristics of the oximeter used. The previous steps are implemented inside a loop, ensuring that a new `Bleak` client is created if the connection fails.

4.3 Data Preprocessing

Regarding the translation of the original value acquired to readable values, python offers a method `int.from_bytes()` that returns the integer represented by a given array of bytes, respecting their endianness. This method can be used with the values from sensors heart rate, respiration, ECG and battery. Regarding the temperature values, a function that translates `float IEEE 11073` to `float` values was used. To translate the IMU packed binary data the module `struct` was used and the module `Stream.extend()` was used to translate the oximeter data.

The calculation of the respiration rate value is done using a vector of respiration values that corresponds to 9 seconds of acquisition, that are subjected to an algorithm based on the Welch's method [80]. The data is filtered with a BandPass Filter. Then, instead of doing Fast Fourier Transform (FFT) in the entire vector, snippets of 3 seconds and overlap of 50% are used in multiple FFT, as shown in Figure 4.3 [15], obtaining each snippet's power spectrum. Then, the Power Spectral Density (PSD) is calculated by averaging the power spectra of each snippet. The final respiratory rate that is stored in the database is the median of the last 3 values calculated. A new value is provided once every three seconds.

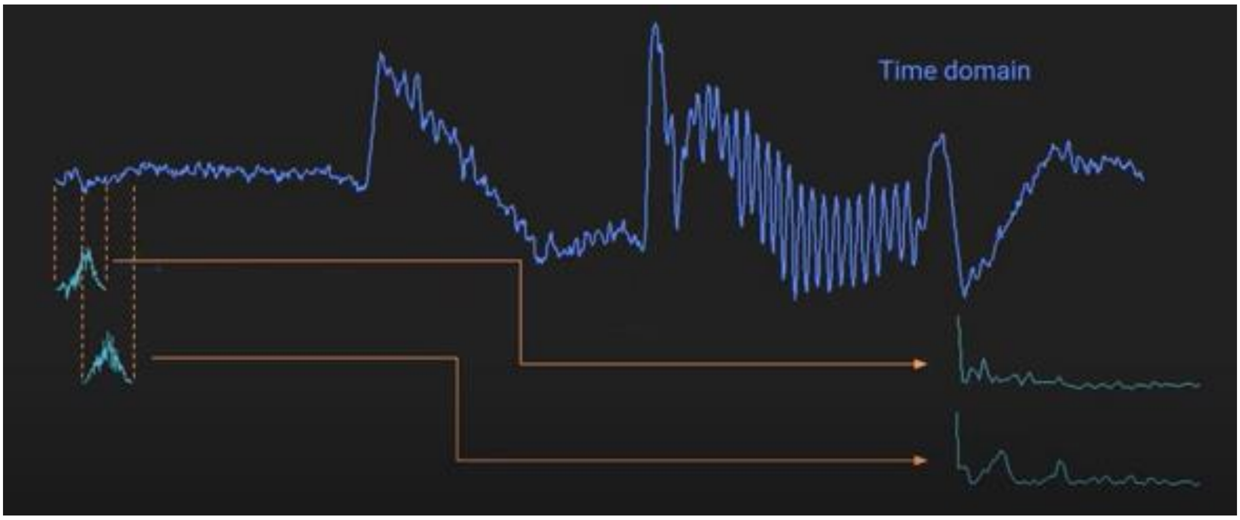


Figure 4.3: Example of snippets and respective power spectrum used in Welch's method.

The implementation of data preprocessing is summarized in Figure 4.4.

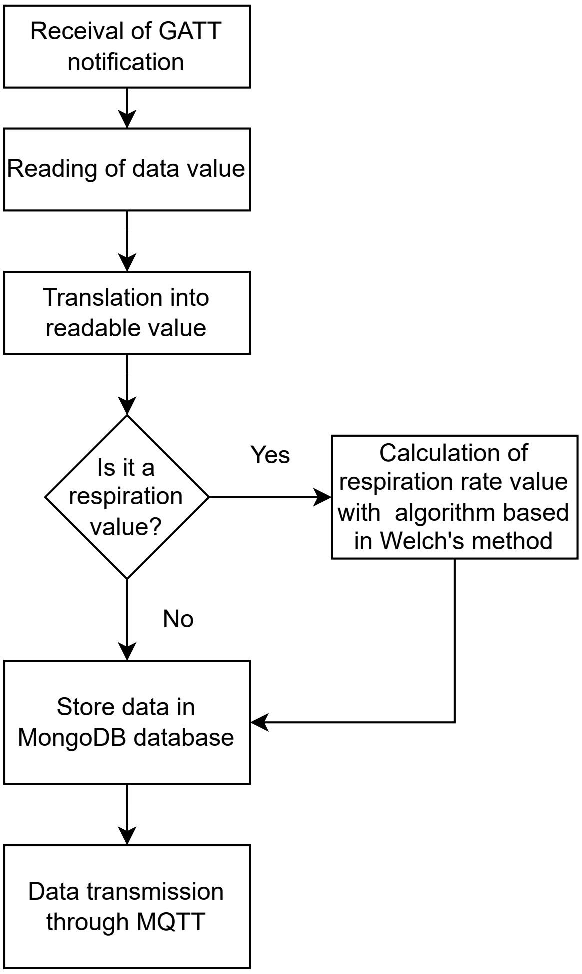


Figure 4.4: Flowchart of the implementation of preprocessing.

4.4 Data Transmission

Paho-mqtt 1.6.1 was used in the implementation of the Wi-Fi communications using MQTT in Python. This library enables applications to connect to an MQTT broker to publish messages, to subscribe to topics and receive published messages [26].

The first step is to create a MQTT client instance and connect it to a broker. If the connection is established, the broker sends a message confirming it. Then, when a value is ready to be transmitted, the JSON message is prepared, inserting the payload according to the specification above explained. Finally, the message is published to the broker and a confirmation is received.

4.5 Data Storage

Pymongo [27] was used to implement the MongoDB database since it contains tools to interact with it using Python.

Firstly, a MongoDB client is created and the MongoDB database named Smart Box_database is initialized. The next step in the implementation is to create a collection, a grouping of MongoDB documents, named sensors_collection. Once the database and its collection are created, the JSON documents can be prepared with the values acquired and inserted into the database. The mongo Shell [82] allows to choose a collection and create the compound index with the fields timestamp and sensor_name.

4.6 Data Visualization

The interface consists of a Node.js server. Node.js is an open source development platform for executing JavaScript code server-side, that can be allied with HTML and Cascading Style Sheets (CSS) to provide a intuitive design.

The Node.js server connects to the database referenced in the previous section. Depending on the page currently selected and the number of points required for the graph, the server retrieves the desired data with queries that use the compound indexes.

The data is then transferred to an HTML and CSS script that displays the data with the intended design. This HTML script is also responsible for the following features: In order to maintain a number of frames per second (FPM) that allows a smooth visualization of the graphs, the page refreshes once every minute; The display setting of the interface can be changed by the user between light and dark modes.

In Figure 4.5, the main steps of the data visualization are presented.

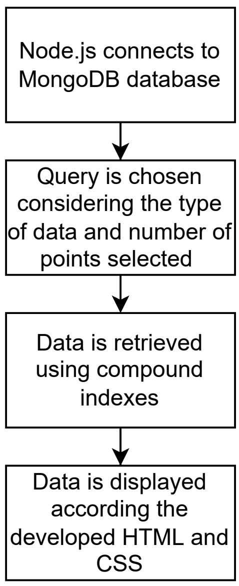


Figure 4.5: Main steps of the data visualization in the GUI.

When the GUI is deployed, the user is presented with an Home Page, shown in Figure 4.6, that consists of an ECG and Respiration graphs on the left and heart rate, oxygen saturation, respiration rate and body temperature values on the right. The WoW project logo is also visible adjacent to an image of battery that is updated according to the battery level of Biosticker.

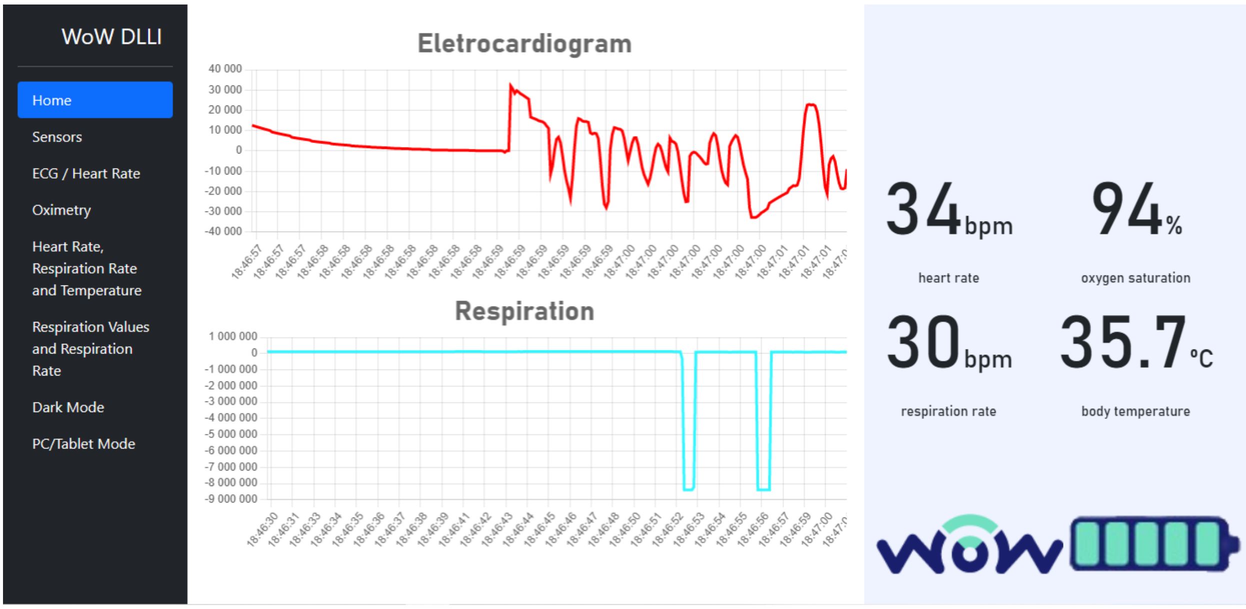


Figure 4.6: GUI home page in light mode.

The Sensors page allows the user to monitor a specific sensor. This page shows a list of the options of sensors. When the "View" option is selected, the interface is redirected to page with the chosen sensor graph. Through the sensors page, only one type of sensor can be selected to be visualized at each time. The user can select the number of points shown in the graphs and stop the auto-updating of the graphs. In Figure 4.7, the chosen sensor is the accelerometer, that presents 3 lines in the same graph representing each axis (x, y and z).

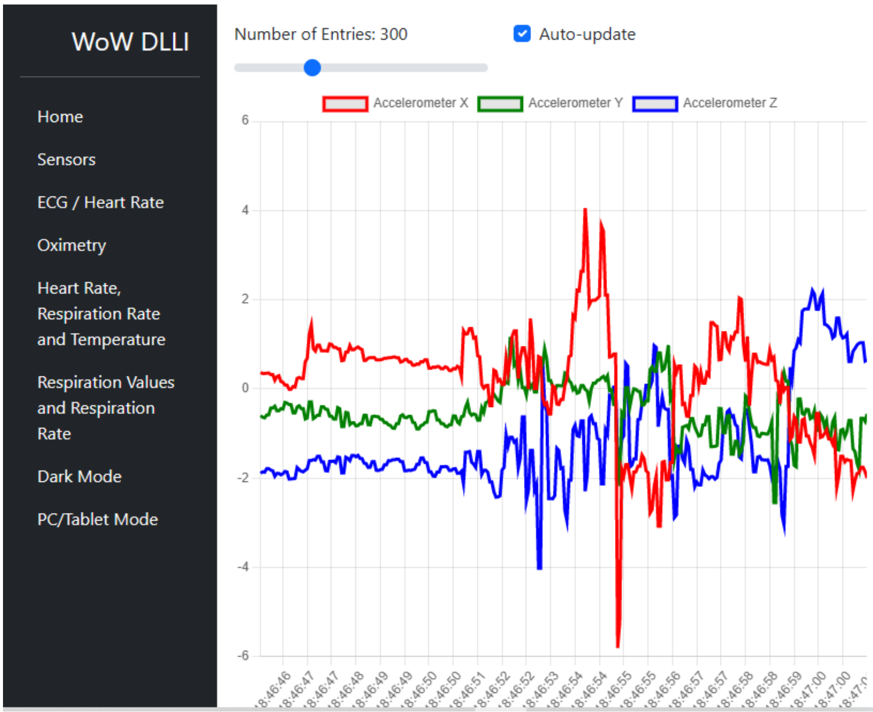


Figure 4.7: IMU graph.

In addition to the Home Page and Sensors page, the user can also select other 3 pages: one that contains a graph with values of Oxygen Saturation, a page that contains a graph with values of Respiration and the last calculated value of respiratory rate and a page with the last values of Heart Rate, Respiration Rate and Temperature.

As already mentioned, there is also a dark mode that is present in multiple figures in this dissertation's annex, such as in Figure A.1.

A video demonstration of the GUI can be found at `https://youtu.be/fNhaHLjvpIc`. In this video, the volunteer was seated talking casually, then taking deep breaths and, finally, sustaining his respiration.

4.7 Services Automation

To improve the usability of the Smart Box, its modules responsible for the data acquisition, data transmission through MQTT, data storage, preprocessing and data visualization were automated. This means that when the Raspberry Pi is turned on, its modules start to work without the intervention of the user, simplifying the job of healthcare professionals.

To achieve the plug and play feature, services in Ubuntu are used to run the codes responsible for the deployment of the GUI and for the BLE acquisition (that is also responsible for the start of the other Smart Box modules). Services are described as essential background processes that usually run while booting up and shut down with the Operating System (OS) [23].

5 Results and Discussion

5.1 Objectives of the Experiments

Experiments were conducted to validate the developed system. In order to assess its stability, the connection between Biosticker and Smart Box was studied in two perspectives. First, the number of failures in the connection and their duration are studied in Subsection 5.4.1. The goal of the tests is to prove that the rate of occurrence and duration of the failures are low. Then, the data rates explored in Subsection 5.4.2 intend to prove that all the data sent from the Biosticker is acquired by the Smart Box. These values reflect the behaviour of the system when the Biosticker and Smart Box are connected.

Using both results provided in Sections 5.4.1 and 5.4.2, the Biosticker-Smart Box connection is characterized and compared to non-wireless DAS in the state of art in order to assess if the proposed system implementation is a viable solution to overcome detachment problems in the state of the art systems.

Similar to the need of a reliable connection between Biosticker and Smart Box, the connection between Smart Box and Gateway must also be stable. The results of Subsection 5.4.4 have the goal to prove the stability in the data transmission and that there is no data loss in the MQTT communication.

One objective of the experiments is to prove that the chosen data storage software is capable of storing enough data to monitor a patient. The results in Subsection 5.4.5 accomplished that goal by studying the amount of data acquired and memory required to store it.

In order to assess if the Raspberry Pi is an adequate device to be used as a Smart Box, The Central Processing Unit (CPU) usage of the Smart Box is studied in Subsection 5.4.7. The CPU usage provides information about how intensively the running programs are being processed. The value represents what percentage of a processor core's total working time is actually being utilized to process data [35].

Regarding the GUI, the results in Subsection 5.4.6 have the aim to prove its adequacy by exploring its latency that provides information about the time needed to display the home page .

Some considerations regarding the distance between Smart Box and Biosticker and its communication range are also presented in this Chapter in Subsection 5.4.8, aiming to prove that the system provides its user ability to move within the hospital room.

5.2 Experimental Setup

As mentioned in Chapter 1, the work that resulted in this dissertation was tested in a pilot in CHUC from 15 to 17 September 2021, along with other work implemented by the WoW project team.

This overall system testing in a real scenario has the goal of gathering feedback from the users of the Biosticker and health professionals in order to implement it and improve clinical usability and economic value. These new implementations will be tested in future pilots.

From the point of view of this dissertation, the CHUC pilot provides the opportunity to gather data in a real scenario (hospital), assuring more credibility to the results presented in Section 5.4.

The CHUC pilot lasted for 3 days and there were 2 volunteers each day. During the experiment, a WoW Kit was assigned to each volunteer, containing a Biosticker, a Smart Box with fixed Internet Protocol (IP) address, a Bluetooth dongle, an oximeter and a tablet, as seen in Figure 5.1. A Gateway receives the data from all the used Smart Boxes.

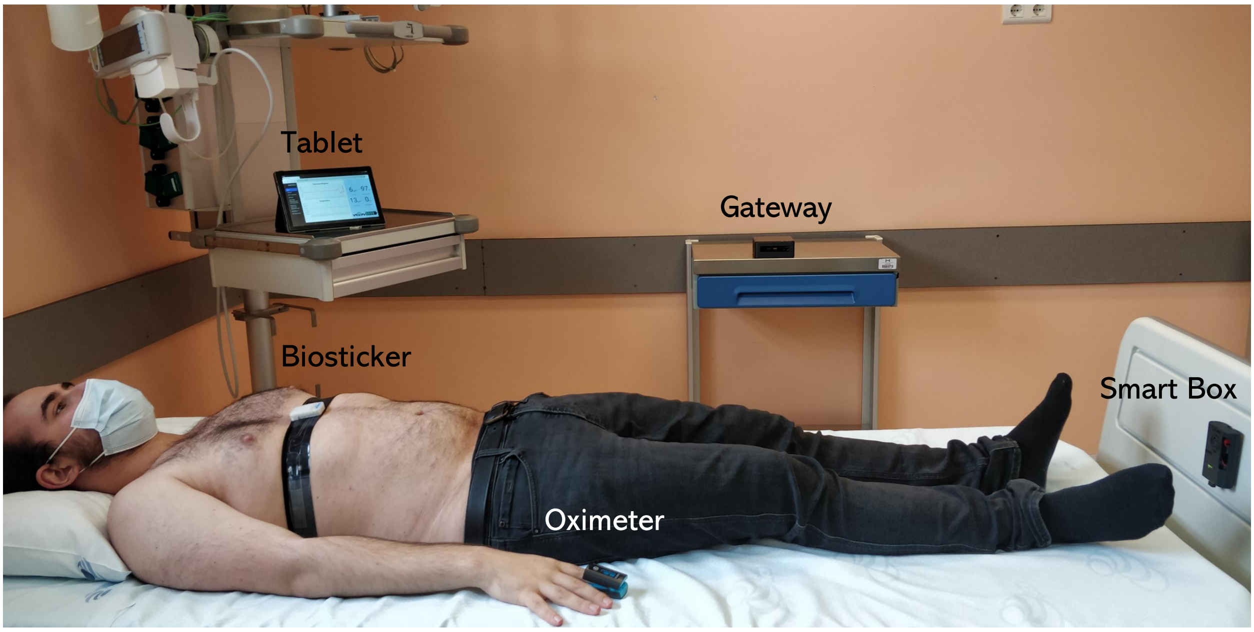


Figure 5.1: Volunteer with the WoW kit.

The chosen volunteers have an equal gender distribution and their information is present in Table 5.1. During the experiment, the volunteers simulated as closely as possible a hospital stay, being allowed different movements, positions and bathroom trips. In order to develop a posture classifier in the future, the posture and activities of the volunteers were registered, namely seating, walking, Fowler (standard patient position in which the patient is seated in a semi-sitting position with a bed angle of 45 to 60 degrees), Semi-Fowler (similar standard patient position to Fowler with a bed angle between 30 degrees and 45 degrees), standing and lying down.

Table 5.1: Information about Volunteers of WoW Project.

Volunteer	Gender (F/M)	Age (years)	Day	Used Kit	Duration
1	F	32	15/9	2	8 hours
2	M	27	15/9	4	8 hours
3	F	24	16/9	2	22 hours 40 minutes
4	M	23	16/9	4	8 hours
5	F	40	17/9	1	8 hours
6	M	22	17/9	3	8 hours

As presented in Table 5.1, one of the volunteers stayed for almost one day to better assess the durability and reliability of the system over a larger period of time, including sleep.

The vital signs heart rate, respiratory rate, pulse oximetry and temperature collected by the WoW System were registered, hourly, by a nurse. To validate these measurements, the volunteers were also monitored using the hospital monitorization system GE B105, that consists of 5 electrodes and a finger oximeter connected by wires to a monitor, as shown in Figure 5.2.

Even though only two kits were used at a time, four were prepared, so that two of those were ready to use as replacement in case of an fortuity. The kits were used alternately and its information can be found in Table 5.2. The only shared material between kits is the tablets, since there were only two available.

Table 5.2: CHUC Pilot Kits information.

Kit	Smart Box MAC Address	Smart Box Fixed IP	Smart Box UUID	Biosticker MAC Address	Oximeter MAC Address	Tablet
1	dc:a6:32:b6:18:57	172.30.5.22	c9453c02-97f0-4582-b499-0212a581a13a	DA:72:04:72:72:94	E8:E1:54:85:AA:61	VANKYO MATRIXPAD S20
2	dc:a6:32:c2:57:ca	172.30.5.23	c30c5f28-4a18-4520-82a0-40ffd741a893	E0:5A:A5:10:EE:58	BA:03:8C:05:5A:00	VANKYO MATRIXPAD S20
3	dc:a6:32:c2:57:97	172.30.5.27	cd55412d-9795-45b6-9d72-3b9d20fc0419	F7:B3:75:90:1E:04	BA:03:8C:0B:63:24	YESTEL MODFEL T5
4	dc:a6:32:c2:57:ac	172.30.5.28	2d598cb5-e600-42e8-9b48-6116f93630e4	F4:48:50:F4:DA:2E	D7:6C:DB:DF:60:A7	YESTEL MODFEL T5

Since reliable Wi-Fi connection is crucial to the WoW system as one of the communication standards used, the CHUC Information technology (IT) department provided an exclusive Wi-Fi network.

Figure 5.2: Volunteer monitorization using WoW and GE B105 systems.

5.3 Experimental Design

In order to calculate the connection failure duration, the timestamp of each sensor item in the database was retrieved and the difference between consecutive items is calculated.

When the difference is superior than 1 second, it is considered a connection failure. The number of disconnections and the amount of disconnection time of each hour of the experimental tests were calculated.

Using the duration of all the disconnections of the experimental tests, the average, median and standard deviation were calculated. This values intend to prove that the disconnections between Smart Box and Biosticker were brief and would not result in false alarms.

The BLE data acquisition rates presented in Subsection 5.4.2 were calculated by counting how many values of each sensor were acquired in each hour and dividing that value by the total connection time calculated in the previous subsection.

Similar to the calculations referent to each hour, the same results were calculated using time spams of 8 hours. The BLE data acquisition rates were then compared to the theoretical data rates of the Biosticker provided by the its firmware author.

In order to validate the adequacy of the connection between Smart Box and oximeters, the number of acquired values f oxygen saturation were counted. Dividing that value by the test duration (8 hours), the data rate is calculated.

The MQTT Data Transmission rates presented in Subsection 5.4.4 were calculated by assessing how many values of each sensor were received by the Gateway during the total connection time in a 8 hours test. The number of bytes of each message was registered and the average value of bytes per message of each type of sensor was calculated. The two values were multiplied to obtain the final value of data transmission rate in bytes per second.

The same algorithm was implemented using the MQTT messages sent by the Smart Box to obtain the expected value.

The comparison of these two values (rate at which MQTT rate is sent and received) and calculation of relative error intends to prove that the MQTT is an adequate protocol to be used in the Smart Box-Gateway communication.

To obtain the results presented in SubSection 5.4.5, the amount of instances of each sensor in the MongoDB of each kit was counted. These values provide a sense of scale of the experiments to the reader.

The memory storage used by the database is obtained from the previously mentioned tool mongo Shell. It intends to show that the chosen storage software MongoDB is capable of storing continuous data for several days, taking into account the mass storage limitations of the Smart Box. If large amounts of data do not require large memory to store it, then the storage software is adequate to be used in patient's data monitorization.

The value presented is an average of values of frames per second (FPS). Each value of FPS represents the number of times the GUI screen is displayed with new data per second. The higher the FPS value, the more fluid the graphs will appear to the human eye. Furthermore, high values of FPS means lower latency, since, in this case, the GUI latency can be described as the amount of time that a page of the GUI needs to be fully displayed.

The CPU usage values presented in Subsection 5.4.7 were obtained using the`htop` tool, that provides a continuously updated list of all actively running processes and their CPU values and memory usage. In the tests, 4 values per second were registered. High CPU usage values suggest that the Raspberry Pi does not have the computational power to complete the necessary tasks and they can cause freezes in the system.

5.4 Experimental Results

5.4.1 Analysis of Connection Failures

The average duration of a connection failure is 7.81 $\pm$ 40.33 seconds and the median value is 1.2135 seconds. The average is significantly superior to the median and the standard deviation is significantly superior to the average, implying that both values are largely affected by disconnections of high duration. In cases such as this one, the median is considered a more representative value of the population.

In total, there were 2816 connection failures during the experimental tests. Considering that the total time of experiments was 62 hours and 40 minutes, there was a connection failure every 80 seconds, on average. The overall time of disconnection between Biosticker and Smart Box was 22016 seconds (approximately 6 hours and 7 minutes), that corresponds to 9.75% of the duration of the tests.

As referred in Chapter 2, one of the gaps in some studies in the state of the art is that their systems use wires to obtain data acquisition, that if applied to healthcare could cause disconnection due to the patient's movement. These results show that the proposed system, in its current state, can not be considered an improvement upon the state of the art, since, even though the median of time of disconnections is low, meaning that they mostly would not cause false alarms that require help, its occurrence is too frequent compared with the expected rate of movement caused disconnections in a non-wireless DAS.

The percentage of disconnection time is not adequate to be used in a healthcare system, since data loss as high as 9.75% can prevent help to arrive to the patient.

There are multiple reasons that cause the dissatisfying values. The longer connections failures were caused by moments of adjustments of the placement of the Biosticker, replacement of their batteries and when the distance between Smart Box and Biosticker exceeded the range of the BLE communication (for example, when the volunteers went to the bathroom). As previously mentioned, these type of disconnections explain the high value of standard deviation, the high difference between the mean and median and the overall magnitude of these values.

The firmware of the Biostickers, whose implementation is not a responsibility of the author of this dissertation, includes random breaks in the transmission of the data that result in short and intermediate duration failures in the connection with the Smart Box. These sudden breaks and disconnections greatly affect the results in this subsection by increasing the number of disconnections.

As stated in Section 5.2, the positions and activities of the volunteers were monitored. Taking this information into account, it is visible that when the volunteers were sleeping the disconnections were considerably less frequent.

This occurrence indicates that a higher movement of the patients results in instability in the connection between Biosticker and Smart Box. Considering that the Smart Box does not move and the patients movement only affects the Biosticker performance, these results suggest that the Biosticker firmware should be improved in order to make the disconnections less frequent.

5.4.2 BLE Data Acquisition Rates

The Figures 5.3, 5.4 and 5.5 represent the hourly data rates of each sensor in 8 hour acquisition periods.

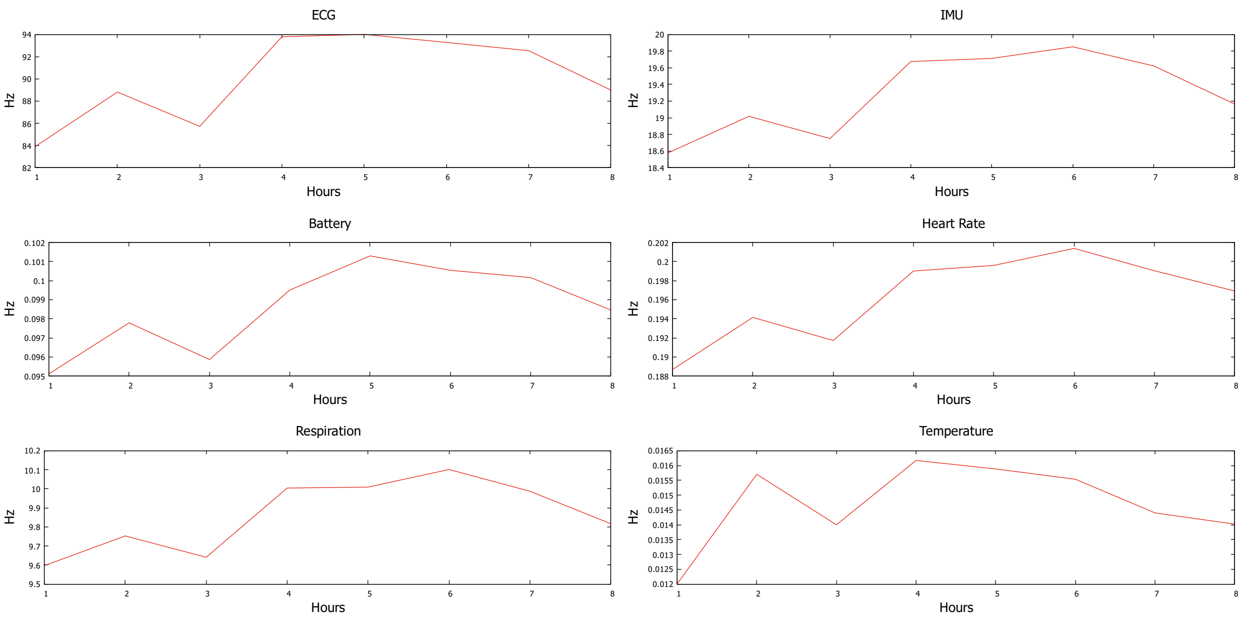


Figure 5.3: Hourly Data Rates from day 1 (8 hours of acquisition).

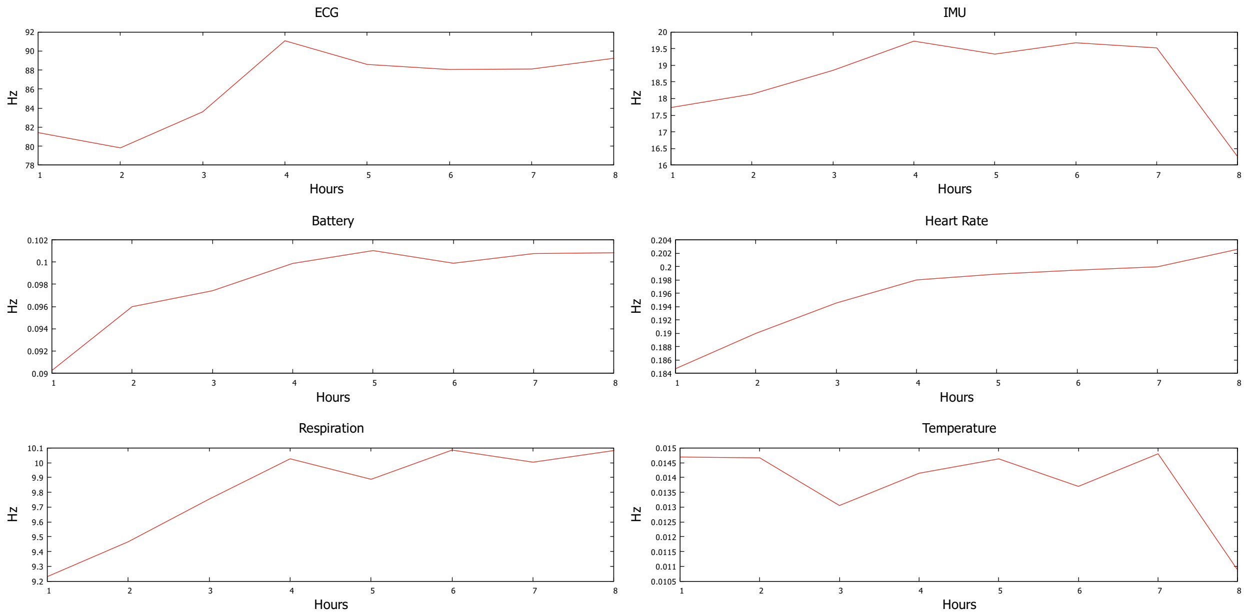


Figure 5.4: Hourly Data Rates from day 2 (8 hours of acquisition).

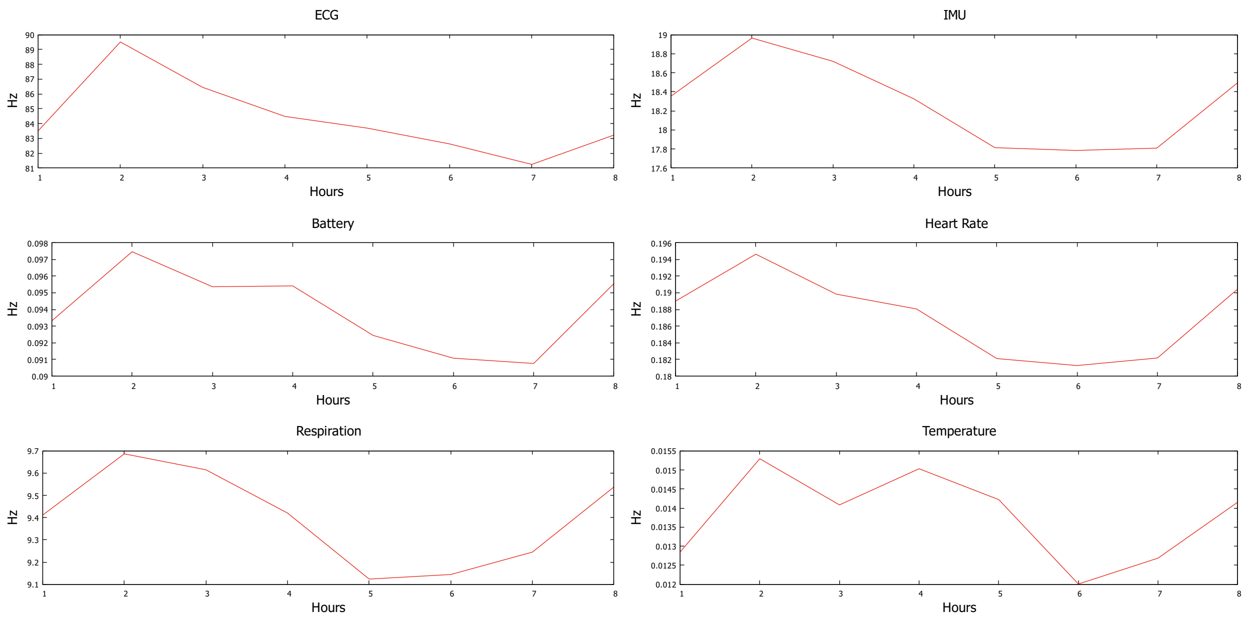


Figure 5.5: Hourly Data Rates from day 3 (8 hours of acquisition).

In the graphs, it is visible that the evolution of data rates of the multiple sensors is similar, which sustains the conclusion that when the connection between Biosticker and Smart Box is unstable the data acquisition of all types of sensor is equally affected.

In Tables 5.4 and 5.3, it is shown that data loss regarding a superior duration of time (8 and 22:40 hours) did not reach 17.5%.

The sensors that have the higher relative errors are ECG, IMU and temperature. During the experimental tests, there were multiple problems non-related to this dissertation with the temperature signal since it was interfering with the ECG signal, which explains their worst performance

Table 5.3: Data Rates in 8 hours tests.

Signals	Calculated Data Rate (Hz)	Expected Data Rate (Hz)	Relative Error (%)
ECG	86.35	100	13.65
ECG	90.05	100	9.95
ECG	86.30	100	13.70
ECG	90.22	100	9.78
ECG	84.40	100	15.60
IMU	18.24	20	8.79
IMU	19.29	20	3.54
IMU	18.63	20	6.87
IMU	18.77	20	6.13
IMU	18.29	20	8.57
Battery	0.093	0.1	7.03
Battery	0.099	0.1	1.42
Battery	0.098	0.1	1.72
Battery	0.096	0.1	3.71
Battery	0.094	0.1	6.05
Heart Rate	0.186	0.2	7.12
Heart Rate	0.196	0.2	1.86
Heart Rate	0.196	0.2	1.95
Heart Rate	0.193	0.2	3.65
Heart Rate	0.187	0.2	6.40
Respiration	9.33	10	6.67
Respiration	9.86	10	1.38
Respiration	9.83	10	1.74
Respiration	9.64	10	3.59
Respiration	9.40	10	6.01
Temperature	0.0145	0.0166	12.95
Temperature	0.0147	0.0166	11.70
Temperature	0.0138	0.0166	17.14
Temperature	0.0138	0.0166	16.85

in the acquisition. Furthermore, the lower performance of IMU and ECG can be caused by their higher data rates. Since the threshold for the difference between two values acquired is 1 second to be considered a disconnection, it is likely that smaller disconnections affect more these sensors with higher data rates.

Another reason for the non-perfect data rates is the interference between the multiple communications occurring simultaneously.

Comparing the two Tables, it is visible that the results obtained with the 22 hours and 40 minuets test were superior to those from the 8 hours test. This analysis proves that the longer duration of acquisition does not result in a worst performance regarding the data acquisition rates.

These values demonstrate that the developed system is capable of acquiring many types of sensors simultaneously, which is one of the gaps in some studies in the state of the art, as referred

Table 5.4: Data Rates in 22 hours and 40 minutes tests.

Signals	Calculated Data Rate (Hz)	Expected Data Rate (Hz)	Relative Error (%)
ECG	97.08	100	2.92
IMU	19.38	20	3.11
Battery	0.099	0.1	1.00
Heart Rate	0.198	0.2	0.92
Respiration	9.93	10	0.66
Temperature	0.01340	0.0166	15.67

in Chapter 2.

5.4.3 Analysis of the Smart Box-Oximeter connection

The data acquisition rates of Oxygen Saturation are present in Table 5.5

Table 5.5: Data Acquisition Rates of Oxygen Saturation in 8 hour duration tests.

Used Oximeter	Calculated Data Rate (Hz)	Expected Data Rate (Hz)	Relative Error (%)
Wellue FS20F	0.795	1.2	33.76
PC-60FW	0,034	1.2	97.18
PC-60FW	0,079	1.2	93.45
PC-60FW	0,003	1.2	99.75
Wellue FS20F	0,214	1.2	82,14

Regarding the data acquisition from the oximeters, the connection was not stable, due to a failure in the reconnecting moments that resulted in high disconnection periods. These high high disconnection periods lead to a substantial difference between the calculated daa rates and the expected data rates.

5.4.4 MQTT Data Transmission Rates

Considering that the calculated data transmission rate refers to the sent MQTT messages by the Smart Box and the expected data transmission rate refers to the received MQTTT by the Gateway, as explained in Section 5.3, the results in Table 5.6 show that the data loss in the connection between the Smart Box and Gateway was below 3%. These results prove that MQTT is an adequate protocol to be used in said communication.

Table 5.6: Data Transmission Rates in 22 hours and 40 minutes test.

Signals	Calculated Data Transmission Rate (bytes/second)	Expected Data Transmission Rate (bytes/second)	Relative Error (%)
ECG	1517.92	1497.7	1.35
IMU	6143.2	6289.33	2.32
Heart Rate	27.4246	28.0759	2.32
Respiration	45.0943	46.2784	2.56
Temperature	2.45906	2.52284	2.53

5.4.5 Data Volume

Table 5.7 provides the amount of values of each sensor in each kit. The kit 2 was the most used, containing values regarding 30 hours and 40 minutes of acquisition. This amount of data resulted in a MongoDB database with memory usage equal to 0,864 GB.

Considering that the Smart Box has a 32 GB SD Card, as referred in Table 3.1, it is proven that the MongoDB is adequate to use as the data storage software for patient monitoring, since, theoretically, it is capable of storing data during 47 days before reaching the SD Card limit.

Furthermore, the retrieval of the data values necessary for the home page of the GUI takes around 2 ms to complete. This value is extremely low and it is not an impediment to the constant data retrievals needed for the GUI.

5.4.6 GUI Latency

In the last day of the experiments, the latency of the GUI was tested to assess its adequacy.

The average FPS value was 3.96 ± 0.21 FPS and its median is 4.0 FPS.

These values are low due to an inadequate choice of version of the library `Chart.js` [10] to display the ECG and Respiration graphs (cf. Figure 4.6), since it demands to much time to do so, which lowers the FPS value.

Table 5.7: Data Volume.

Signals	Number of acquired instances Kit 1	Number of acquired instances Kit 2	Number of acquired instances Kit 3	Number of acquired instances Kit 4	Total all Kits
ECG	2293517	4777710	2176657	4312645	13560529
IMU	2536268	5901768	2829780	5563544	16831360
Battery	2448	5063	2423	4813	14747
Heart Rate	4899	10104	4828	9593	29424
Oxygen Saturation	34	29696	5501	2684	37915
Pulse Rate	75	30127	5527	2731	38460
Respiration	245095	508079	242410	481321	1476905
Temperature	0	755	356	695	1806

With a different version of the library `Chart.js` (2.9.4 instead of 3.2.0), the GUI values of FPS reached around 20, which makes graphs in the GUI appear fluid.

5.4.7 CPU Usage

In a 8 hours test, the average value of CPU usage is 39.77 $\pm$ 15.68% and its median is 44.70%.

The high standard deviation results from the peaks of CPU usage at the moments of reconnection, search for BLE devices and initialization of the GUI.

These values are adequate and do not indicate that the CPU usage might became excessive, proving that the Raspberry Pi 4 has the needed computing power to be used as a Smart Box and fulfill its purposes.

In a 22 hours and 40 minutes test, the average value of CPU usage is 33.32 $\pm$ 14.80% and its median is 29.5%.

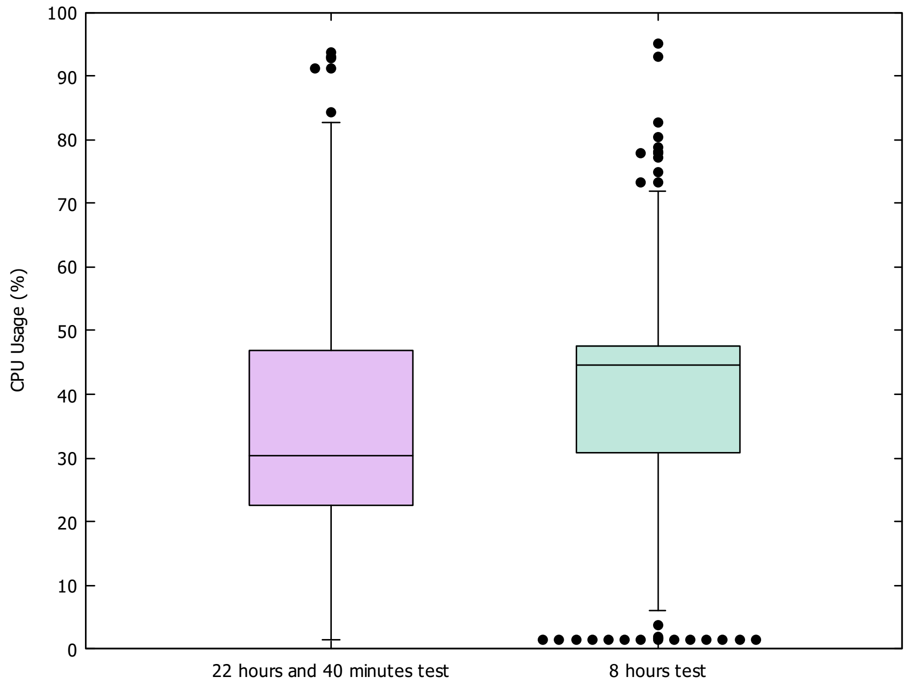


Figure 5.6: Boxplot representing the CPU usage values in 8 hours test and 22 hours and 40 minutes test.

Comparing the two boxplot graphs in Figure 5.6, it is visible that the CPU usage values during the 22 hours and 40 minuets test were, in general, lower than the ones from the 8 hours test. The lower movement of the volunteer, considering that he/she slept through approximately 8 hours of the test, can explain the observed lower values, since with a more stable Biosticker-Smart Box connection decreases the number of peaks of CPU usage at the moments of reconnection. This analysis sustains the conclusion that the longer duration of acquisition does not result in a worst performance regarding the GUI.

5.4.8 Communication Range

During the experiments, it was noticed that there were more disconnections between Biosticker and Smart Box when the distance is greater than 2,5 meters.

This phenomenon can be explained by the non-perfect conditions such as the movement of the patients and the various simultaneous communications that result in interference, damaging each connection and its range.

6 Conclusion

In this dissertation, a DAS was proposed to be used in an IoT node of a healthcare system. Comparing it to similar cases in the state of the art, the proposed system implementation has the advantage of not requiring any wires to collect the data that restrict the movement of its users. Furthermore, for each patient, the system is able to acquire, preprocess and transmit data referent to multiple types of sensors, which is not a functionality of many systems in the literature. Additionally, a GUI is deployed to allow both patients and healthcare professionals to observe the data in real time, facilitating the analysis of vital data.

Overall, the developed system and implemented methodology were adequate to achieve the proposed goals and to be used in a healthcare monitoring system. Data acquisition was achieved wirelessly with unsatisfying results regarding its number of disconnections. Even though most disconnections with the Biosticker have short duration, the percentage time of disconnection time is too high to be used in a healthcare system whose failure can be fatal in life or death situations that require immediate attention by emergency teams. Reliable data transmission was achieved, assuring that the acquired data can be used in higher level modules and the chosen data storage software was adequate. Data visualization was achieved through a GUI. Comparing it with the ones present in the state of art, the developed GUI has a more appealing design that presents every vital sign in its home page but needs improvement to lower its latency.

6.1 Future Work

In order to improve the stability of the connection between Biosticker and Smart Box and minimize the duration and the number of occurrence of disconnections, both firmwares of Smart Box and Biosticker should be revised.

To improve the reliability of the Oximeter-Smart Box BLE connection the reconnection method must be re-implemented. The combination of both acquisition services into one can be a solution to solve this problem.

In the future, some functionalities can be added to the Smart Box-Gateway communication.

The broker should reserve a topic which is used to request the status of every Smart Box currently connected to it. This status reflects if the Smart Box and Biosticker are connected. Furher security measures must also be implemented to prevent cybersecurity attacks.

Regarding the GUI, its latency can be improved in future implementations by using a library that displays the necessary graphs faster without compromising its design.

Regarding the design of the GUI, the usage of the same acronym for both heart and respiration rate units of measurement raised some confusion in the analysis of the GUI, since "bpm" can either mean breaths per minute and beats per minute. In the future, it is advised to change to another unit of measurement of respiration also used by healthcare professionals, such as cycles per minute and its acronym "cpm". Another information that can appear in the home page of the GUI is the state of Biosticker-Smart Box and oximeter-Smart Box connections, so two icons should be added to reflect it. Also, the button that interchanges between dark and light mode should be placed in a more intuitive location.

Future implementation of data classification algorithms such as IMU patient state classification to determine its posture using IMU data is beneficial to detect patient's falls and alert healthcare professional to provide help.

Finally, it was noticed that the respiration values were not reliable when the user of the Biosticker made sharp movements, since those movements also cause strain in the respiration sensor. So, a classifier that uses the IMU data to assess if the respiration values are valid should be implemented in the future.

Bibliography

[1] C. Ajchariyavanich, T. Limpisthira, N. Chanjarasvichai, T. Jareonwatanan, W. Phongphanpanya, S. Wareechuensuk, S. Srichareonkul, S. Tachatanitanont, C. Ratanamahatana, N. Prompoon, and M. Pipattanasomporn. Park king: An iot-based smart parking system. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 729–734, 2019.

[2] Kazi Masudul Alam, Mukesh Saini, and Abdulmotaleb El Saddik. Toward social internet of vehicles: Concept, architecture, and applications. *IEEE Access*, 3:343–357, 2015.

[3] C. Arcadius Tokognon, Bin Gao, Gui Yun Tian, and Yan Yan. Structural health monitoring framework based on internet of things: A survey. *IEEE Internet of Things Journal*, 4(3):619–635, 2017.

[4] Asus. Bluetooth 5.0 usb adapter. `https://www.asus.com/Networking-IoT-Servers/Adapters/All-series/USB-BT500/`, 2021.

[5] Rachmad Atmoko, Rona Riantini, and M Hasin. Iot real time data acquisition using mqtt protocol. *Journal of Physics: Conference Series*, 853:012003, 05 2017.

[6] S. B. Baker, W. Xiang, and I. Atkinson. Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access*, 5:26521–26544, 2017.

[7] Bleak. Bleak. `https://bleak.readthedocs.io/en/latest/`, 2020.

[8] Kang-Ming Chang, Shin-Hong Liu, and Xuan-Han Wu. A wireless semg recording system and its application to muscle fatigue detection. *Sensors (Basel, Switzerland)*, 12:489–99, 12 2012.

[9] Kuor-Hsin Chang. Bluetooth: a viable solution for iot? [industry perspectives]. *IEEE Wireless Communications*, 21(6):6–7, 2014.

[10] Chart.js. Chart.js. `https://www.chartjs.org/`, 2022.

[11] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar. Craiot: Concept, review and application(s) of iot. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–4, 2019.

[12] Ho Ting Cheng and Weihua Zhuang. Bluetooth-enabled in-home patient monitoring system: Early detection of alzheimer's disease. *IEEE Wireless Communications*, 17(1):74–79, 2010.

[13] Hongmei Chi, Temilola Aderibigbe, and Bobby C. Granville. A framework for iot data acquisition and forensics analysis. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5142–5146, 2018.

[14] Neramitr Chirakanphaisarn, Thadsanee Thongkanluang, and Yuwathida Chiwpreechar. Heart rate measurement and electrical pulse signal analysis for subjects span of 20–80 years. In *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 70–74, 2016.

[15] Mike X Cohen. Welch's method for smooth spectral decomposition. `https://www.youtube.com/watch?v=YK1F0-3VvQI`, 2020.

[16] Mario Collotta and Giovanni Pau. A novel energy management approach for smart homes using bluetooth low energy. *IEEE Journal on Selected Areas in Communications*, 33(12):2988–2996, 2015.

[17] Hesham El-Sayed and Gokulnath Thandavarayan. Congestion detection and propagation in urban areas using histogram models. *IEEE Internet of Things Journal*, 5(5):3672–3682, 2018.

[18] Mouser Electronics. Maxim integrated max30205 human body temperature sensor. `https://pt.mouser.com/new/maxim-integrated/maxim-max30205-sensor`, 2021.

[19] EBV Elektronik. Maxim integrated max30205 human body temperature sensor. `https://www.avnet.com/wps/portal/ebv/products/new-products/npi/2017/maxim-max30205`, 2021.

[20] Symmetry Eletronics. Classic bluetooth vs bluetooth low energy - a round by round battle. `https://www.symmetryelectronics.com/blog/classic-bluetooth-vs-bluetooth-low-energy-a-round-by-round-battle/`, 2014.

[21] Katalin Ferencz and József Domokos. Iot sensor data acquisition and storage system using raspberry pi and apache cassandra. In *2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, pages 000143–000146, 2018.

[22] Food and Drug Administration. What is digital health? `https://www.fda.gov/medical-devices/digital-health-center-excellence/what-digital-health`, 2020.

[23] It's FOSS. How to start, stop restart services in ubuntu and other linux distributions. `https://itsfoss.com/start-stop-restart-services-linux/`, 2020.

[24] Python Software Foundation. asyncio — asynchronous i/o. `https://bleak.readthedocs.io/en/latest/`, 2022.

[25] Python Software Foundation. Event loop. `https://docs.python.org/3/library/asyncio-eventloop.html`, 2022.

[26] Python Software Foundation. paho-mqtt 1.6.1. `https://pypi.org/project/paho-mqtt/`, 2022.

[27] Python Software Foundation. pymongo 4.0.1. `https://pypi.org/project/pymongo/`, 2022.

[28] K. R. Fowler. The future of sensors and sensor networks survey results projecting the next 5 years. In *2009 IEEE Sensors Applications Symposium*, pages 1–6, 2009.

[29] Emmanouil Georgakakis, Stefanos Nikolidakis, Dimitrios Vergados, and Christos Douligeris. An analysis of bluetooth, zigbee and bluetooth low energy and their use in wbans. volume 55, pages 168–175, 10 2010.

[30] Lena Gohlke, Frederik Dreyer, Monica Pimiento Álvarez, and Jens Anders. An iot based low-cost heart rate measurement system employing ppg sensors. In *2020 IEEE SENSORS*, pages 1–4, 2020.

[31] Sarthak Gupta, Virain Malhotra, and Vasudha Vashisht. Water irrigation and flood prevention using iot. In *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 260–265, 2020.

[32] CMI Health. Fingertip pulse oximeter | pc-60fw. `https://www.cmihealth.com/products/fingertip-pulse-oximeter-pc-60fw`, 2021.

[33] Alfin Hidayat, Subono, Vivien Arief Wardhany, Ajie Setyo Nugroho, Sofyan Hakim, Mirtha Jhoswanda, Ika Noer Syamsiana, and Nur Anis Agustina. Designing iot-based independent pulse oximetry kit as an early detection tool for covid-19 symptoms. In *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, pages 443–448, 2020.

[34] intel. Intel nuc kits. `https://www.intel.com/content/www/us/en/products/details/nuc/kits.html`, 2021.

[35] Digital Guide IONOS. High cpu usage: What does this mean? `https://www.ionos.com/digitalguide/server/know-how/cpu-usage/`, 2020.

[36] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, 2015.

[37] Nobuhiro Iwamatsu. gatttool - tool for bluetooth low energy device. `http://manpages.ubuntu.com/manpages/bionic/man1/gatttool.1.html`, 2015.

[38] Antonio J. Jara, Francisco J. Belchi, Alberto F. Alcolea, José Santa, Miguel A. Zamora-Izquierdo, and Antonio F. Gómez-Skarmeta. A pharmaceutical intelligent information system to detect allergies and adverse drugs reactions based on internet of things. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 809–812, 2010.

[39] Nobuhiro Kajikawa, Yuya Minami, Eitaro Kohno, and Yoshiaki Kakuda. On availability and energy consumption of the fast connection establishment method by using bluetooth classic and bluetooth low energy. In *2016 Fourth International Symposium on Computing and Networking (CANDAR)*, pages 286–290, 2016.

[40] Arshiya Khan and Ajitkumar Khachane. Survey on iot in waste management system. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, pages 27–29, 2018.

[41] Shivanjali Khare and Michael Totaro. Big data in iot. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7, 2019.

[42] Ravi Kishore Kodali, Govinda Swamy, and Boppana Lakshmi. An implementation of iot for healthcare. In *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 411–416, 2015.

[43] Siddharth A. Kokalki, Akshay R. Mali, Pawan A. Mundada, and Ritesh H. Sontakke. Smart health band using iot. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pages 1683–1687, 2017.

[44] Mwiza Kumwenda. How to manage bluetooth devices on linux using bluetoothctl. `https://www.makeuseof.com/manage-bluetooth-linux-with-bluetoothctl/`, 2021.

[45] Seungmin Lee, Seungyoon Nam, and Hyunsoon Shin. The analysis of sleep stages with motion and heart rate signals from a handheld wearable device. In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1135–1137, 2016.

[46] life.augmented. inemo 6dof inertial measurement unit (imu), for consumer electronics. `https://www.st.com/en/mems-and-sensors/lsm6ds3.html`, 2021.

[47] Hasmah Mansor, Muhammad Helmy Abdul Shukor, Siti Sarah Meskam, Nur Quraisyia Aqilah Mohd Rusli, and Nasiha Sakinah Zamery. Body temperature measurement for remote health monitoring system. In *2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, pages 1–5, 2013.

[48] maxim intgrated. Max30003. `https://www.maximintegrated.com/en/products/analog/data-converters/analog-front-end-ics/MAX30003.html`, 2021.

[49] Marcel Holtmann Maxim Krasnyansky and Fabrizio Gennari. hcitool(1) - linux man page. `https://linux.die.net/man/1/hcitool`, 2021.

[50] Lepu Medical. Pc-60fw fingertip pulse oximeter. `https://en.lepumedical.com/products/pc-60f-fingertip-oximeter/`, 2021.

[51] MongoDB. Compound indexes. `https://docs.mongodb.com/manual/core/index-compound/`, 2021.

[52] MongoDB. Mongodb. `https://www.mongodb.com/pt-br`, 2021.

[53] Dhanurdhar Murali, Deepthi R Rao, Swathi R Rao, and M Ananda. Pulse oximetry and iot based cardiac monitoring integrated alert system. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2237–2243, 2018.

[54] Ted Nachazel. What is a strain gauge and how does it work? `https://www.michsci.com/what-is-a-strain-gauge/`, 2020.

[55] Masayuki Nakamura, Jiro Nakamura, Guillaume Lopez, Masaki Shuzo, and Ichiro Yamada. Collaborative processing of wearable and ambient sensor system for blood pressure monitoring. *Sensors (Basel, Switzerland)*, 11:6760–70, 12 2011.

[56] B. Nath, F. Reynolds, and R. Want. Rfid technology and applications. *IEEE Pervasive Computing*, 5(1):22–24, 2006.

[57] Ilya Nikolaevskiy, Dmitry Korzun, and Andrei Gurtov. Security for medical sensor networks in mobile health systems. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–6, 2014.

[58] Saowakhon Nookhao, Vipa Thananant, and Thanakorn Khunkhao. Development of iot heartbeat and body temperature monitoring system for community health volunteer. In *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON)*, pages 106–109, 2020.

[59] Alex Page, Shurouq Hijazi, Dogan Askan, Burak Kantarci, and Tolga Soyata. Research directions in cloud-based decision support systems for health monitoring using internet-of-things driven data acquisition. *International Journal of Services Computing*, 4:18–34, 04 2016.

[60] Maria Rita Palattella, Mischa Dohler, Alfredo Grieco, Gianluca Rizzo, Johan Torsner, Thomas Engel, and Latif Ladid. Internet of things in the 5g era: Enablers, architecture, and business models. *IEEE Journal on Selected Areas in Communications*, 34(3):510–527, 2016.

[61] Zhibo Pang. Technologies and architectures of the internet-of-things (iot) for health and well-being. 2013.

[62] Iuliu Alexandru Pap, Stefan Oniga, Ioan Orha, and Alexandru Alexan. Iot-based ehealth data acquisition system. In *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pages 1–5, 2018.

[63] Emanuele Portuese, Silvia Buscaglione, Domenico Formica, and Davide Lanaro. Assessment of running training sessions using imu sensors: evaluation of existing parameters and choice

of new indicators. In *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT*, pages 121–124, 2020.

[64] K. Premkumar, K. Gokila Devi, B. Jamuna, and G. Priyadharshini. Prototype design for energy consumption in smart home using iot. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–6, 2019.

[65] Abdul Rehman, Nazar Abbas Saqib, Syed Muhammad Danial, and Syed Hassaan Ahmed. Ecg based authentication for remote patient monitoring in iot by wavelets and template matching. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 91–94, 2017.

[66] G M Siddesh, S Manjunath, and K G Srinivasa. Application for assisting mobility for the visually impaired using iot infrastructure. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pages 1244–1249, 2016.

[67] Sai Sreekar Siddula, Phaneendra Babu, and P. C. Jain. Water level monitoring and management of dams using iot. In *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–5, 2018.

[68] Mandeep Singh and Neelu Jain. Performance and evaluation of smartphone based wireless blood pressure monitoring system using bluetooth. *IEEE Sensors Journal*, 16(23):8322–8328, 2016.

[69] Grant Maloy Smith. What is data acquisition (daq or das)? the ultimate guide. `https://dewesoft.com/daq/what-is-data-acquisition#daq-system-components`, 2020.

[70] Elisa Spanò, Stefano Di Pascoli, and Giuseppe Iannaccone. Low-power wearable ecg monitoring system for multiple-patient remote monitoring. *IEEE Sensors Journal*, 16(13):5452–5462, 2016.

[71] Yupin Suppakhun. Flood surveillance and alert system an advance the iot. In *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pages 325–328, 2019.

[72] Lu Tan and Neng Wang. Future internet: The internet of things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, volume 5, pages V5–376–V5–380, 2010.

[73] Song Tan, Debraj De, Wen-Zhan Song, Junjie Yang, and Sajal Das. Survey of security advances in smart grid: A data driven approach. *IEEE Communications Surveys Tutorials*, PP:1–1, 10 2016.

[74] Liane Margarida Rockenbach Tarouco, Leandro Márcio Bertholdo, Lisandro Zambenedetti Granville, Lucas Mendes Ribeiro Arbiza, Felipe Carbone, Marcelo Marotta, and José Jair Cardoso de Santanna. Internet of things in healthcare: Interoperatibility and security issues. In *2012 IEEE International Conference on Communications (ICC)*, pages 6121–6125, 2012.

[75] Daniel Taub, Elmer Lupton, Roderick Hinman, Steven Leeb, John Zeisel, and Susan Blackler. The escort system: A safety monitor for people living with alzheimer's disease. *IEEE Pervasive Computing*, 10(2):68–77, 2011.

[76] Kinh Tran, Tu Le, and Tien Dinh. A high-accuracy step counting algorithm for iphones using accelerometer. In *2012 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 000213–000217, 2012.

[77] Alexander Vo, G Brooks, Md Ee, Ralph Farr, and Ben Raimer. Benefits of telemedicine in remote communities use of mobile and wireless platforms in healthcare. 01 2011.

[78] Huy Q. Vu, T. N. Kien Vu, and Dung X. Le. Research and development of a human step-counting and light-sleep wake-up device for health improvement. In *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, pages 296–299, 2018.

[79] Rolf H. Weber. Internet of things – new security and privacy challenges. *Computer Law Security Review*, 26(1):23–30, 2010.

[80] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[81] Wellue. Fs20f bluetooth finger oximeter. `https://getwellue.com/pages/fs20f-bluetooth-finger-oximeter`, 2021.

[82] Shanika Wickramasinghe. Compound indexes. `https://docs.mongodb.com/manual/core/index-compound/`, 2020.

[83] WoW. Wow deliverable d1.2: "functional specification and system integration architecture", howpublished = "`https://inovglintt.com/financiamento/wow/`", year = 2020.

[84] Tina Wu and Andrew Martin. Bluetooth low energy used for memory acquisition from smart health care devices. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1256–1261, 2018.

[85] Guangyu Xu. Iot-assisted ecg monitoring framework with secure data transmission for health care applications. *IEEE Access*, 8:74586–74594, 2020.

[86] Yuehong YIN. The internet of things in healthcare: An overview. *Journal of Industrial Information Integration*, 1, 03 2016.

[87] Guigang Zhang, Chao Li, Yong Zhang, Chunxiao Xing, and Jijiang Yang. Semanmedical: A kind of semantic medical monitoring system model based on the iot sensors. In *2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 238–243, 2012.

Appendix A

Annex

Table A.1: Biosticker Device Information Characteristics.

Characteristic	Handle	UUID	Property	Value
Software Revision String	52	00002a28-0000-1000-8000-00805f9b34fb	read	soft-SIM
Firmware Revision String	50	00002a26-0000-1000-8000-00805f9b34fb	read	fw-SPM_2.05S
Hardware Revision String	48	00002a27-0000-1000-8000-00805f9b34fb	read	hw-rev3B
Serial Number String	46	00002a25-0000-1000-8000-00805f9b34fb	read	SN1
Model Number String	44	00002a24-0000-1000-8000-00805f9b34fb	read	Model1
Manufacturer Name String	42	00002a29-0000-1000-8000-00805f9b34fb	read	WoW_Biosticker_ISR

Table A.2: PC-60FW Generic Access Characteristics.

Characteristic	UUID	Property	Value
Device Name	0x2A00	read, write	PC-60F_SN834070
Appearance	0x2A01	read	Unknown
Peripheral Preferred Connection Parameters	0x2A04	read	Connection Interval: 40.00 ms - 50.ms Slave Latency: 0 Supervision Timeout Multiplier
Central Address Resolution	0x2AA6	read, notify	Address resolution supported

Table A.3: PC-60FW Nordic UART Service Characteristics.

Characteristic	UUID	Property
RX Characteristic	6e400002-b5a3-f393-e0a9-e50e24dcca9e	write, write no response
TX Characteristic	6e400003-b5a3-f393-e0a9-e50e24dcca9e	notify

??

Table A.4: Wellue FS20F GATT Primary Services.

Service	UUID
Generic Access	0x1800
Generic Attribute	0x1801
Device Information	0x180A
Unknown Service	0xFFE0
Unknown Service	0xFFE5
Unknown Service	0xFFC0
Unknown Service	0xFF90
Unknown Service	0xFD00

Table A.5: Wellue FS20F Generic Access Characteristics.

Characteristic	UUID	Property	Value
Device Name	0x2A00	read	VTM 20F
Appearance	0x2A01	read	Unknown

Table A.6: Wellue FS20F Generic Attribute Characteristic.

Characteristic	UUID	Property	Value
Service Changed	0x2A05	indicate	Notifications and indications disabled

Table A.7: Wellue FS20F Device Information Characteristics.

Characteristic	UUID	Property
Unknown Characteristic	00490220-0147-7070-4121-01d002290078	read
Firmware Revision String	0x2A26	read
Hardware Revision String	0x2A27	read
Manufacturer Name String	0x2A29	read

Table A.8: Wellue FS20F Unknown Service with UUID 0xFFE0 Characteristics.

Characteristic	UUID	Property
Unknown Characteristic	0xFFE4	notify

Table A.9: Wellue FS20F Unknown Service with UUID 0xFFE5 Characteristics.

Characteristic	UUID	Property
Unknown Characteristic	0xFFE9	notify, write no response

Table A.10: Wellue FS20F Unknown Service with UUID 0xFFC0 Characteristics.

Characteristic	UUID	Property
Unknown Characteristic	0xFFC1	write no response
Unknown Characteristic	0xFFC2	notify

Table A.11: Wellue FS20F Unknown Service with UUID 0xFF90 Characteristics.

Characteristic	UUID	Property
Unknown Characteristic	0xFF91	read, write, write no response
Unknown Characteristic	0xFF92	notify, write no response
Unknown Characteristic	0xFF93	read, write, write no response
Unknown Characteristic	0xFF94	read, write, write no response
Unknown Characteristic	0xFF95	read, write, write no response
Unknown Characteristic	0xFF96	read, write, write no response
Unknown Characteristic	0xFF97	read, write, write no response
Unknown Characteristic	0xFF98	read, write, write no response
Unknown Characteristic	0xFF99	read, write, write no response
Unknown Characteristic	0xFF9A	read, write, write no response

Table A.12: Wellue FS20F Unknown Service with UUID 0xFD00 Characteristics.

Characteristic	UUID	Property
Unknown Characteristic	0xFD01	write no response
Unknown Characteristic	0xFD02	notify, write no response

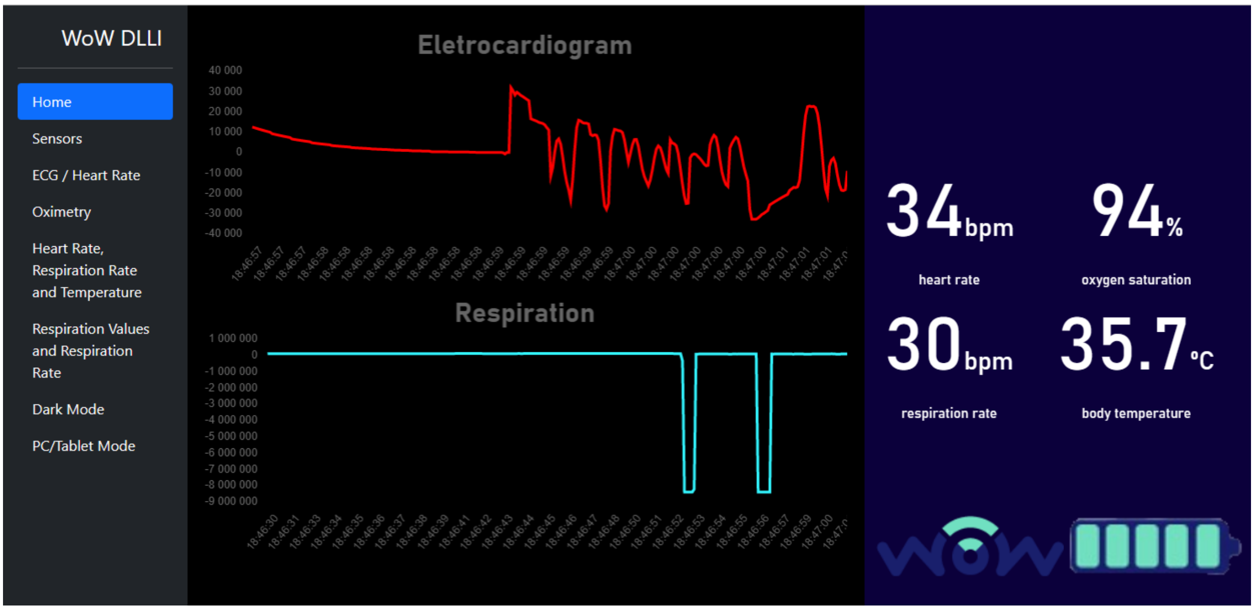


Figure A.1: GUI home page in dark mode.

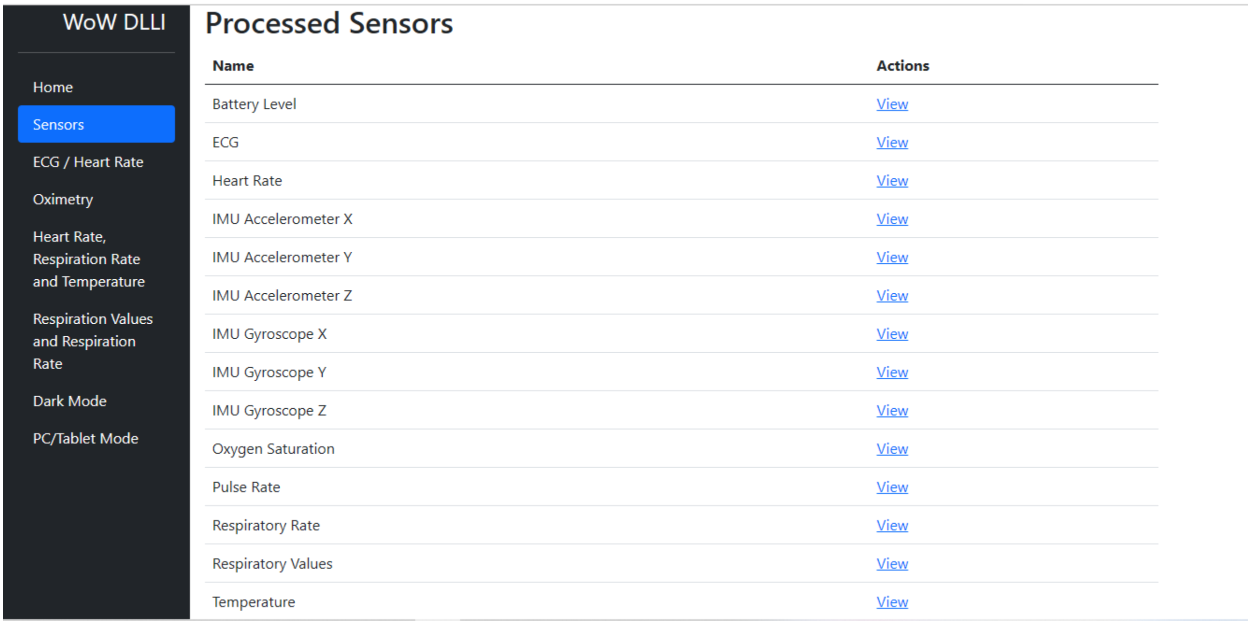


Figure A.2: Sensors list in light mode.

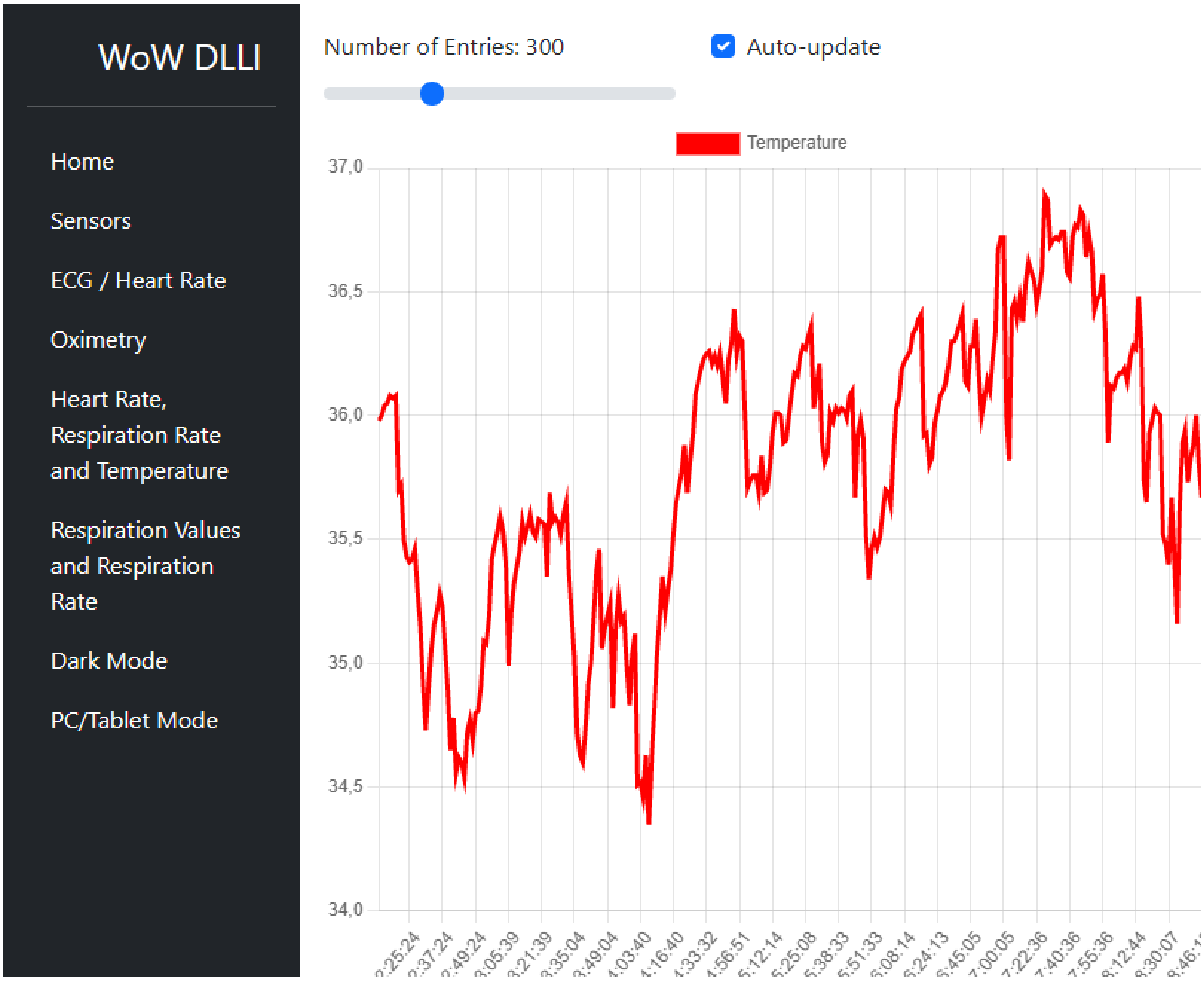


Figure A.3: Temperature graph.

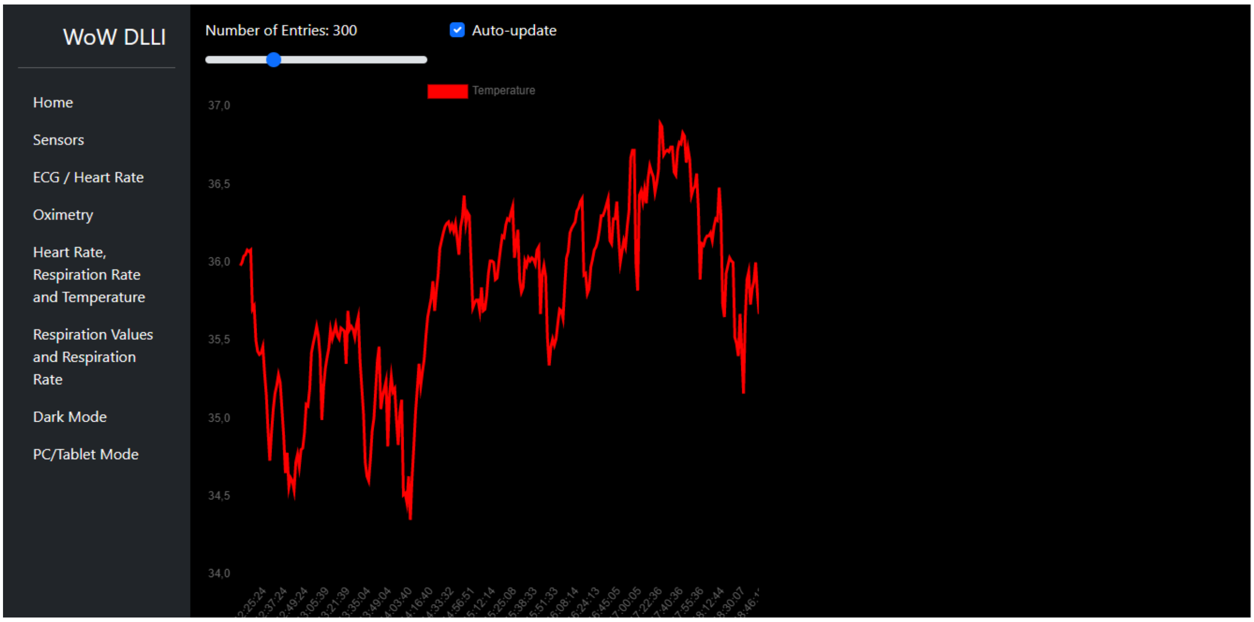


Figure A.4: Temperature graph in dark mode.

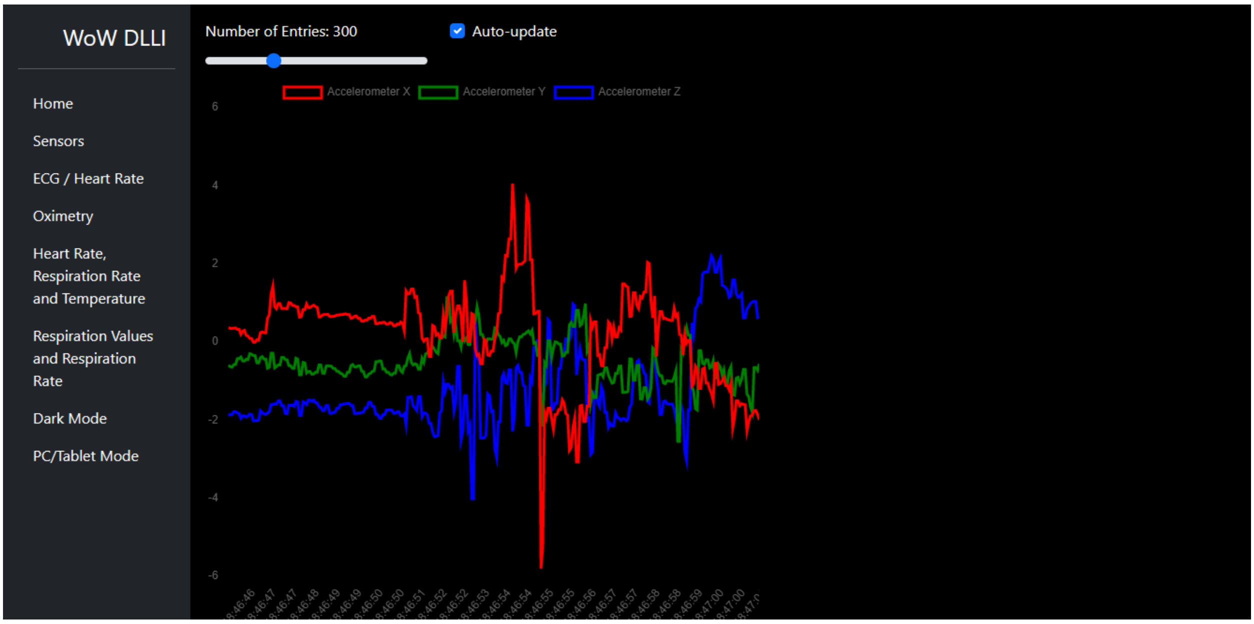


Figure A.5: IMU graph in dark mode.

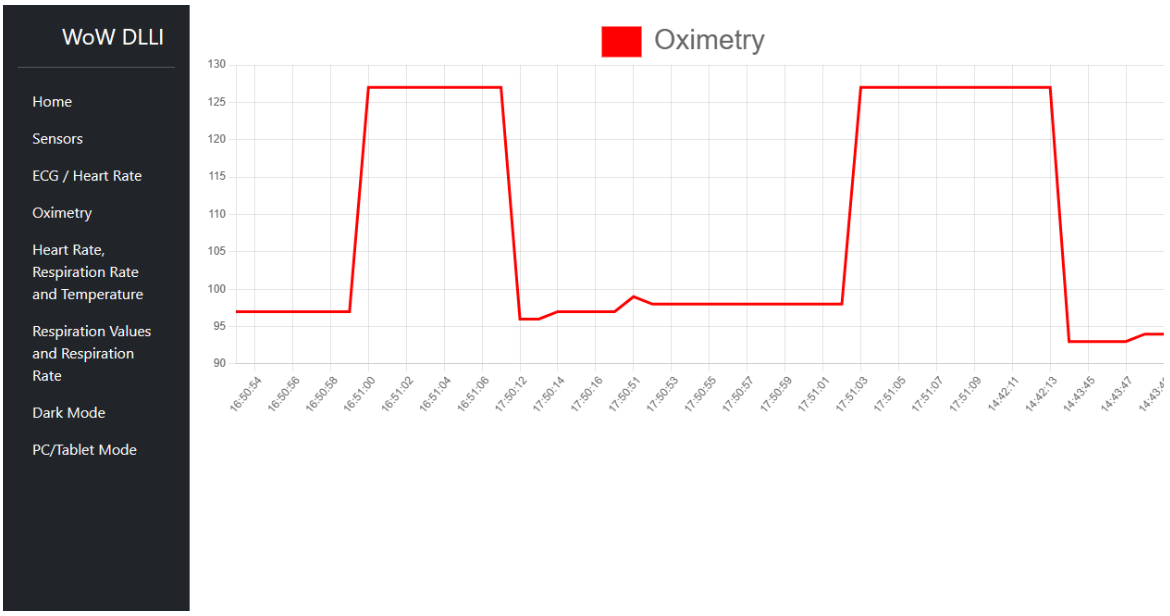


Figure A.6: Oximetry graph in light mode.

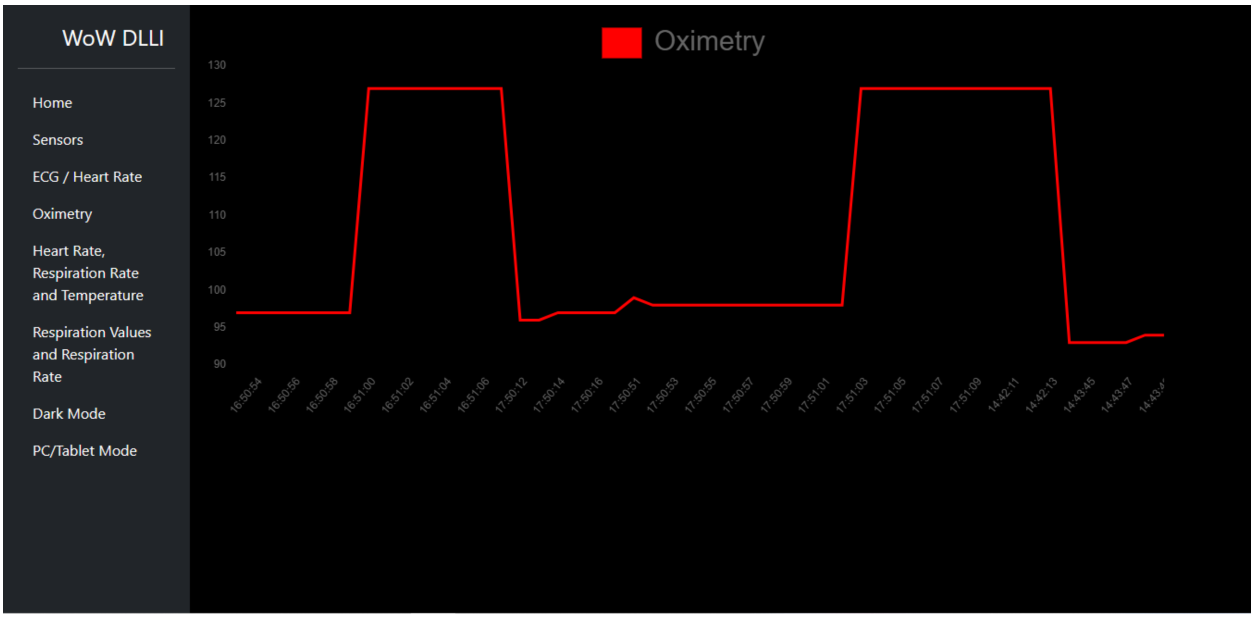


Figure A.7: Oximetry graph in dark mode.

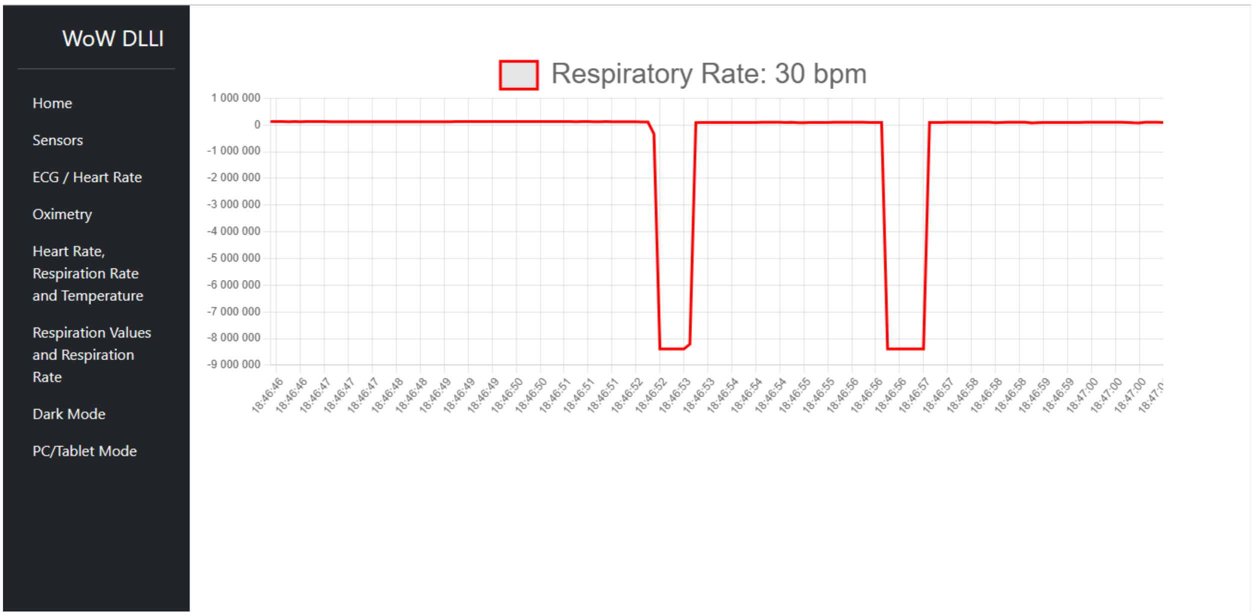


Figure A.8: Respiration graph in light mode.

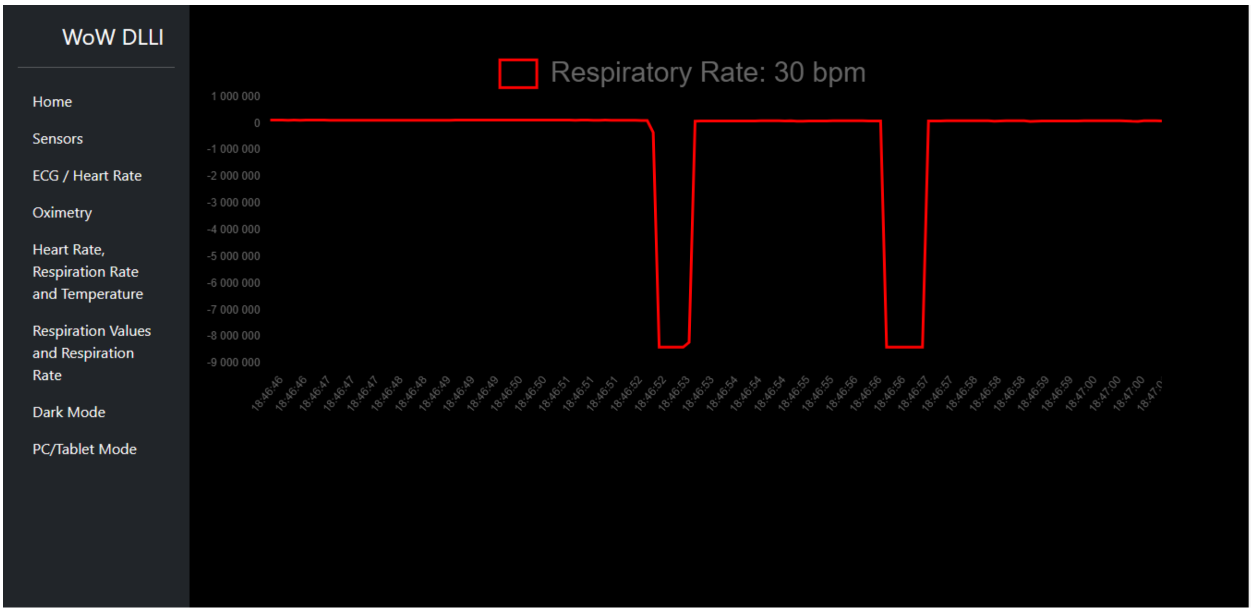


Figure A.9: Respiration graph in dark mode.

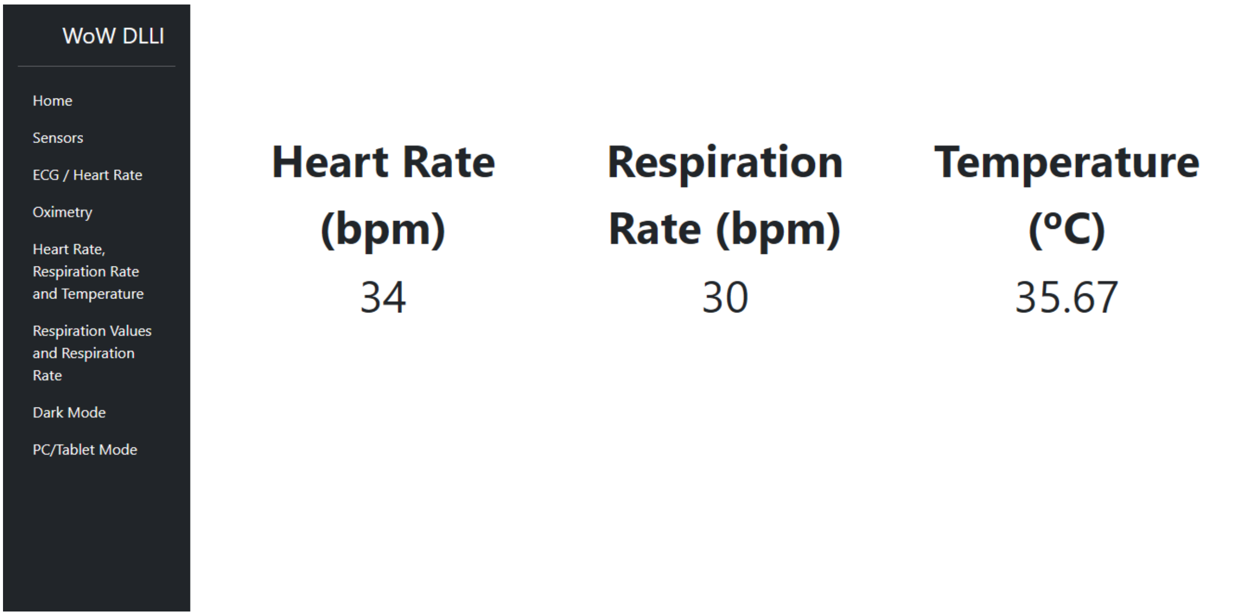


Figure A.10: Heart rate, respiration rate and temperature measurements.

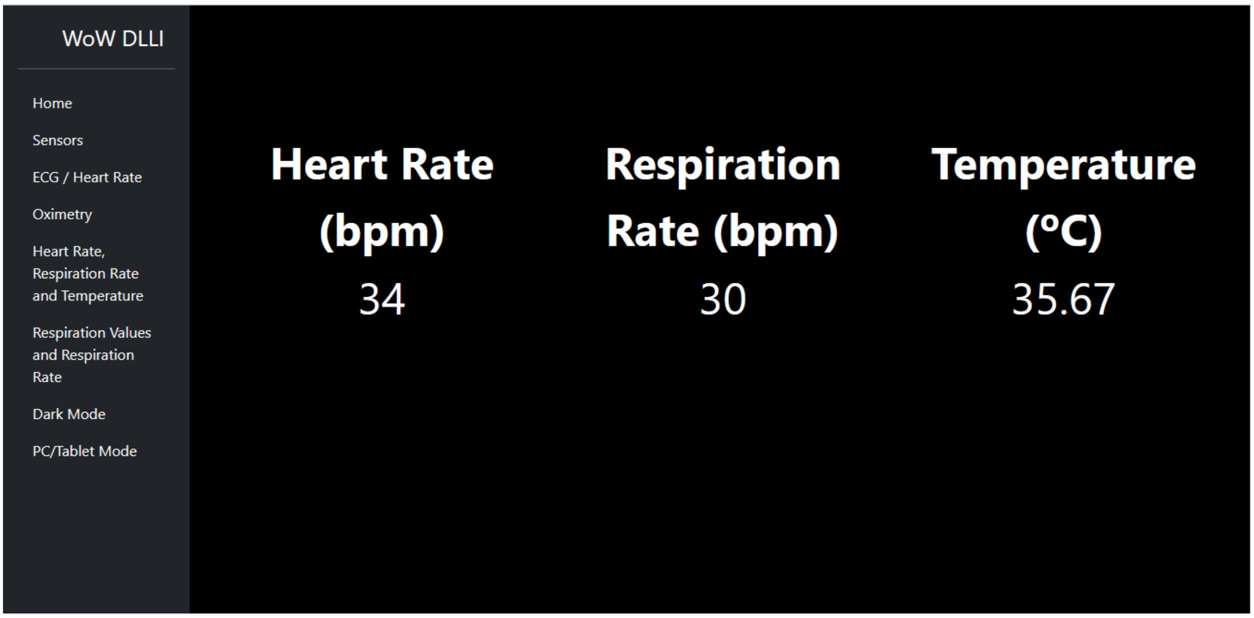


Figure A.11: Heart rate, respiration rate and temperature measurements in dark mode.