RTLS Tag – Indoor mobile positioning device

Pedro Bobião Costa

UMinho | 2022

Universidade do Minho
Escola de Engenharia

Pedro Bobião Costa

RTLS Tag - Indoor mobile positioning device

October 2022

Universidade do Minho
Escola de Engenharia

Pedro Bobião Costa
A80565

RTLS Tag – Indoor mobile positioning device

MSc Dissertation
Integrated Master's in Engineering and Management of Information Systems

Thesis performed under supervision of
Professor Adriano Moreira
Professor Filipe Meneses

October 2022

COPYRIGHT

ACKNOWLEDGEMENTS

I would like to give my parents and sister endless thanks for their constant love and for promoting the value of education.

I would like to thank Francisca Barros for her presence, understanding and unconditional love throughout this journey. Without her, I would have encountered far greater difficulties.

Throughout this development process, I acknowledge professors Adriano Moreira and Filipe Meneses for their invaluable suggestions and support.

"Velha Guarda" is a group of friends who have offered me friendship and encouragement over the past five years. We have truly made Universidade do Minho a home.

The ERASMUS+ friends I have made throughout our months abroad have always been a source of love and support to me. I will always cherish these memorie

DECLARATION OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

RTLS Tag – Indoor mobile positioning device

Regarding mobile services, one of the most popular classes is location-based services (LBS), as they significantly impact everyday lives, for example, when it is necessary to find the exact position of a person or object in the real world. These services are primarily available outdoors, typically using the Global Positioning System (GPS). GPS has benefits daily, however, and despite high accuracy and precision, these systems require line-of-sight access to orbiting satellites, and therefore, indoors, the signal is compromised. Consequently, positioning techniques for indoor environments are becoming a market segment that provides new business opportunities.

This study aims to develop a small device (named tag) that incorporates the Wi-Fi and BLE technologies and brings a solution to the problem of Indoor Localisation. Among others, the work includes a literature review about indoor localisation, the different existing techniques, a comparative analysis of algorithms, an analysis of technologies, the exploration of solutions on the market and a comparative analysis concerning performance and applications, and the proposal of a new tag and its evaluation. To this end, the orientations of a research method were followed, contemplating four steps: familiarisation with the area of positioning and navigation in indoor spaces; identification of desirable (optimal) characteristics for the tag and possible hardware solutions; tag development (hardware and programming) and performance evaluation of the developed solution. It is expected that this new tag to allow better development of indoor localisation systems, overcoming the difficulty felt in other solutions and responding to the need for an indoor localisation product, enhancing more excellent capability.

Keywords: Bluetooth Low Energy, Fingerprinting; Indoor Localisation and Wi-Fi

RESUMO

RTLS Tag – Dispositivo móvel de posicionamento em espaços interiores

Quando se refere o conceito de serviços móveis, umas das classes mais populares são os serviços baseados na localização (LBS), uma vez que estes têm um grande impacto nas vidas quotidianas, por exemplo quando é necessário encontrar a posição exata de uma pessoa ou objeto no mundo real. Esses serviços estão disponíveis, na sua maioria, em espaços ao ar livre, normalmente utilizando o *Global Positioning System* (GPS). O GPS apresenta benefícios diariamente, contudo, e apesar de apresentar níveis de exatidão e precisão altos, estes sistemas requerem acesso de linha de vista com os satélites em órbita, pelo que em espaços interiores o sinal fica comprometido. Desta forma, as técnicas de posicionamento para ambientes interiores estão a tornar-se num segmento de mercado que proporciona novas oportunidades de negócios.

O objetivo deste estudo é desenvolver um pequeno dispositivo (designado por *tag*), que incorpore as tecnologias Wi-Fi e BLE e traga uma solução para o problema de Localização *Indoor*. Entre outros, o trabalho inclui uma revisão de literatura sobre a localização *indoor*, sobre as diferentes técnicas existentes, uma análise comparativa de algoritmos, uma análise de tecnologias, a exploração de soluções no mercado e análises comparativas de desempenho e aplicações, e a proposta de uma nova *tag* e a sua avaliação. Para tanto, seguiram-se as orientações de um método de pesquisa, contemplando quatro etapas: familiarização com a área de posicionamento e navegação em espaços interiores, identificação de características desejáveis (ótimas) para a *tag* e possíveis soluções de *hardware*, desenvolvimento da *tag* (*hardware* e programação) e avaliação do desempenho da solução desenvolvida. Espera-se que esta nova *tag* permita um melhor desenvolvimento da localização *indoor*, ultrapassando a dificuldade sentida em outras soluções e respondendo à necessidade de um produto de localização *indoor*, potenciando uma maior capacidade.

Palavras-chave: *Bluetooth Low Energy*, *Fingerprinting*, Localização *Indoor* e *Wi-Fi*

INDEX

LIST OF ACRONYMS

AOA Angle of Arrival

AP Access Point

API Application Programming Interface

BLE Bluetooth Low Energy

GNSS Global Navigation Satellite System

GPS Global Positioning System

HTTP Hypertext Transfer Protocol

ILS Internet Locator Server

IMU Inertial Measurement Unit

INS Inertial Navigation System

IPv4 Internet Protocol version 4

ISM Industrial, Scientific and Medical band

IoT Internet of Things

JSON JavaScript Object Notation

LBS Location-Based services

PDR Pedestrian Dead Reckoning

PTH Pin Through Hole

RSSI Received Signal Strength Indicator

RTLS Real-Time Location System

SMD Surface-Mounted Device

TDOA Time Difference Of Arrival

TOA Time Of Arrival

UWB Ultra-Wideband

WLAN Wireless LAN

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This chapter briefly introduces the project and the motivations behind its development. It also explains how the document is structured and its objectives.

1.1. Project Context

Indoor positioning consists of locating an object or a person inside buildings. In an outdoor space, the GPS, or other Global Navigation Satellite Systems (GNSS), can be used to estimate the position of something with reasonable accuracy. However, in an indoor area, without a direct line of sight to the satellites, it is impossible to use the GPS to accurately know the position of something. Thus, and thanks to this problem, indoor localisation has suffered a great demand from companies and researchers over time.

Several application examples can be mentioned: storage companies, where there is a constant concern to know where the articles are inside the warehouse; in health, to know the location of patients and medical/nurse staff and in shopping centres and car parks, to know how to reach the desired stores and empty spaces, respectively. In short, it is necessary to resort to positioning and navigation systems for indoor environments.

1.2. Project Objectives

Regarding the objectives, it can be mentioned that despite the positioning technology in indoor environments has evolved a lot in recent years, it continues to be verified that few mobile positioning devices (commonly called tags) are available for integration into other systems. Additionally, it appears that the capabilities of existing tags are reduced and, above all, limited in communication with open systems.

To design a good solution, it is essential to obtain information about the past state, present and future expectations. The future state is directly related to capacity planning, ensuring that the solution developed and implemented in the present is supported as the environment changes. Therefore, this work aimed to design and implement a tag for indoor positioning systems. Secondary objectives were defined that helped to respond to the main goal, namely:

- Interpret the problem and select a microcontroller that fulfils the main objective;
- Use Wi-Fi technology to scan networks;
- Use Bluetooth Low Energy technology to scan devices;

- Post scan information to an interface;
- Set parameters for scans externally and save them permanently;
- Test tag's autonomy with a powerbank.

In conclusion, the tag can identify existing Wi-Fi and BLE networks in a location and report the collected radio information to an existing service. The tag was built using low-cost computing platforms (Arduino style), and its implementation was supported by the experience already developed within the scope of ongoing research projects.

1.3. Document Structure

The remainder of this document is arranged as follows. Chapter 2 briefly describes Indoor Localisation techniques and technologies in general. It also presents some related work, briefly describing prior work on Indoor Localisation. Chapter 3 describes the process of designing the whole system. Chapter 4 gives an overview of the work plan development, describing the activities conducted and software implementation. Chapter 5 provides an overview of evaluation planning and testing. Chapter 6 concludes the document and Chapter 7 provides all extra support for this thesis.

2. LITERATURE REVIEW

An overview of the research strategies for constructing the literature review is presented in this chapter. In this chapter, topics related to Indoor Localisation are discussed, and finally, a quote on the critical view of the state-of-the-art is presented, where the author shares his personal thoughts.

2.1. Research Strategy

The Research Strategy is the step-by-step action plan that directs the thought process. It enables to conduct the research systematically and on schedule. The primary purpose is to introduce the principal components of the study, such as the research topic, areas, central focus, research design and methods. At first, the subject was proposed by the advisors, and research had to be conducted to fully understand the topic of Indoor Localisation since it was the first time the author would cover it. Research strategy involves the analysis of available information, such as articles covering the topic of indoor localisation that were recommended by the advisors or were found to be related to the topic and relevant for gaining insight into the subject. Human experience provided by the advisors was also crucial to better understand the research problem and critically analyse and investigate other researchers' prior findings to formulate a solution to the problem.

2.2. State-of-the-Art

Several key and essential concepts are explained in state-of-the-art to support all the work proposed in the dissertation. As such, this section is divided into subsections that tackle different components of the topics, such as Indoor Localisation and Signal Properties.

2.2.1. Indoor Localisation

Indoor localisation is a method used to determine or estimate the location of a user or a device in an indoor environment. Indoor location systems have made it easier to locate people or objects. Global Navigation Satellite Systems (GNSS) can provide reasonable location estimates in an outdoor environment. However, GNSS-based solutions cannot be used indoors. In this kind of environment, typically called a GPS-denied environment, the satellite signals are inferior, resulting from a lack of line of sight between the satellites and the receiver. There has been an increased demand for indoor localisation due to its vast number of applications. Therefore, another solution had to be made using sensors and technologies to enable localisation (Zghair et. al, 2019).

Indoor location systems have become a vital topic of research and development in recent years due to the wide variety of applications they can offer in indoor environments. More than a hundred companies are working on positioning, tracking and navigation technologies and systems. Figure 1 illustrates the concept of navigation.

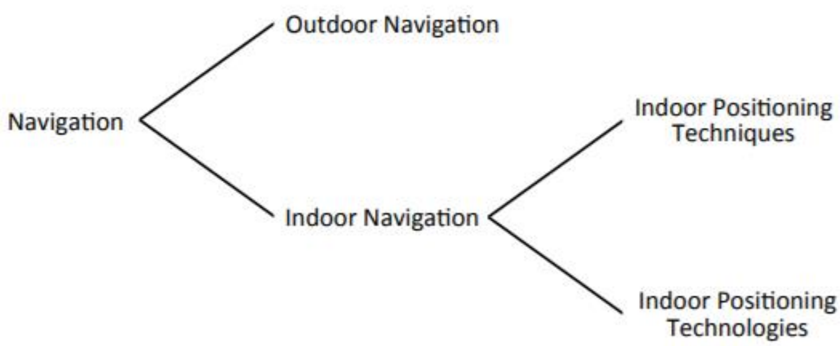


Figure 1 - Flow diagram of Navigation[1].

There are many applications of indoor localisation: detection of the location of products in the warehouse; monitoring patients, staff, and equipment in hospitals to improve navigation; helping customers locate stores or products in a shopping mall; detecting the location to find the place where the car was parked in a parking lot.

2.2.2. Indoor Positioning Techniques

Positioning techniques for indoor environments can be based on estimating the distance to anchor nodes with known positions and on internode measurements. Node cooperation enhances position estimation and is primarily beneficial when traditional localisation techniques fail to accurately assess (Yassin et al., 2016).

This section divides the positioning techniques by signal properties and positioning algorithms, as shown in Figure 2.

[1] This image was taken from the article "A state-of-the-art survey of indoor positioning and navigation systems and technologies" (Sakpere et al., 2017).

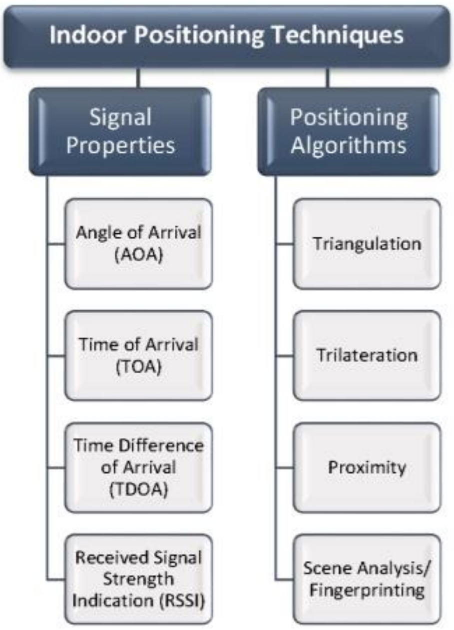


Figure 2 - Indoor Positioning Techniques[2].

2.2.3. Signal Properties

The use of geometrical parameters is necessary, such as angle, distance and signal intensity, known as signal properties, to calculate the position of a device (Sakpere et al., 2017).

Angle Of Arrival (AOA)

The Angle Of Arrival technique uses antenna arrays at the receiver's end to calculate the angle by which the transmitted signal affects the receiver by employing and calculating the time difference in the arrival at individual antenna array elements. The position is estimated through the intersection of direction lines starting at the reference points (Sakpere et al., 2017).

The angle of arrival's main advantages is that the device or the user's position can be measured in a 2D environment with as few as two measuring units or three measuring units in a 3D environment and that no synchronisation of time is required between the measuring units. However, the drawback is that it has relatively complex hardware requirements, and location estimate deterioration will happen as the mobile target moves away from the units of measurement. Figure 3 illustrates the concept of the Angle of Arrival.

[2] This image was taken from the article "A state-of-the-art survey of indoor positioning and navigation systems and technologies" (Sakpere et al., 2017).

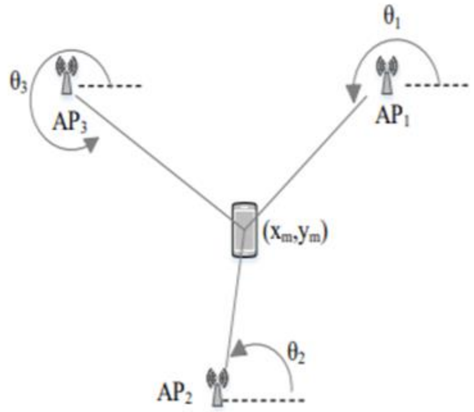


Figure 3 - Localisation based on Angle of Arrival (AOA) measurement[3].

Time Of Arrival (TOA)

Time of Arrival is used to measure the distance between the transmitter and the receiver by using the signal propagation time. The distance between the reference node and the device can be calculated by multiplying the propagation speed with the TOA value. Time of Arrival requires strict coordination between the transmitter and the receiver, which adds to the system's cost (Subedi & Pyun, 2020). It has been recognised that with the aid of ultra-wideband technology, the time of arrival technique could best support the fine-time resolution. Frequency domain resolution techniques are widely used to achieve a high-resolution Time of Arrival from the channel frequency response. Figure 4 shows an example of the Time of Arrival technique.

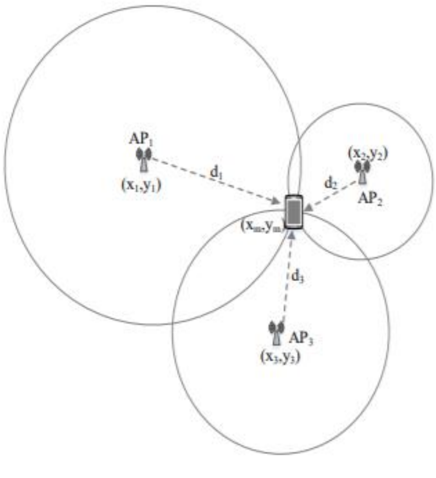


Figure 4 - Localisation based on Time of Arrival (TOA) measurement[4].

---

[3] This image was taken from the article "A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies" (Subedi & Pyun, 2020).

[4] This image was taken from the article "A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies" (Subedi & Pyun, 2020).

TDOA (Time Difference of Arrival)

The Time Difference of Arrival technique takes advantage of the difference in signal transmission times measured at the receiver from different transmitters. TDOA differs from the time of arrival technique, where total signal propagation time is used. In contrast, the latter time measurements are used instead of optional time measurements at each receiving node. The transmitter must be in a hyperboloid with a constant difference in the range between the two units of measurement for each TDOA measurement. The time difference of the arrival technique does not require the receiver's synchronisation; however, the transmitters need to be synchronised (Yassin et al., 2016). TDOA is usually implemented, as shown in Figure 5.

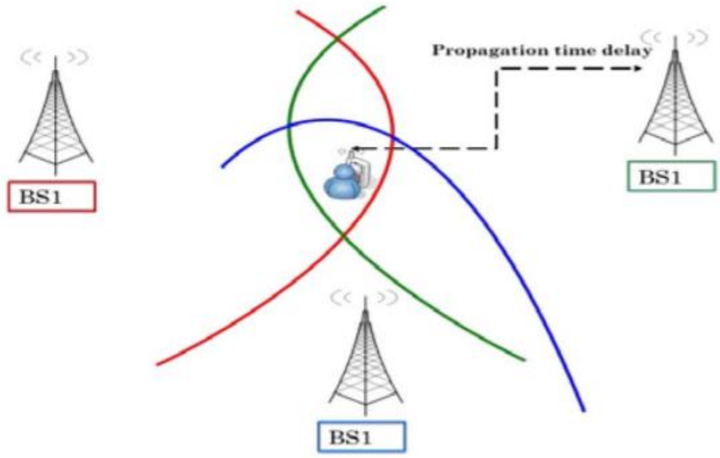


Figure 5 - Time Difference of Arrival (TDOA) technique[5].

RSSI (Received Signal Strength Indicator)

Received Signal Strength Indicator, as shown in Figure 6, is one of the most used techniques for indoor localisation. It is based on measuring the strength of a signal received from an access point at a client device. More information can be collected in proportion to the increase in available access points. Hence, accuracy can be increased over the information obtained. However, it also works as a trade-off, as more access points would increase the interference between different signals (Yassin et al., 2016). While RSSI-based localisation is one of the cheapest and most straightforward techniques to implement, its accuracy is not the best. Therefore, filtering is needed to improve its accuracy.

[5] This image was taken from the article "Recent Advances in Indoor Localisation: A Survey on Theoretical Approaches and Applications" (Yassin et al., 2016).

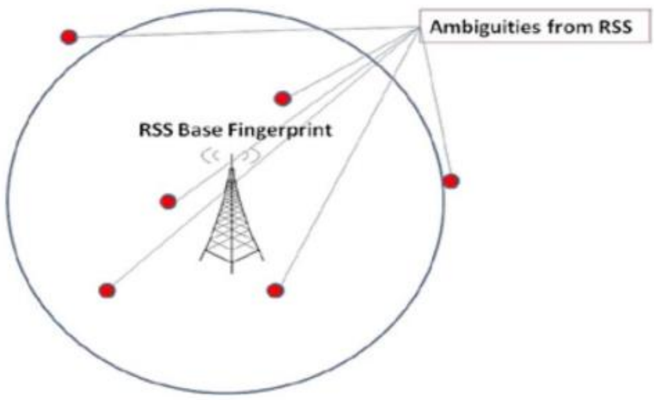


Figure 6 - RSS based Fingerprinting Approach[6].

Advantages and Disadvantages

Every positioning technique has advantages and disadvantages, but most lack accuracy (Yassin et al., 2016). The advantages and disadvantages of the position techniques are listed in Table 1[7].

Table 1 - Summary of Signal Properties[7]

Signal Property	Measurement Metric	Pros	Cons
Angle Of Arrival (AOA)	Angle-based	High accuracy at room level	Complex, expensive and low accuracy at wide coverage
Time Of Arrival (TOA)	Distance-based	High accuracy	Complex and expensive
Time Difference Of Arrival (TDOA)	Distance-based	High accuracy	Expensive
Received Signal Strength Indicator (RSSI)	Signal-based (RSS)	Low cost	Medium accuracy

2.2.4. Positioning Algorithms

[6] This image was taken from the article "Recent Advances in Indoor Localisation: A Survey on Theoretical Approaches and Applications" (Yassin et al., 2016).

[7] This table was taken from the article "A state-of-the-art survey of indoor positioning and navigation systems and technologies" (Sakpere et al., 2017).

To calculate the position of an object, positioning algorithms are used. These process the signal property and output a position so that, this way, the positioning algorithms allied to the signal property determine the position of an object. It is also important to note that positioning algorithms have advantages and disadvantages, so using multiple of them will improve position accuracy and performance (Sakpere et al., 2017).

Triangulation

The triangulation technique is used in conjunction with the geometric concept of triangle angles to determine the receiver's position. Rather than measuring distances between nodes like trilateration, triangulation-based positioning is based on angle measurements. Although they work similarly, the position of the target node can be determined by intersecting several pairs of angle direction lines with triangulation.

This approach requires only two measuring devices to estimate the location in a two-dimensional plane (Subedi & Pyun, 2020). Three or more devices are usually used for location estimation to enhance accuracy. It requires highly directive antennas or antenna arrays to measure the direction.

Trilateration

The trilateration technique estimates the object's location by measuring the distance from three reference points based on the triangle's geometric properties. In trilateration, at least three fixed points are needed to determine a position in 2D. Depending on three non-collinear distance measurements, the target object's absolute location is calculated (Subedi & Pyun, 2020).

The target object's location is determined by the intersection of imaginary circles in a 2D dimension plane.

Scene Analysis / Fingerprinting

Algorithms for scene analysis begin by collecting features that are the fingerprints of a scene, then match online measurements with the closest prior location fingerprints to estimate the location of an object. RSSI-based location fingerprinting is commonly used in scene analysis (Subedi & Pyun, 2020).

Fingerprinting technique uses radio signals to learn the user or device's location. It works by collecting the signal values (fingerprints) during an offline phase, measuring the different calibration points throughout the area of interest, and recording it in a database which will then be compared with the online measurements where the system is deployed to estimate the user's location. During the online part, the constructed radio map will be used to approximate the tracked object's position (Eisa et al., 2013). Online positioning, therefore, requires a selection process for the access points. A range of algorithms is presented to match offline calculations with online calculations. It is essential to know that fingerprinting techniques can provide accurate localisation estimates; however, they are susceptible to environmental changes over time.

Characteristics of Positioning Algorithms

The characteristics of each Positioning Algorithm are described in Table 2[8].

Table 2 - Summary of Positioning Algorithms[8]

Positioning algorithm	Signal property	Pros	Cons
Triangulation	AOA	Simple, low-cost and high accuracy at room level	Complex, expensive and low accuracy at wide coverage
Trilateration	TOA/TDOA	High accuracy	Complex and expensive
Scene analysis/fingerprinting	RSSI	High performance	Complex, expensive, medium accuracy and time consuming

2.2.5 Indoor Localisation Technologies

Numerous technologies can ensure that someone does not get lost geographically. There are artificial and natural techniques, such as satellite and radio signals, celestial bodies and natural landmarks, respectively (Sakpere et al., 2017).

This section demonstrates several existing technologies that aim to provide indoor location services. This theme presents a diversity of algorithms and techniques. Many technologies have evolved to respond to positioning and navigation. However, in this section, indoor environments will be the main focus.

- Wi-Fi

The Wi-Fi technology, also known as IEEE 802.11 standard, operates in the Industrial, Scientific, and Medical (ISM) band. It is mainly used to provide networking capabilities and access to the Internet to various devices

[8] This table was taken from the article "A state-of-the-art survey of indoor positioning and navigation systems and technologies" (Sakpere et al., 2017).

in personal, public, and industrial environments. Wi-Fi nowadays has a reception range of about 1 kilometre (km) in special conditions and using high-gain antennas.

Most of the newest laptops, smartphones and other mobile user devices are Wi-Fi enabled, making it the best suitable for indoor localisation and one of the most examined technologies. Existing Wi-Fi networks are typically deployed for communication purposes rather than location; therefore, new and efficient algorithms are needed to improve their location accuracy (Zafari et al., 2017).

Challenges in Wi-Fi-based Localisation System

Wi-Fi based location systems are time-consuming (surveying the site), cumbersome and multipath is influenced by physical objects, which may interfere with other applications on the 2.4GHz ISM band. It is also important to mention that the variation of the signal strength concerning time is considered a weakness of this system, causing deterioration in the location accuracy. Frequent updating of the signal strength map is crucial, as the movement of people, doors and furniture often causes a variation in signal strength. It is therefore considered the most significant disadvantage of Wi-Fi fingerprint systems (Yassin et al., 2016).

- Bluetooth

Established as IEEE 802.15.1, it comprises the specification of physical and MAC layers to connect various fixed or moving wireless devices within a particular personal space. Bluetooth Low Energy (BLE), commonly defined as Bluetooth smart, the latest version of Bluetooth, delivers an increased data rate of 24Mbps and more energy efficiency with a coverage range of 70-100 meters. While BLE can be used with different localisation techniques, such as RSSI and AoA, most existing BLE-based localisation solutions rely on RSS-based inputs as RSS-based systems are less complex (Zafari et al., 2017).

The advantage of using Bluetooth technology in indoor localisation is that devices are small and easy to find.

Challenges and drawbacks of Bluetooth-based positioning

Problems associated with BLE make such positioning systems difficult. The range (coverage) and signal of radio waves are limited, and they are prone to disturbances such as storms and organic matter, which can disturb or distort radio signals. As an RF signal travels through air and other media in an indoor environment, it exhibits certain propagation behaviours or effects, such as absorption, reflection, scattering, refraction, interference, multipath, and attenuation. These behaviours are signal impediments that affect the transmission of a signal between two locations, resulting in signal degradation and loss. Its effects can sometimes be useless and harm performance and accuracy.

- Ultra-Wideband (UWB)

This technology works in the microwave band, where high frequencies can penetrate building materials like walls. The technology is used mainly for short-range communication systems like laptop peripherals and alternative indoor applications.

The ultra-wideband has been a very enticing technology for indoor localisation because of its resistance to interference from different signals, low power consumption, and effective penetration through dense materials (Subedi & Pyun, 2020).

Challenges in UWB-Based Localisation System

UWB technology faces numerous challenges induced by extremely cluttered operating environments that cause multipath, NLOS and shadow artefacts. These challenges are cited in the following list:

1. Broadband radio devices are hard to implement for a bandwidth greater than 500 MHz. The advancement of technology has enabled some studies to develop platforms with accuracies of 10 to 15 centimetres or more.
2. Ultra-Wideband interference occurs when the UWB signal is configured and propagated incorrectly over the same bandwidth as the existing narrowband system. For its implementation, at least three receivers with a direct path unlocked to the transmitter are required for the standard ToA positioning algorithm.
3. For the system to be effective and achieve maximum accuracy, it is necessary to perform the acquisition, tracking and signal synchronisation with high precision concerning the pulse rate.

All these problems are currently being studied and analysed towards the evolution of technology (Yassin et al., 2016).

- Ultrasound

Ultrasound-based location systems are referred to as delicate grain systems with centimetre-level location accuracy. They depend on the time of arrival calculation of the ultrasound signal measured using sound velocity. Ultrasound networks which have been built are either narrowband or wideband. Relatively to the ultrasound positioning system, it has high room-level accuracy. Some ultrasound positioning systems use narrowband or wideband signals that have shown high accuracy during implementation. Several types of ultrasonic tags can be used on both people and objects. These tags serve as receivers or transmitters; when one is fixed, the other is in motion (Sakpere et al., 2017).

Ultrasound position systems deliver various benefits like low device cost, accuracy, reliability, scalability, and high-power performance.

Challenges and drawbacks of Ultrasound

Ultrasonic positioning systems are typically cheap at room level. A large-scale implementation would diminish its advantages as it would be costly to deploy and maintain. In addition, ultrasound systems experience multipath effects such as noise, reflection, and interference. Thus, system accuracy and performance are degraded (Sakpere et al., 2017).

- Dead Reckoning

The dead Reckoning technique estimates the current position based on the last calculated position and motion information. Here two approaches can be used to estimate a person's travelled distance. The first is pedometer-based dead reckoning, in which step detection and step length are estimated, and then steps are counted to measure the distance travelled. The other is pedestrian dead reckoning which calculates the distance travelled directly. The inertial measurement unit (IMU) is attached to a person's foot and calculates the acceleration and angular velocity changes. The data obtained is then analysed and transformed into referential motions, and the movement's distance and orientation are measured. Once initialised, this system can continuously calculate its position, direction and velocity independently, without external references. It is simple, low-cost and can estimate positions in real-time. Moreover, it can combine smoothly and efficiently with other positioning systems due to its inertial sensors. Affectless by signal issues, emitting no signals or radiation and does not require a network to operate.

Challenges and drawbacks of Pedestrian Dead Reckoning

A significant problem with PDR and INS is the gradual drift in operation due to the progressive accumulation of errors over time during motion leading to low accuracy due to sensor noise, bias, and magnetic disturbances. There is the possibility of using a hybrid system with a GPS satellite system or other positioning systems to eliminate this error. Thus, the system's complexity increases, as does the cost (Sakpere et al., 2017).

2.2.6 Performance

The development of indoor positioning and navigation has been studied and analysed, and consequently, evolving to guarantee scalability, complexity, privacy, usability and precision, among others. Several issues, challenges and limitations have been explored to respond to the need for progress in these systems.

- Accuracy

Good accuracy is a significant user prerequisite for positioning systems. Different systems can have different accuracies. However, accuracy is still a very challenging area of interest for many researchers. As mentioned earlier, indoor environments provide a challenging space for location systems to operate due to obstacles and multipath effects. Therefore, the system needs to limit the impact of multipath effects and other ambient noise to achieve highly accurate systems that require extensive signal processing and noise mitigation which is a highly challenging task. The location system must locate the user or object of interest, ideally with an accuracy of 10 cm (known as micro-location) (Zafari et al., 2017).

- Cost

Another challenge is the monetary value of the system. Localisation systems require additional infrastructure that giant corporations might be able to afford, whereas smaller organisations are limited in the matter of costs. Hence, some of the infrastructure's equipment and bandwidth can be avoided using the existing infrastructure. Therefore, the location system can easily penetrate the consumer market and be widely adopted while keeping costs low (Zafari et al., 2017).

- Energy Efficiency

The indoor localisation system's energy efficiency is critical for widespread acceptance. Most current localisation systems use incredible energy to deliver higher precision. In today's localisation systems, achieving high accuracy without draining the device's battery is challenging, leading to user dissatisfaction. Therefore, it is necessary to improve the energy consumption of the localisation systems. Technologies such as BLE can be applied as it has lower power consumption, offloading the computational aspect of the location algorithm to a server and any entity that has access to the uninterruptible power supply and has high processing power. The fundamental compromise is between the location system's energy consumption and latency (Zafari et al., 2017).

- Scalability

One of the essential requirements is that the system must be scalable. That is to be capable of simultaneously finding or providing facilities to many users in a large area. Low scalability can lead to poor system performance. A scalable positioning system functions correctly when its scope gets larger. Usually, localisation performance reduces with increased distance between the transmitter and receiver.

Further, a positioning system may require scaling on two axes, density and geography. Geographic scaling represents an area or volume coverage, whereas density scaling represents the number of units positioned per unit geographic space or area per period. Wireless signal channels may become congested as more area is covered or the units in such an area are crowded; hence, further computation or communication infrastructure may be required for localisation. In addition, the dimension of a system is another metric for scalability. A positioning system may locate objects in 2-D space, 3-D space or both (Liu et al., 2020). Because of all these challenges, there is much room for improvement in this field.

2.2.7 Applications

The LBS service is required for both indoor and outdoor environments. In this section, a detail of the main applications of indoor location is presented. Indoor positioning and navigation for mobile devices occupied 4 billion dollars in 2018, meaning that this is a reliable, easy-to-use and accurate solution for indoor positioning and navigation that has the potential to open the door to new applications. Creating new business opportunities in numerous scenarios is a cornerstone in realising the vision of the Internet of Things (Yassin et al., 2016).

A. Asset Management and tracking: Detection of the location of products in the warehouse

Location-based services include asset tracking and enable factory automation to ensure workplace safety. Asset management will also allow for better inventory management and optimised operation management.

Additionally, personnel and asset tracking analytics and motion data collection have proven outdated. Using novel energy-efficient techniques and algorithms will eliminate the need for expensive proprietary hardware (Khanh et al., 2020).

B. Health Services: Monitoring patients, staff, and equipment in the hospitals to improve navigation

The health sector can significantly benefit from indoor localisation as it can help save valuable lives. It can help the hospital staff, patients, and visitors. If a patient needs medical assistance, the current protocol requires broadcasting the message or paging a specific doctor or staff member who may not be in the vicinity of the patient. The staff's arrival delay might even cause the patient's death. Similarly, broadcasting the message will cause other staff members to receive irrelevant notifications. A location-based solution would allow tracking of the position of the staff members. In an emergency, the localisation system would find the staff member close to and with the necessary qualifications to handle the emergency (Marques et al., 2012), avoiding delay and not spamming the other staff members. Indoor localisation can also allow doctors to track various patients and their mobility to ensure patient safety. Visitors who intend to visit patients can find their destination using a localisation system without any hassle.

C. Contextual Aware Location-based Marketing: Help customers locate stores or products

There are several types of usability for these applications. However, they can direct users to a store located inside a building, access a specific book in a library, or the boarding gate at an airport, which benefits this system. iBeacons provide proximity-based services for travellers to enhance their experience. In this way, JFK in New York, Heathrow in London, Miami International and many other airports have started to use it so that all their tourists feel integrated with the whole space. As well as for public transport, such as buses, there is the LBS app that can be used at bus or train stations to navigate the bus stop (Zafari et al., 2017).

D. Location-Based Services: Detecting the location to find the place where the car was parked in a parking lot

With the growth of cities observed in recent decades, some spaces and buildings have become vast and complex, which is the case for large shopping centres and parking spaces. In these spaces, finding a particular car is not always easy, especially if the visitor is not a regular user of the same area. These problems can be minimised using navigation systems for indoor environments. These tools are like the standard GPS used to find the way to a specific destination. It facilitates navigation within the parking lot. The primary function of these solutions is to indicate the shortest route to a particular car.

2.3. Critical View of State-of-the-Art

This point is critical as it reflects the entire theoretical framework where the essence of this work fits. At first, it was sought to analyse other studies related to indoor localisation and several applications to obtain comparative information regarding the critical success factors present in a process of this nature. What values a tag from the users' perspective is the effectiveness of delivering quality services. Together, these services must provide the accurate geographical location of the user. The fundamental principle of indoor localisation is that it does not have access to satellites to know a user's location. As an indoor device, this tag does not focus on GPS or other outdoor technologies that enable localisation but instead on indoor technologies to provide a location in an indoor structure.

The main conclusions drawn from the state-of-the-art study were the following:

- Indoor localisation services are an integral part of localisation systems.
- Localisation is critical and knowing the whereabouts of any asset or user in an indoor environment is an untapped niche market.
- There is scientific evidence that effective solutions have been implemented for indoor localisation processes, but there is room for improvement.
- There are several good practice frameworks and standards for managing localisation. Several studies prove the benefits of increasing the different number of technologies and techniques, improving the efficiency and effectiveness of location services, noise effect reduction and improvement of the productivity of the solutions when implemented successfully.
- The most commonly used technology is the IEEE 802.11 standard, commonly known as Wi-Fi.
- The target technologies of this study case are Bluetooth and Wi-Fi to be implemented within a tag and using the fingerprinting technique. The literature review found studies that prove the existence of a high probability that this choice will lead to reaching the expected benefits.
- However, common mistakes can lead to tag development and implementation failure, which should be avoided.

The study case to be developed will focus on measuring the impact of the indoor localisation processes and this tag's efficiency, effectiveness, and robustness.

3. SYSTEM DESIGN

In extensive facilities or localised areas, real-time location systems (RTLS) tags enable people, objects and materials to be tracked digitally in real-time. A real-time location system in an intelligent factory can track workers in real-time, enabling critical safety applications such as collision avoidance and safety zoning. They will also track critical assets like production equipment, vehicles, and inventories to support enterprise asset management, automation, workflow optimisation and more. Healthcare can also benefit from clinical-grade RTLS tags to help track staff, patients and critical medical equipment such as defibrillators in real-time, which helps them deliver vital care in an urgent situation.

Before describing the details of the development process, a section explaining the approach is required. In the first phase, “off-the-shelf” hardware was used, and software was developed. Tag's requirements – functional (what it should do) and non-functional are also described in this section.

3.1. High-Level Architecture

The user interacts with the prototype through its two interfaces. Those are the user Input Form and the ILS Server. In the Input Form, developed in HTML, the user can submit data to the ESP32. This data is then stored in permanent files. Since the tag will be used in different places and different circumstances, there was a need to change the operating parameters quickly (without having to recompile the program and rewrite the software in the microcontroller), which led the author to develop a webserver (containing the Input Form) inside the tag. After the user completes the Input Form, the tag will acquire data from scanning Wi-Fi networks and Bluetooth devices and generate JSON objects with the data from the scans. These JSON objects are then loaded into a relational database and posted in the ILS interface to provide the desired information to the user. Considering these principles, Figure 7 shows the architecture of the developed system.

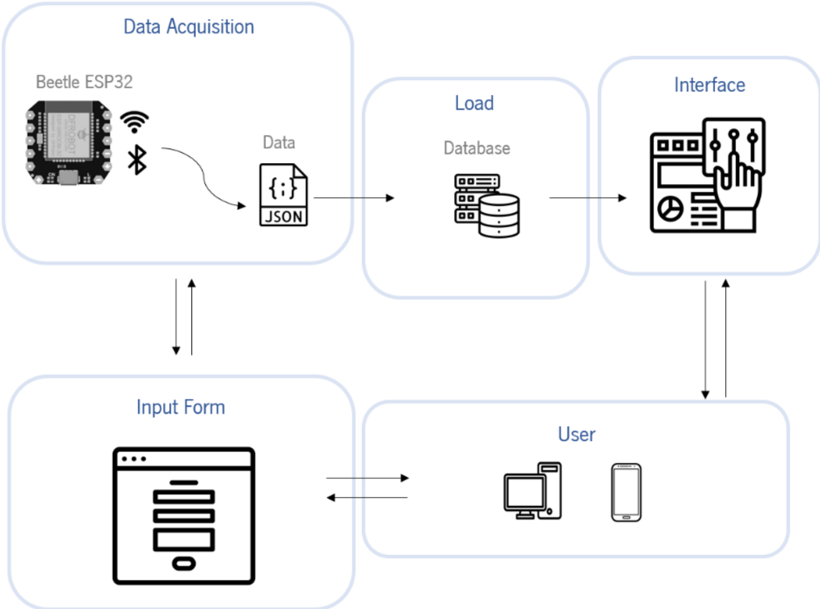


Figure 7 - High-Level Architecture.

It is worth noting that both the “Load” and “Interface” sections of the architecture were not developed by the author. It is an external software provided by advisors. The author is solely responsible for integrating these services with their implementation. The tag is part of a more comprehensive system where some components already exist. A feature already operational is a positioning engine that processes the tag's generated data. This engine has several interfaces, but for the interest of this work, the i2a interface receives readings made by the tag and allows the user to see what was registered on the server.

3.2. System Requirements and Specifications

The functional requirements are: the tag must start in access point mode; when users connect to it, an HTML form must be displayed so that they can change operational parameters (name of the tag, Wi-Fi and BLE scan intervals); then, the users have the option of starting the scans and posts right away or waiting for a delay of 1 minute to let the system start automatically. Before starting scans and posts, the parameter values are stored in the permanent file of the ESP32 (Figure 8).

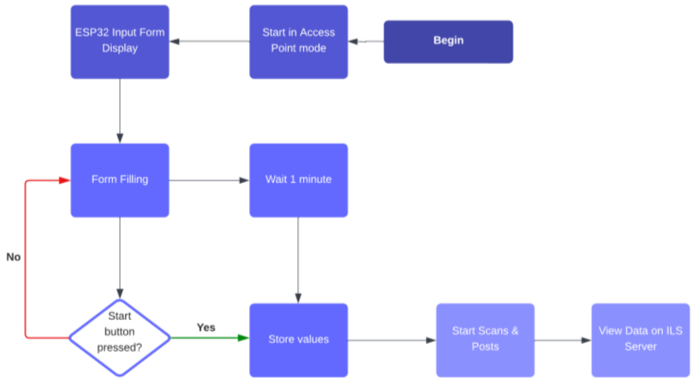


Figure 8 - System's Flowchart.

The ideal sequence would be to do Wi-Fi scans, send the data collected for the server, and do BLE scans and posts in parallel. This is the only way to understand the usefulness of the various threads running parallel to have Wi-Fi and BLE scans simultaneously. The tag's requirements are:

- Effectiveness

The tag must receive and send data with good transfer speeds, and there should be no information losses.

- Cost

The tag must be inexpensive to be economical to have multiple tags on various objects.

- Autonomy (energy consumption & battery)

The tags' autonomy must be pretty high (for example, several hours), so there is no need to recharge the tags' batteries constantly.

- Size and Weight

The size and weight of the tags must be minimal and light to be able to place them on any object.

- Operating System Limitations (privacy and energy)

The Operating System should not be a hindrance because of its device's data privacy and power consumption restrictions.

- Start & stop (with an accelerometer), hibernate

The tag must have a start and stop system with the help of an accelerometer. When it is verified that the object in which the tag is inserted is not being used, the tag will be able to transition to a "hibernate" state that limits its operation to save energy. Likewise, it must transition from this state to "active" as soon as the object starts to move.

- Channel Duality (the server communicating with the tag and the tag with the server)

The tag must be able to communicate with the server as well as receive data from the server.

- Shifting communication protocols (from UDP to TCP and vice-versa)

The tag should be able to transition from mode UDP to mode TCP. In mode UDP, it communicates through a UDP protocol that uses less energy and resources in exchange for less reliability. On the other hand, Mode TCP communicates via a TCP/IP protocol with information reliability at higher energy costs.

- Openness

Users can access shared resources via uniform communication interfaces, which enhances the system's openness.

- Standalone

As a standalone reader and transmitter, the device must not be integrated into another machine and can function independently.

3.2.1. Use Cases

This section describes how the tag interacts with the system. The use cases identify success and failure scenarios and any critical variations.

Case 1: Delayed Post (Figure 9)

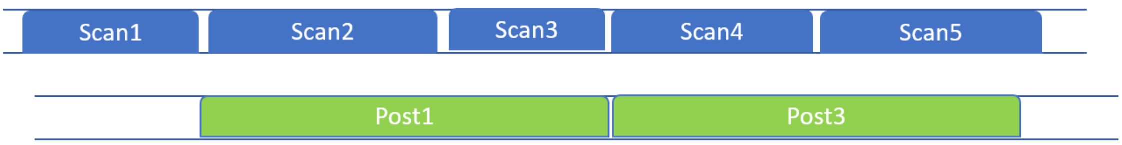


Figure 9 - Case 1.

The post thread takes more time to complete than the scan thread. After a post is completed, information from the last scan will be ready to be retrieved from the following post thread.

Case 2: Delayed Scan (Figure 10)

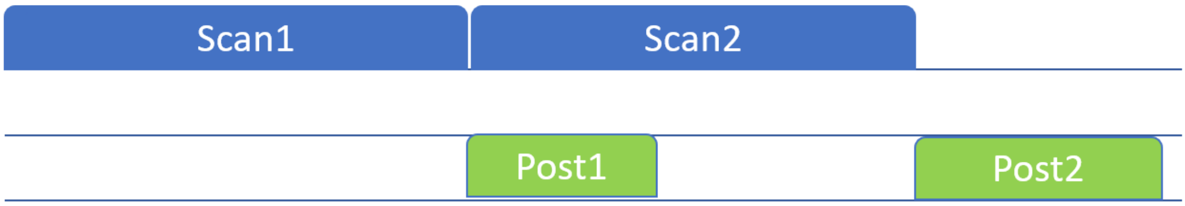


Figure 10 - Case 2.

Since the post thread starts after the scan is complete, it will wait until the scan is finished and the new information is available to be sent to the server. Case 2 is the standard procedure for the program. Ideally, the POSTs would be very fast so as not to lose data from any scans.

Case 3: Unsuccessful Post (Figure 11)

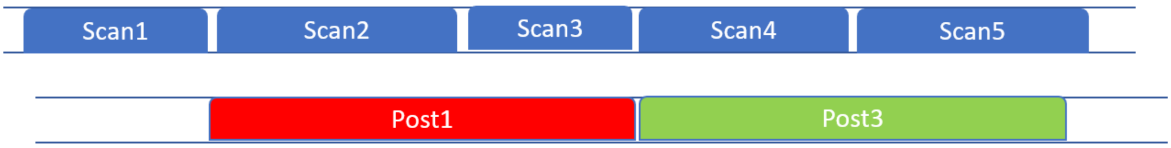


Figure 11 - Case 3.

Here it is explained how the program proceeds if a post is unsuccessful. The post thread will finish with an inadequate response from the server. Then, the information contained in the queue (from the last scan) will be retrieved by the following post thread and will be posted.

Case 4: Unsuccessful Scan (Figure 12)

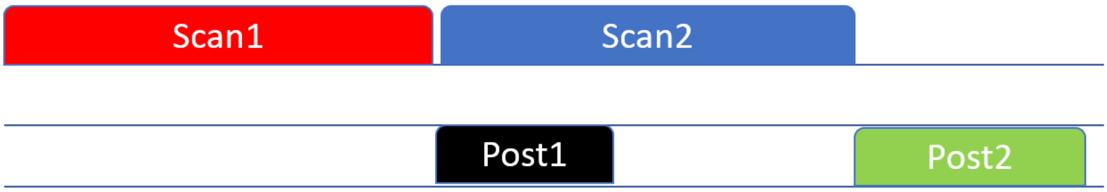


Figure 12 - Case 4.

Over here, how the program will deal with an unsuccessful scan is explained. If a scan is unsuccessful, the queue will be empty, meaning the post must not be performed. The condition states that the post will not be performed if the scan information is blank. Therefore, the post thread will wait for the following scan to provide new data and post it.

3.3. Microcontroller Boards Selection

The tag to be considered should deliver 2-4 meters level accuracy via Wi-Fi and BLE for high-performance location tracking that supports diverse RTLS needs. Featuring low-power consumption, wireless configuration and industrial-grade reliability, deployment-ready hardware can be easily set up, configured and maintained—supporting ranging, real-time tracking and bi-directional communication.

It was decided to build the tag around a microcontroller board to minimise the hardware development effort. At first, many microcontroller boards were considered that could be used as the main component of a tag, bearing in mind that they would later be disregarded based on the essential functional requirements. Table 3 shows the first set of boards that were considered.

Table 3 - Preliminary Research

Name	Wi-Fi	Bluetooth	Price Range
Arduinos	Yes	Yes	Low
Beaglebones	Yes	Yes	High
SparkFuns	Yes	Yes	Medium
ESP32 Modules	Yes	Yes	Low

A table listing all the specifications of each microcontroller board is available in section A.1 Preliminary Research.

After the initial consideration, many more microcontroller boards were explored, and a few previously considered ones were disregarded. That is because of functional requirements. A focus on the size and weight of the boards and more extensive research drove this next section to consider the following, as shown in Table 4.

Table 4 - Outlining Possibilities

Name	Length	Width	Soldering
Arduinos	Medium	Medium	No
SparkFuns	Medium	Medium	No
ESP32 Modules	Small	Small	Yes
ESP32 Development Boards	Medium	Medium	No

Section A.2 Outlining Possibilities displays an extended analysis to get a complete picture.

Finally, a verdict had to be reached regarding functional aspects of the board and extras that would be practical and useful for the work to be developed. These four boards were considered optimal and suitable for the project. The four boards selected for purchasing and worked on are listed in Table 5.

Table 5 - Final Decision

Name	Justification
Beetle	A small-scale electronic device with a great range of benefits such as low power consumption, an enticingly small form factor, a surprising amount of functionality and much computational power
SparkFun Thing Plus	A powerful Wi-Fi and Bluetooth MCU module, scalable and adaptive, easy to use, and utilises the Qwiic Connect System, meaning no soldering or shields are required to connect it.
Arduino Nano RP2040 Connect	A versatile plug & play board with dual-core, connectivity, audio and machine learning capabilities. It also takes less time to compile with the benefits of a full Arduino Cloud support
ESP32 Module	Based on its value for money, wide range of features, small size and relatively low power consumption, it is well suited to the IoT application. A good understanding of soldering is required.

A whole perspective is given in A.3 Final Decision about the microcontroller boards. The first module to be tested at this stage would be the ESP32.

4. System Development

This chapter will explain how to set up the ESP32 and the programming environment. Then, how to model a web server architecture. Afterwards, how to implement the interface communication and perform an architecture-based evaluation. These demonstrations are also consistent with observational, experimental (simulation) and descriptive (scenarios) design evaluation models.

4.1. Software Development Environment

To facilitate the replication of this process by investigators, this section of the document provides instructions for setting up the software development environment for hardware based on the Espressif ESP32 chip (Figure 13). After that, a simple example will show how to use Arduino IDE, namely the menu configuration and how to build and flash software onto an ESP32 board (Maker Tutor, 2018).

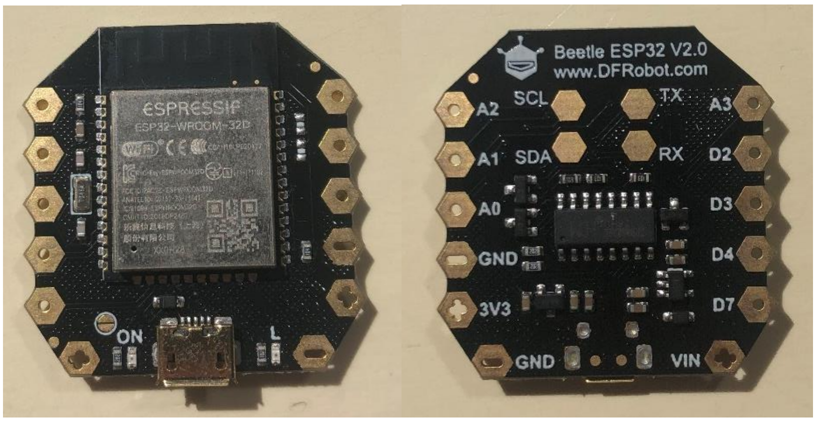


Figure 13 - DFR0575 microcontroller.

1. Setup

Arduino IDE (https://www.arduino.cc/en/software) is a more user-friendly approach[9] (Figure 14).

9 In accordance with the manufacturer's website (DFRobot, 2022a)

Figure 14 - Arduino Download.

2. Adding the Board URL to Arduino IDE

Open Arduino IDE (version 1.8.19).

Navigate to file> Preferences and click the checked button as shown below (Figure 15):

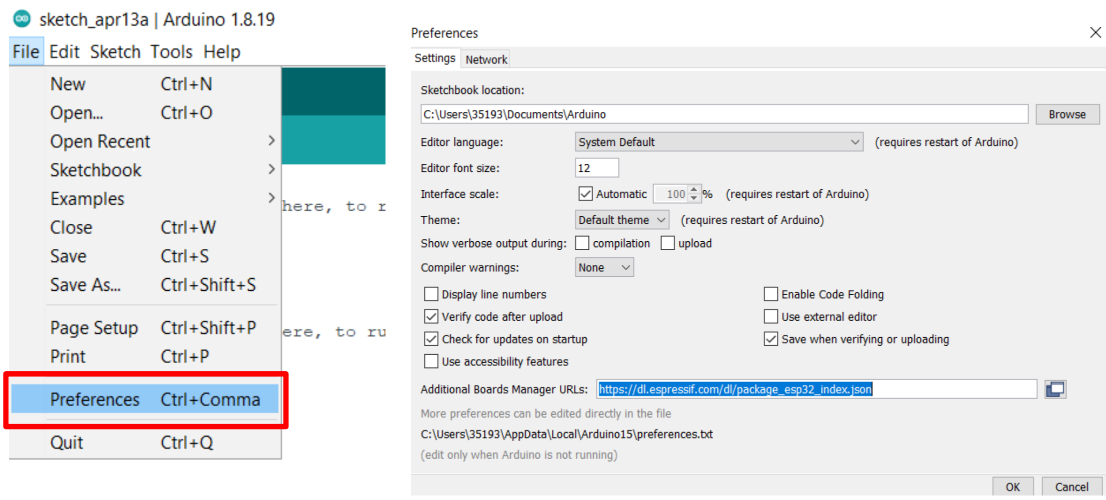


Figure 15 - Arduino Preferences.

In Additional Boards Manager URLs, the following URL must be entered:

- https://dl.espressif.com/dl/package_esp32_index.json[10]

After the input is completed, click OK.

---

[10] Third-party board platforms to be added to the Board Manager (Espressif, 2022)

3. Installing ESP32

After adding the mainboard URL of ESP32 (ESP-WROOM-32) to Arduino IDE (DFRobot, 2022b), it is needed to update the board list and use Arduino IDE to download ESP32 libraries (Arduino, 2022). It is possible to see this in Figure 16.

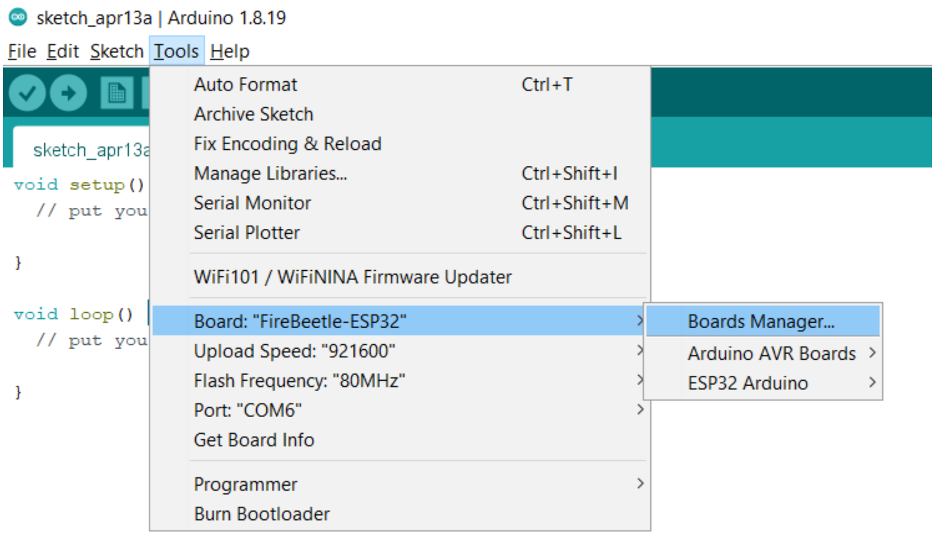


Figure 16 - Arduino Boards Manager.

After opening the development board manager, the board list will automatically be updated. When the update is finished, type esp32, select it and click "Install". The manager will automatically download the relevant libraries (Figure 17).

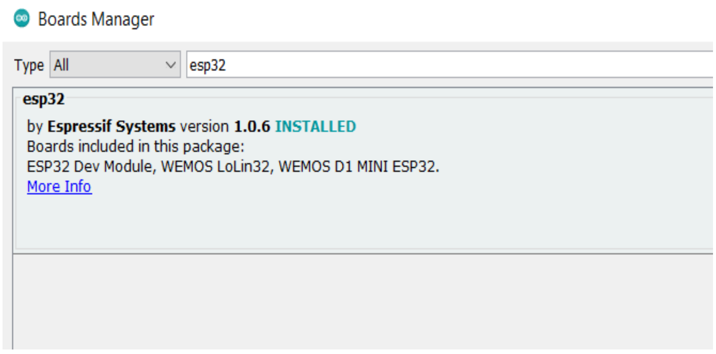


Figure 17 - ESP32 library.

4. Selecting the mainboard

Open tools → board and select **FireBeetle-ESP32** as the mainboard, as shown in the following (Figure 18):

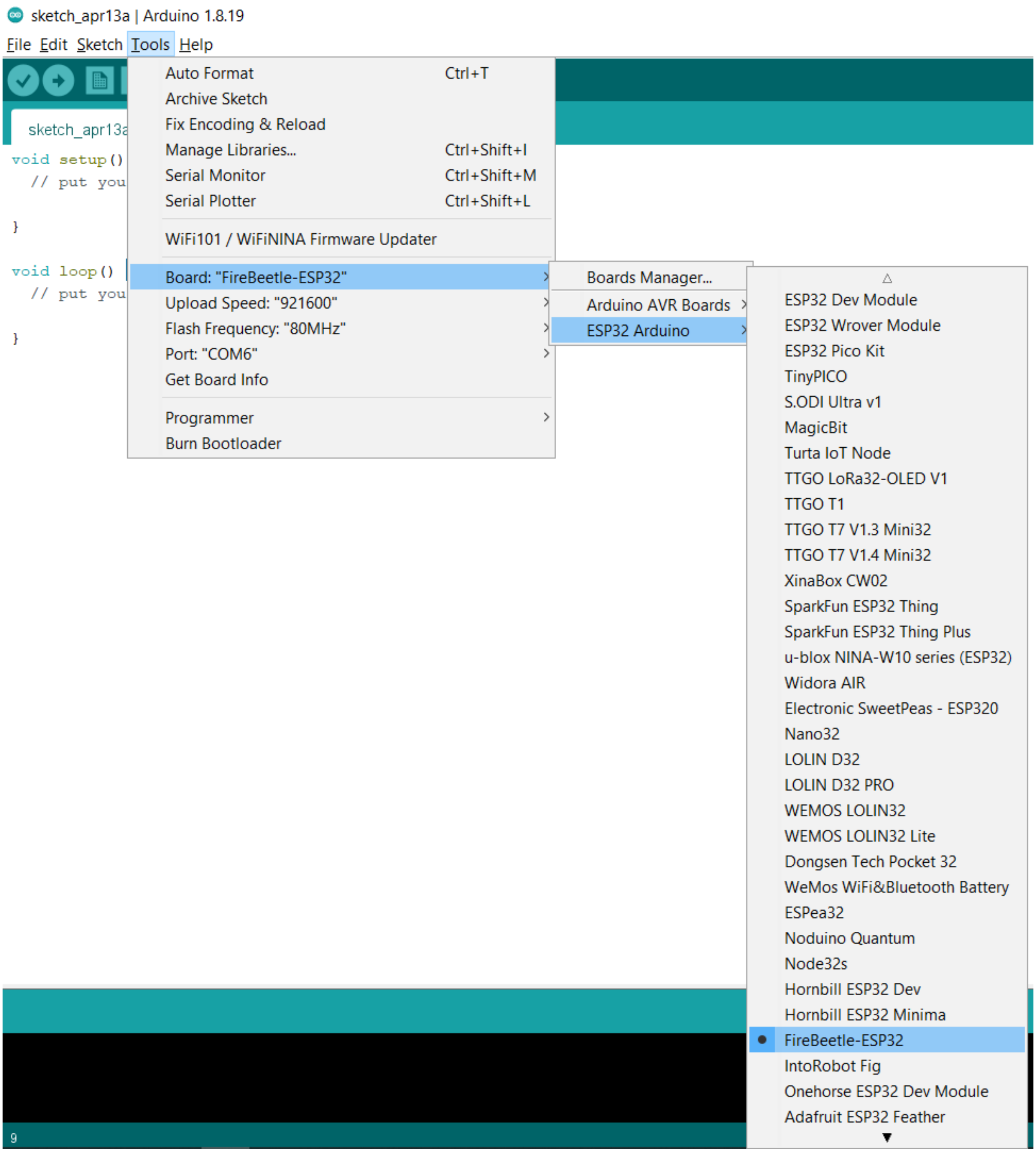


Figure 18 - Step 4.

The board should then be ready for use (Figure 19).

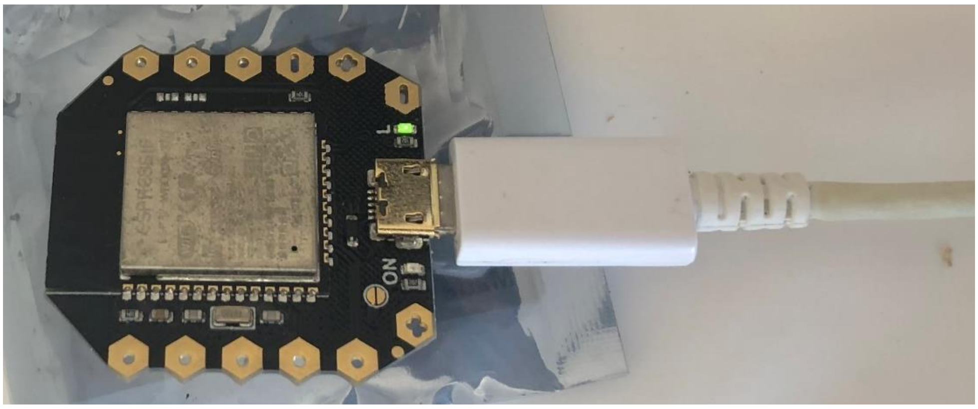


Figure 19 - Board connected.

4.1.1. Partition Schemes

A Partition table defines the flash memory organisation and the different kinds of data stored on each partition (XTronical, 2020). Creating a complex sketch can lead to memory outages, especially when using Wi-Fi or Bluetooth libraries. Errors like this can occur (Figure 20).

```
Sketch too big. Sketch uses 2233498 bytes (113%) of program storage space. Maximum is 1966080 bytes.
```

Figure 20 - Oversized Sketch.

1. **First, it is needed to locate the boards.txt, which is in the directory:**

C:\Users\35193\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.6

2. **Once the file is opened, locate the board that is being used, which, in this case, firebeetle32 (Figure 21):**

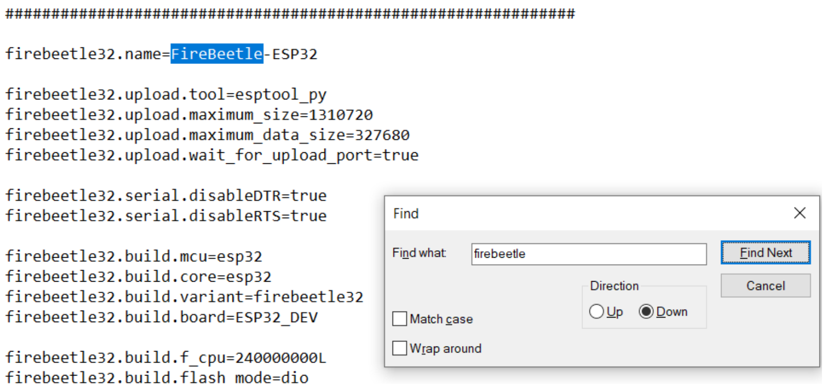


Figure 21 - Firebeetle32 partition schemes.

3. **Add the partition schemes:**

firebeetle32.menu.PartitionScheme.default=Default

firebeetle32.menu.PartitionScheme.default.build.partitions=default

firebeetle32.menu.PartitionScheme.minimal=Minimal (2MB FLASH)

firebeetle32.menu.PartitionScheme.minimal.build.partitions=minimal

firebeetle32.menu.PartitionScheme.no_ota=No OTA (Large APP)

firebeetle32.menu.PartitionScheme.no_ota.build.partitions=no_ota

firebeetle32.menu.PartitionScheme.no_ota.upload.maximum_size=2097152

firebeetle32.menu.PartitionScheme.min_spiffs=Minimal SPIFFS (Large APPS with OTA)

firebeetle32.menu.PartitionScheme.min_spiffs.build.partitions=min_spiffs

firebeetle32.menu.PartitionScheme.min_spiffs.upload.maximum_size=1966080

firebeetle32.menu.PartitionScheme.fatflash=16MFat

firebeetle32.menu.PartitionScheme.fatflash.build.partitions=ffat

4. **Save the file and restart the Arduino IDE.**

5. **The partition scheme tab will be displayed in the IDE (Figure 22):**

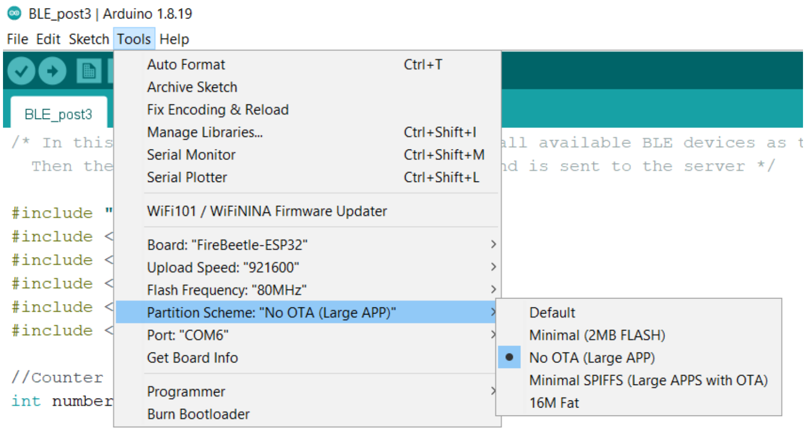


Figure 22 - Arduino partition.

4.1.2. Exception Stack Trace Decoder

It is challenging to pinpoint where a runtime error occurred. A tool must be installed to determine precisely where the problem is (G6EJD - David, 2018). With this Arduino plugin, the user can get a more descriptive explanation of the stack trace.

Installation

- The tool archive can be downloaded from the releases page[11] (Figure 23).

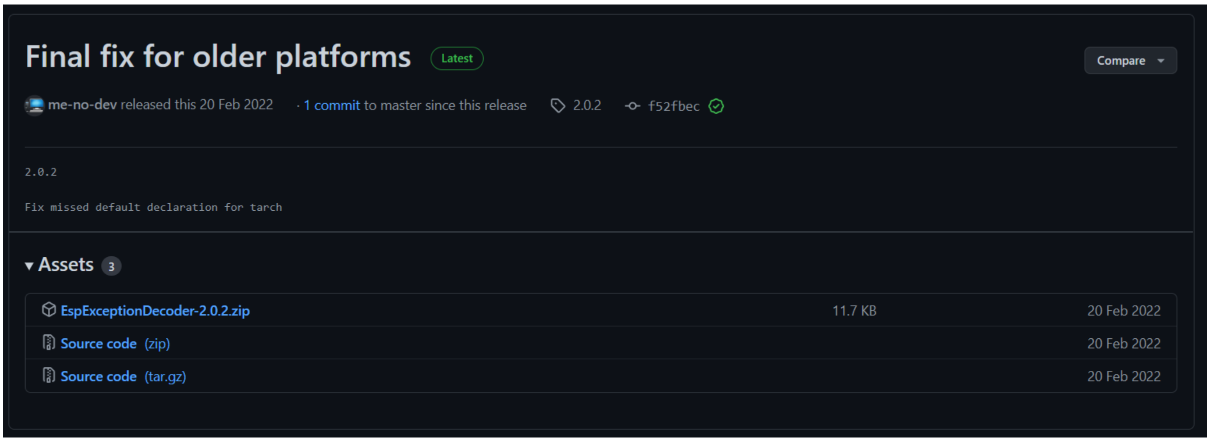


Figure 23 - Releases page.

- The tool must be unpacked into the tools directory C:\Users\35193\Documents\Arduino\tools (Figure 24).

[11] From the operator's official GitHub (Dev, 2016/2022)

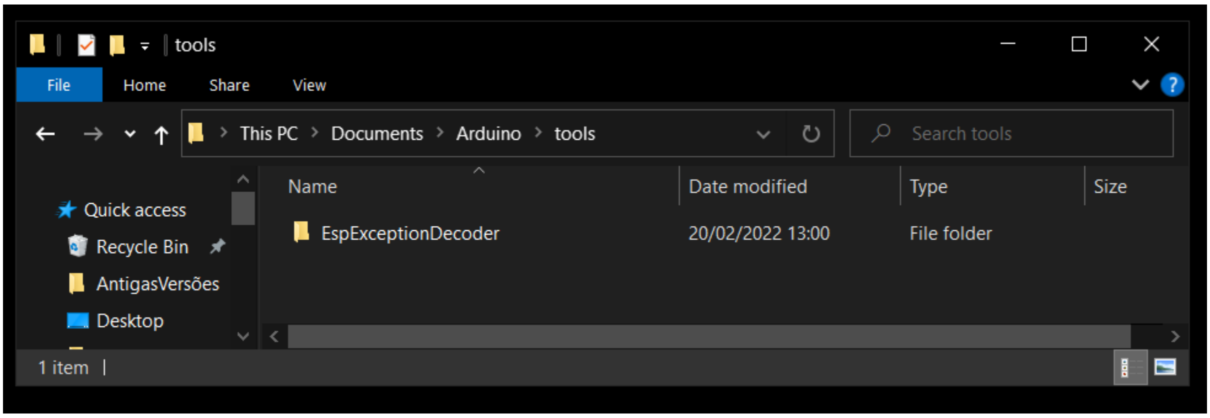


Figure 24 - Tools directory.

- Arduino IDE should be restarted.

Usage

- Compile and upload a sketch.
- Copy the backtrace line when it fails with a runtime error (Figure 25).

```
Guru Meditation Error: Core  1 panic'ed (IntegerDivideByZero). Exception was
Core 1 register dump:
PC      : 0x400d0bff  PS      : 0x00060530  A0      : 0x800d13f0  A1      :
A2      : 0x00000000  A3      : 0x00000032  A4      : 0x00000020  A5      :
A6      : 0x00000000  A7      : 0x00000000  A8      : 0x800d0c11  A9      :
A10     : 0x3ffbfe88  A11     : 0x00000002  A12     : 0x0000000a  A13     :
A14     : 0x00000003  A15     : 0x00000000  SAR     : 0x0000001f  EXCCAUSE:
EXCVADDR: 0x00000000  LBEG    : 0x400014fd  LEND    : 0x4000150d  LCOUNT  :

ELF file SHA256: 0000000000000000

Backtrace: 0x400d0bff:0x3ffb1f90 0x400d13ed:0x3ffb1fb0 0x400860ed:0x3ffb1fd0
```

Figure 25 - Runtime error.

- The result will be displayed in the bottom pane of the exception decoder once the stack trace is pasted into the top pane (Figure 26).

Figure 26 - Exception Decoder.

4.2. Tag Internal Architecture

This section describes the features that exist in the ESP32, the tag software's architecture, and its implementation details. Figure 27 represents the architecture of the software that runs on the tag.

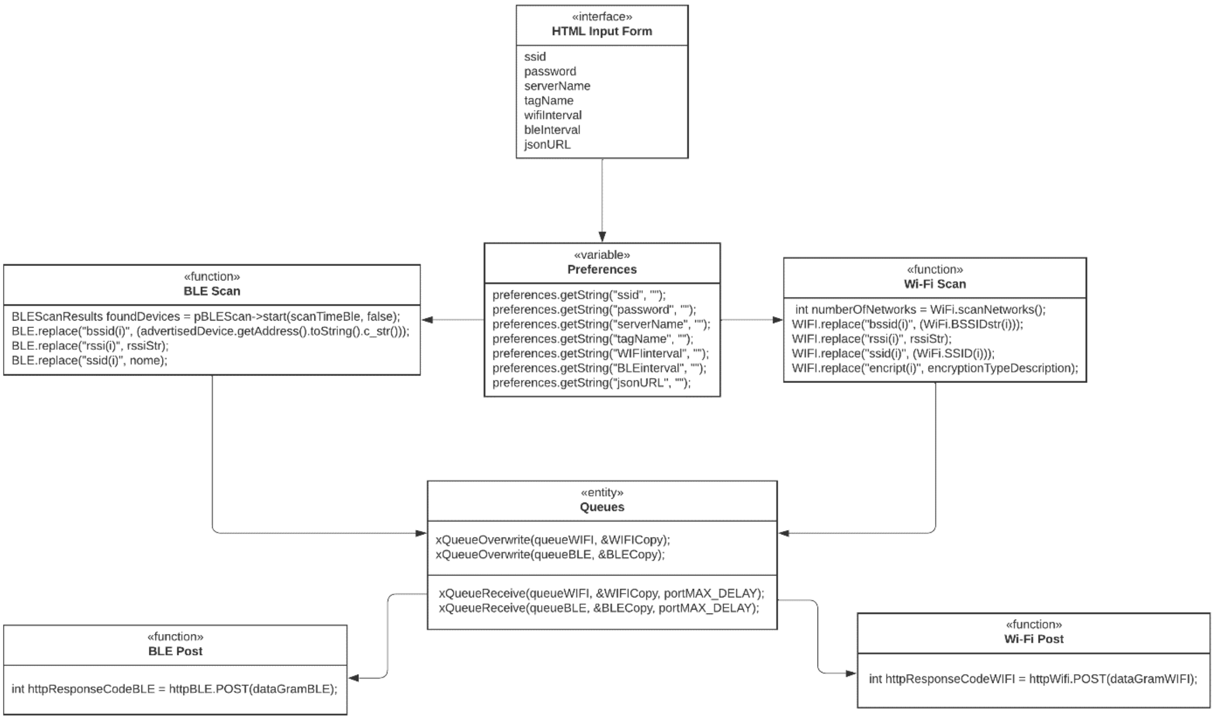


Figure 27 - Tag Internal Architecture.

Figure 27 helps the reader understand how the main software blocks present in the tag are connected. BLE Scan BLE, Wi-Fi Scan, BLE Post and Wi-Fi Post are the focal methods of the software. After withdrawing the parameters from Preferences, scans and posts act independently and repetitively (depending on the interval the user defines in the HTML Input Form). Two queues (one for BLE and another for Wi-Fi) hold the scan data as it waits for its turn to be posted. The implementation of the software written in detail is provided next.

- Debug

A serial.print statement used for debugging is data coming out of the serial Port. The TX pin on the chip of the ESP32 connects to the USB port and transmits messages in both directions as the programme requires. The program can significantly bloat if there are too many serial.print statements taking up memory. Serial.print strings use SRAM (runtime memory) and slow down the sketch because the UART that transmits all the messages is slower than the processor.

A pre-processor is incorporated that defines whether or not debugging is on so that memory can be saved (Ralph S Bacon, 2021).

```
#define DEBUG 1

#if DEBUG == 1
#define debug(x) Serial.print(x)
#define debugln(x) Serial.println(x)
#define debugf(x , y) Serial.printf(x , y)
#else
#define debug(x)
#define debugln(x)
#define debugf(x , y)
#endif
```

- Access Point

The ESP32 has a Wi-Fi chip that can generate its network. This configuration is called AP (Access Point) mode. The ESP32 can function as a Wi-Fi station and an access point. The ESP32 is connected to a Wi-Fi network (STATION mode) and has a Wi-Fi access point activated (AP mode). This mode is called WIFI_AP_STA (Figure 28[12]). Each mode has a different IP address because their interface is in a separate network (uPesy, 2022).

[12] This image was retrieved from the online article “Create a Wi-Fi access point with an ESP32” (uPesy, 2022)

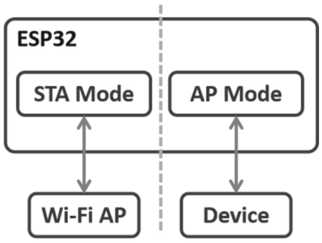


Figure 28 - ESP32 WIFI_AP_STA mode.

One possible application is to connect the local Wi-Fi network created by the ESP32 in Soft AP mode to the Internet, as shown in Figure 29[13].

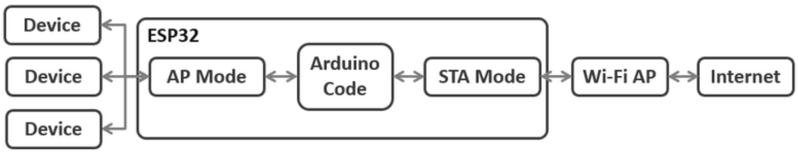


Figure 29 - Architectural Example.

A mixture of Station and AP management is implemented in the Arduino code. The code is straightforward when already using the two modes separately:

```
WiFi.mode(WIFI_AP_STA);
```

- Input Data on HTML Form Web Server

An asynchronous web server was built using the ESPAsyncWebServer library that displays seven input fields to pass values for updating the values of a few parameters. Each time a new value is submitted, it is stored in a SPIFFS file. This web page includes a placeholder to display the current value (Random Nerd Tutorials, 2019b).

To build the asynchronous web server, installing these libraries is fundamental: <ESPAsyncWebServer>[14], <AsyncTCP>[15], <SPIFFS.H> and <WiFi.h>. The first two libraries are unavailable to install through the Arduino Library Manager. These steps must be followed: Sketch > Include Library > Add .zip Library and select the libraries downloaded.

```
#include <SPIFFS.h>
#include <WiFi.h>
```

13 This image was retrieved from the online article "Create a Wi-Fi access point with an ESP32" (uPesy, 2022).

14 ESPAsyncWebServer (Dev, 2015/2022).

15 AsyncTCP (Dev, 2017/2022).

```
#include <ESPAsyncWebServer.h>
#include <AsyncTCP.h>
```

When submitting the values, a window opens saying the value was saved to SPIFFS. For that, the JavaScript function must be added. In this case, submitMessage() pops an alert message saying the value was saved to SPIFFS. After that pop-up, it reloads the web page to display the current values.

```
<script>
    function submitMessage() {
      alert("Saved value to ESP32 SPIFFS");
      setTimeout(function(){ document.location.reload(false); }, 500);
    }
</script>
```

Form for the first input (ssid). Input field names display (%inputSsid%) placeholders that the current inputSsid value will replace.

```
<form action="/get" target="hidden-form">
    inputSsid (current value %inputSsid%): <input type="text" name="inputSsid">
    <input type="submit" value="Update" onclick="submitMessage()">
</form>
```

The target attribute and a <iframe> remain on the same page after submitting the form.

```
<iframe style="display:none" name="hidden-form"></iframe>
```

Then, some functions to read and write from SPIFFS are coded. A file's content is read using readFile().

```
String readFile(fs::FS &fs, const char * path) {
  File file = fs.open(path, "r");
  if (!file || file.isDirectory()) {
    return String();
  }
  String fileContent;
  while (file.available()) {
    fileContent += String((char)file.read());
  }
  file.close();
  return fileContent;
}
```

The writeFile() writes content to a file:

```
void writeFile(fs::FS &fs, const char * path, const char * message) {
  File file = fs.open(path, "w");
  if (!file) {
    return;
  }
  if (file.print(message)) {
  } else {
  }
  file.close();
}
```

The processor() replaces placeholders in the HTML text with actual values saved on SPIFFS.

```
String processor(const String& var) {
  if (var == "inputSsid") {
    return readFile(SPIFFS, "/inputSsid.txt");
  }
  else if (var == "inputPassw") {
    return readFile(SPIFFS, "/inputPassw.txt");
  }
  else if (var == "inputServer") {
    return readFile(SPIFFS, "/inputServer.txt");
  }
  else if (var == "inputTag") {
    return readFile(SPIFFS, "/inputTag.txt");
  }
  else if (var == "inputWifi") {
    return readFile(SPIFFS, "/inputWifi.txt");
  }
  else if (var == "inputBle") {
    return readFile(SPIFFS, "/inputBle.txt");
  }
  else if (var == "inputUrl") {
    return readFile(SPIFFS, "/inputUrl.txt");
  }
  else if (var == "inputState") {
    return readFile(SPIFFS, "/inputState.txt");
  }
  return String();
}
```

The HTML page is displayed to the client when accessing the route URL, which is stored in the index_html variable.

```
  server.on("/", HTTP_GET, [](AsyncWebServerRequest * request) {
    request->send_P(200, "text/html", index_html, processor);
  });
```

Then, handling what happens when receiving a request on the /get routes is programmed.

```
  server.on("/get", HTTP_GET, [] (AsyncWebServerRequest * request) {
```

When the request contains input (i.e. PARAM_), the inputMessage variable is set to the value submitted on the input form. Then, that value is saved to SPIFFS.

```
    if (request->hasParam(PARAM_SSID)) {
      inputMessage = request->getParam(PARAM_SSID)->value();
      writeFile(SPIFFS, "/inputSsid.txt", inputMessage.c_str());
    }
```

If requesting an invalid URL, the notFound() function is called, defined at the beginning of the sketch.

```
void notFound(AsyncWebServerRequest *request) {
  request->send(404, "text/plain", "Not found");
}
```

Finally, the server.begin() function is called to handle clients.

```
  server.begin();
```

- Saving Data Permanently using Preferences Library

Preferences allow saving data like network credentials and threshold values in flash memory that endures shutdowns or resets (Random Nerd Tutorials, 2021b).

```
#include <Preferences.h>
```

The data saved using preferences is structured like this:

```
    credentials {
  ssid: "ZON-C510"
  password: "Boas1234"
```

```
  serverName: "http://ils.dsi.uminho.pt/ar-ware/S02/i2a/i2aSamples.php"
  tagName: "tagPedro"
  WIFIinterval: "5000"
  BLEinterval: "7000"
  jsonURL: "192.168.0.108"
  state: "0"
}
```

An instance of the Preferences library must be initiated.

```
Preferences preferences;
```

The begin() method opens a “storage space” with a defined namespace. The false argument will be used for the read/write mode.

```
  preferences.begin("credentials", false);
```

Then, a key called ssid is created that saves the SSID value (ssid variable) – using the putString() method.

```
      preferences.putString("ssid", inputMessage.c_str());
```

The getString() method is used for obtaining the SSID. The first argument is the key name used to save the variables.

```
  ssid = preferences.getString("ssid", "");
```

The end() method is used to close preferences.

```
      preferences.end();
```

- FreeRTOS Queues[16]

For inter-task communication, queues provide a safe way to send messages from one task to another. Inserting pointers to data as elements of the queue is possible, which is helpful since the messages to exchange are big. Since the pointer is copied to the queue, not the message itself, it is unnecessary to ensure

16 Queue API specifications can be found in the following link (FreeRTOS, 2022).

it does not change. In this scenario, elements will be overwritten and consumed from a queue.
Initially, it is necessary to declare a global variable QueueHandle_t, the type to reference a FreeRTOS queue (techtutorialsx, 2017).

```
QueueHandle_t queueWIFI;
QueueHandle_t queueBLE;
```

Next, the queue will be created with a call to the xQueueCreate function. This function receives as the first input the maximum number of elements the queue can hold at a given time and, as the second argument, the size (in bytes) of each component. Queue elements should have equal lengths.
So, two queues that can hold a maximum of 1 element will be created. Upon successful execution, the xQueueCreate function will return a handle for the queue, which is of type QueueHandle_t, the same as the variable declared globally. If a problem with the queue allocation occurs, it will return NULL. Thus, a NULL check is done on the setup function to warn if a situation arises.

```
  queueWIFI = xQueueCreate( 1, 20 );
  queueBLE = xQueueCreate( 1, 20 );

  if (queueWIFI == NULL) {
    debugln("Error creating the queue");
    debugln("------------------------------------------------------------");
  }

  if (queueBLE == NULL) {
    debugln("Error creating the queue");
    debugln("------------------------------------------------------------");
  }
```

Overwriting the values in the queue for later consumption will then start. The xQueueOverwrite function is called for overwriting an item, which overwrites an element in the queue.
Firstly, it receives a queue handle, a global variable declared and assigned with the result of queuing. An item pointer is passed as second input (passing a pointer means copying the item).
Given how the program was built, trying to insert when the queue is full is possible since it will be overwritten, so the program will not block.

```
  xQueueOverwrite(queueBLE, &adicionarBLECopy);
  xQueueOverwrite(queueWIFI, &adicionarWIFICopy);
```

A pointer is being passed to the same variable, but since its current value will be copied, changing it to a new value in each iteration is no problem.

For consuming an item, the xQueueReceive function needs to be called. Inputs include the queue handle, a buffer pointer and the number of ticks to wait when the queue is empty.

The queue will be cleared once the item has been consumed.

```
  xQueueReceive(queueBLE, &adicionarBLECopy, portMAX_DELAY);
  xQueueReceive(queueWIFI, &adicionarWIFICopy, portMAX_DELAY);
```

- HTTP POST: URL Encoded

HTTP operates as a request-response protocol between the client and the server. POST sends data to the server to create resources (Random Nerd Tutorials, 2020).

The following libraries need to be installed: <WiFi.h> and <HTTPClient.h>, which are in the Arduino IDE Library Manager.

```
#include <HTTPClient.h>
#include <WiFi.h>
```

```
// The Domain name with URL path or IP address with path
        httpBle.begin(clientBle, serverName);
```

```
// Specify content-type header
      httpBle.addHeader("Cache-Control", "no-cache");
           httpBle.addHeader("Content-Type",   "application/x-www-form-urlencoded",
"Content-Length", tamanhoBLE);
      httpBle.addHeader("Accept", "*/*");
      httpBle.addHeader("Accept-Encoding", "gzip, deflate, br");
      httpBle.addHeader("Connection", "keep-alive");
```

```
// Concatenate both parts to originate the final datagram to be sent
    String dataGramaBLEFinal = dataGramaBLE + adicionarBLECopy;
```

```
// Send HTTP POST request
      int httpResponseCodeBLE = httpBle.POST(dataGramaBLEFinal);
```

- Wi-Fi and BLE scans

The ESP32 scans Wi-Fi networks and BLE devices within its vicinity (Random Nerd Tutorials, 2021a). Each scan will amount information to be sent in the JSON object. Firstly, including the necessary libraries in the code is required.

```
#include <WiFi.h>
#include <BLEAdvertisedDevice.h>
#include <BLEDevice.h>
#include <BLEScan.h>
#include <BLEUtils.h>
```

Then, it is paramount to start scanning to access information. The first part is from the Wi-Fi scan, and the second is from the BLE scan.

```
  int numberOfNetworks = WiFi.scanNetworks();
```

```
  BLEScanResults foundDevices = pBLEScan->start(scanTimeBle, false);
```

After scanning, the parameters of each network are retrieved (Random Nerd Tutorials, 2019a). These parameters are included in the JSON object by replacing the preset values.

```
//The preset value of bssid is replaced by the bssid of the network number i found
    adicionarWIFI.replace("bssid(i)", (WiFi.BSSIDstr(i)));
    String rssiStr = String (WiFi.RSSI(i));
//The preset value of rssi is replaced by the rssi of the network number i found
    adicionarWIFI.replace("rssi(i)", rssiStr);
//The preset value of ssid is replaced by the ssid of the network number i found
    adicionarWIFI.replace("ssid(i)", (WiFi.SSID(i)));
    String encryptionTypeDescription =
translateEncryptionType(WiFi.encryptionType(i));
//The preset value of encryption is replaced by the encryption of the network number
i found
    adicionarWIFI.replace("encript(i)", encryptionTypeDescription);
```

```
//The preset value of bssid is replaced by the bssid of the device found
      adicionarBLE.replace("bssid(i)",
(advertisedDevice.getAddress().toString().c_str()));
      String rssiStr = String (advertisedDevice.getRSSI());
//The preset value of rssi is replaced by the rssi of the device found
      adicionarBLE.replace("rssi(i)", rssiStr);
```

```
      String nome = advertisedDevice.getName().c_str();
//The preset value of ssid is replaced by the ssid of the device found
      adicionarBLE.replace("ssid(i)", nome);
```

- Task Watchdog

ESP32 is built around the FreeRTOS operating system using Arduino IDE, which assigns one IDLE task per core. Watchdog timers also monitor these tasks, meaning that the chip is reset if they are not executed for a certain period. A significant amount of FreeRTOS "household" work is done in the IDLE tasks, so they must be given sufficient time. In addition, IDLE tasks have the lowest priority, so if any other tasks are running with a higher priority, they take precedence over IDLE tasks (Sam, 2021).

Tasks with higher priority should be short enough to prevent the watchdog from being triggered. If this is not feasible, pauses must be inserted in sufficient intervals by calling delays. Upon completing the currently running task, FreeRTOS will start another task. If no other higher-priority tasks are waiting, the IDLE task will finally be allowed to run. No code should preoccupy the CPU 100% of the time for longer than the watchdog timeout period (fear, 2021).

```
  delay(100);
```

Keeping the Arduino loop() empty will trigger the watchdog timer because it is not really "empty". After all, the loop() function is internally wrapped in an infinite loop, so the CPU consumption would skyrocket without doing anything useful (Tarmo, 2021).

```
void loop() {
//The task which runs setup() and loop() is created on core 1 with priority 1.
  vTaskDelete(NULL);
// to delete the task and free its resources because it is not planned to be used.
}
```

- Dual Core Multithreading

Two Xtensa 32-bit LX6 microprocessors are integrated into the ESP32, making it a dual-core device, as illustrated in Figure 30[17]. When running code on Arduino IDE, it runs on core 1. This section will explain how to run code on the ESP32 using both cores by creating tasks and using threads. Running pieces of code simultaneously on both cores is possible, making the ESP32 a multitasker (Random Nerd Tutorials, 2018).

---

[17] This image was retrieved from the online article "ESP32 Dual Core with Arduino IDE" (Random Nerd Tutorials, 2018).

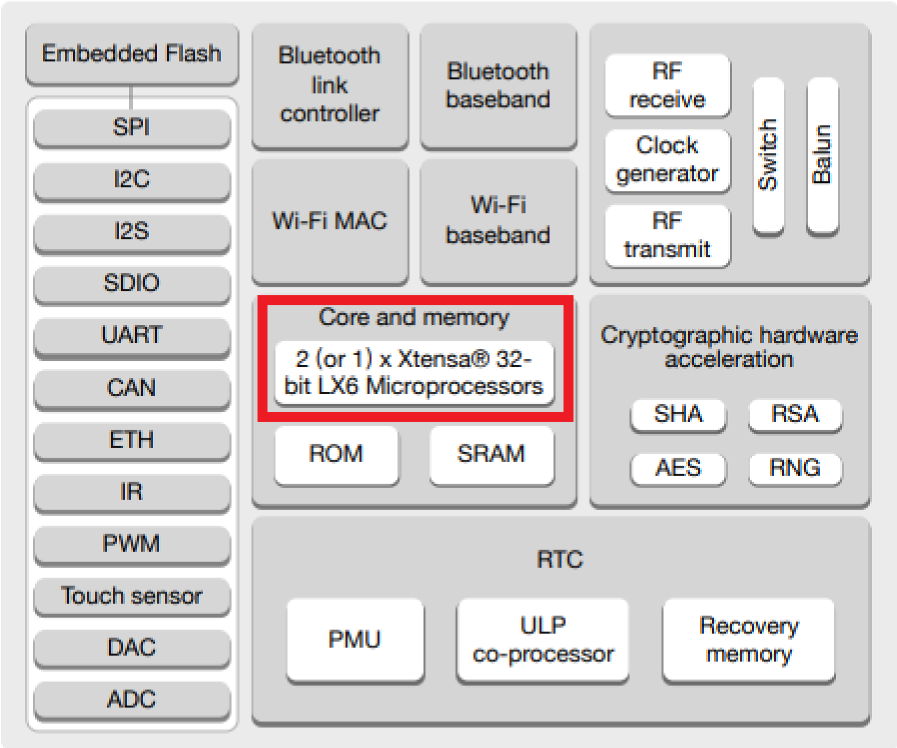


Figure 30 - ESP32 Specifications.

The FreeRTOS operating system supports the parallel processing of independent tasks.

An assignment of specific parts of code to a particular core is accomplished by creating tasks. When creating a task, it can be chosen in which core it will run and its priority. Processors prioritize tasks based on priority values; zero is the lowest priority, which means they run tasks of higher priority first.

Five tasks running on different cores will be created: Tasks 1, 3 and 5 run on core 1; Tasks 2 and 4 run on core 0.

To handle all tasks, the code begins by creating a task handle for each task.

```
TaskHandle_t Task1, Task2, Task3, Task4, Task5;
```

Then, the tasks will be created using the xTaskCreatePinnedToCore() function:

```
  /* create Mutex */
  /*Syntax for assigning a task to a core:
    xTaskCreatePinnedToCore(
                    TaskFunc,      // Function to implement the task
                    "TaskLabel",   // Name of the task
```

```
                    10000,        // Stack size in bytes
                    NULL,         // Task input parameter
                    0,            // Priority of the task
                    &TaskHandle,  // Task handle.
                    TaskCore);    // Core where the task should run
  */

  xTaskCreatePinnedToCore(    codeForTask1,    "serverForm",    8000,    NULL,
 1,    &Task1,    1);
  delay(100);
  xTaskCreatePinnedToCore(    codeForTask2,    "scanBLE",    8000,    NULL,    1,
   &Task2,    0);
  delay(100);
  xTaskCreatePinnedToCore(    codeForTask3,    "scanWIFI",    8000,    NULL,    1
,    &Task3,    1);
  delay(100);
  xTaskCreatePinnedToCore(    codeForTask4,    "postBLE",    8000,    NULL,    1,
   &Task4,    0);
  delay(100);
  xTaskCreatePinnedToCore(    codeForTask5,    "postWIFI",    8000,    NULL,    1,
   &Task5,    1);
  delay(100);
```

The tasks will be implemented with the TaskXcode() function. So that function needs to be created later in the code. All the tasks have the same priority.

After creating the tasks, the functions that will execute those tasks needs to be created.

```
void codeForTask4( void * parameter ) {
  for (;;) {
    if (stateCounter > 0) {
      postBLE();
      delay(100);
    }
    else {
      delay(100);
    }
  }
}
```

4.3. Interface Communication

The i2a interface is intended for communication with tags. From there, it receives Wi-Fi and BLE radio samples (fingerprints) and status messages from the tags. (The i2a interface is intended for communication with tags. It provides a service to receive Wi-Fi and BLE samples and estimate their position based on fingerprinting)

The interface specification for receiving radio samples is the following:

http://ils.dsi.uminho.pt/ar-ware/S02/i2a/i2aSamples.php[18]

The requests are of the POST type, and only the scanData variable is sent in the request body. A scanData variable is a JSON object whose structure depends on the tag reading performed. The semantics of the fields used in these JSON structures are described in the following table (Table 6):

Table 6 - JSON Structure

Field	Description
tagName	String with a name assigned to the tag
tagBSSID	MAC address of the tag's Wi-Fi interface (it uniquely identifies the tag)
tagNetwork	The Wi-Fi network the tag is connected to
dataType	Type of data being sent. There are two possible values: BLE and Wi-Fi
scanMode	"auto" – means the tag is in automatic mode, where it performs a scan every x seconds "manual" – means that the tag is in manual mode. That is, it only reads the radio environment when the user asks (press the physical button on the tag)
BLEData	List of what was collected from the BLE radio interface.
WiFiData	List of what was collected from the Wi-Fi radio interface.
bssid	Address of a BLE device detected in the vicinity of the tag or address of a Wi-Fi AP detected in the vicinity of the tag

18 Receiver Interface (ILS, 2022b)

Field	Description
rssi	The signal strength of the BLE device detected or the signal strength of a Wi-Fi AP detected in the vicinity of the tag
name	Name of the detected BLE device
ssid	Wi-Fi network name
encrypt	Type of encryption used on the Wi-Fi network

At each POST request to send a sample, the i2aSamples interface responds with a 200 OK. If the request body is empty, the string “No data on POST” will be sent.

5. Evaluation

This chapter intends to define how this project was evaluated, describing the implementations referred to in previous chapters and how they were validated to solve the problems exposed. The results of this assessment are incorporated into the annexe section of this document.

5.1. Functional Evaluation

This section will comprise the development tests and functionality tests.

5.1.1. Testing with Postman

When importing a JSON file as a collection with Postman[19], an example of a POST type request is obtained. Using this example, the communication with the interface can be tested.

The request arrives at the web server but must be well formatted. The server returns a “200 OK” HTTP Status Code and an empty body text (Figure 31).

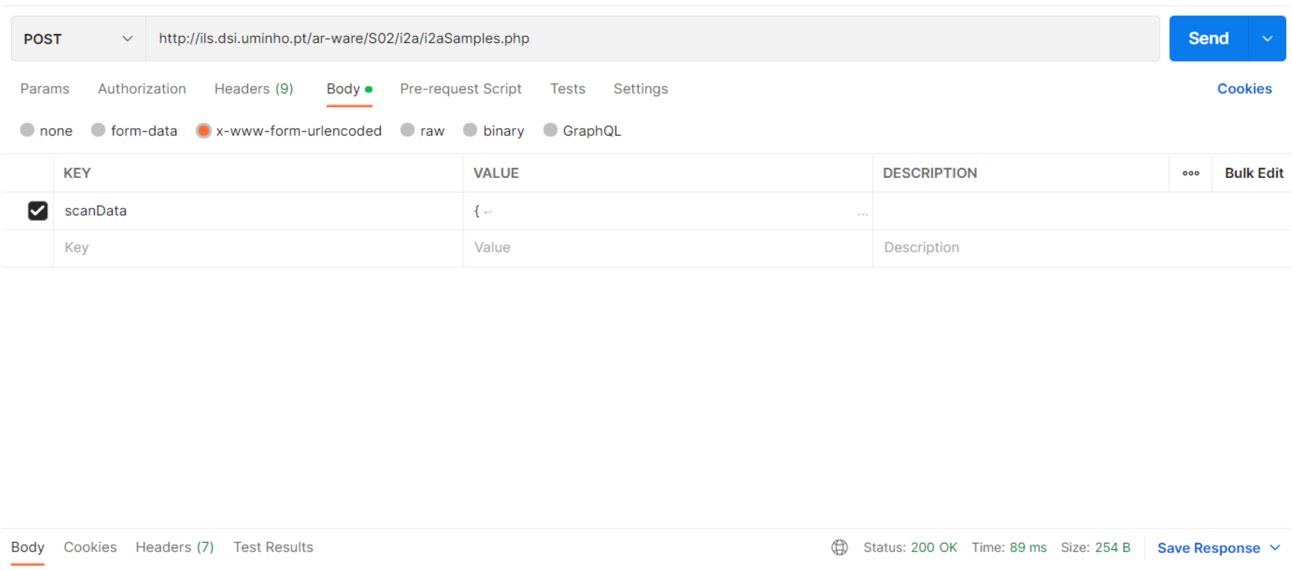


Figure 31 - 200 OK Status Code.

[19] API platform for building and using APIs (Postman, 2022)

View of the web server's registrations (Figure 32).

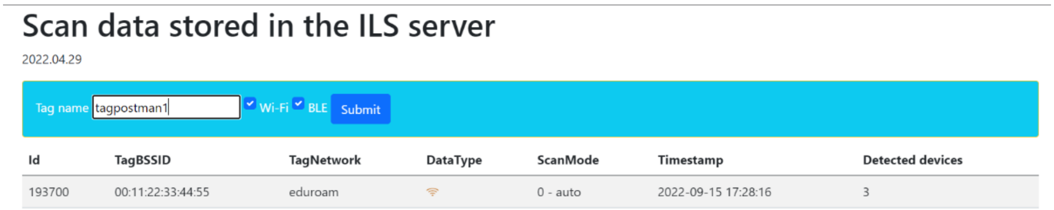


Figure 32 - Server Log.

The system should display a warning if it does not find the data in the requested location, resulting in a "No data on post" error (Figure 33). This situation typically occurs when the JSON is not sent in a variable called "scanData".

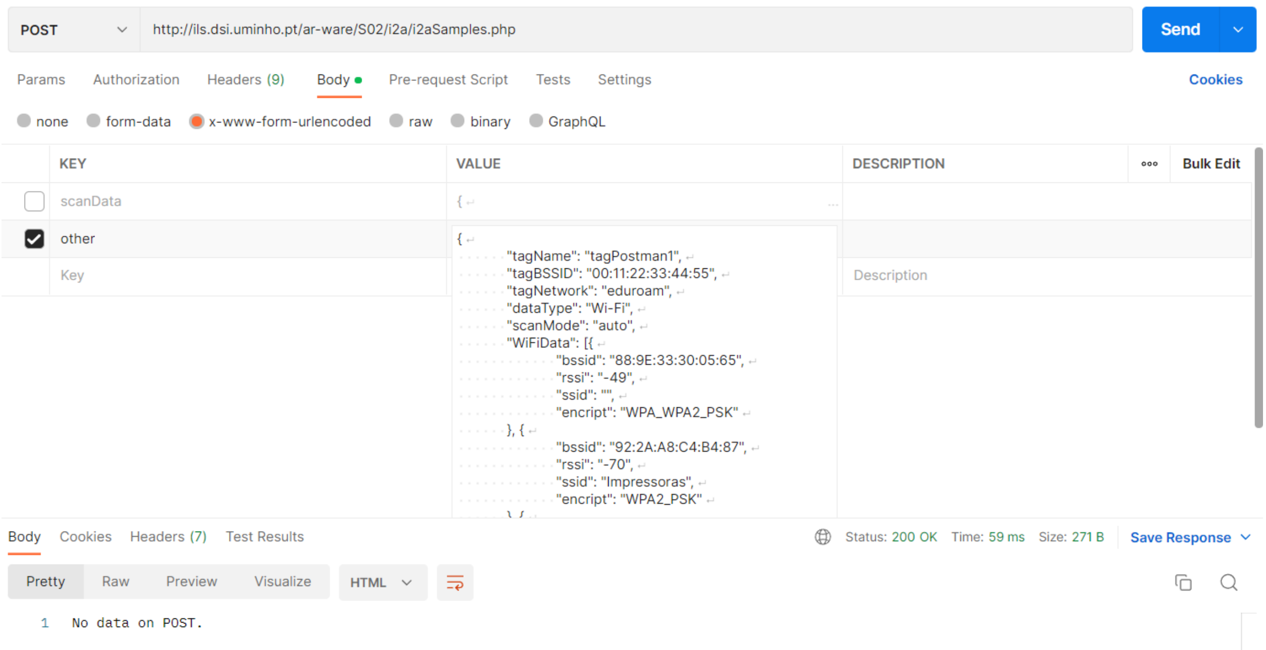


Figure 33 - No data on Post error.

5.1.2. Negative Response Codes

It is possible to determine whether an HTTP request has been completed successfully by looking at the response status code. Nevertheless, is there a meaning to negative response codes?

After implementing the project and verifying everything was working correctly, it was observed that the logs response occasionally is a negative code rather than a 200 OK.

For some Posts, an error arose [20] A negative response code is received, which could be -1 or -11[21] (Figure 34), resulting in the entire process getting terminated, so further investigation was required to handle this issue.

```
-> ############# ##############
-> c: 13041
-> d: 13041
-> HTTP Response code: -1
-> e: 13041
->
-> ###########################
-> scanData= { "tagName": "tag
-> ###########################
-> a: 13325
-> b: 13326
-> Number of networks found: 2
-> ############ ###############
-> c: 20041
-> d: 20041
-> HTTP Response code: -1
-> e: 20041
->
-> ###########################
-> scanData= { "tagName": "tag
-> ###########################
-> a: 20321
-> b: 20322
-> Number of networks found: 2
-> ########### ###############
-> Number of networks found: 2
-> ########### ###############
-> Number of networks found: 2
-> ########### ###############
-> c: 38754
-> d: 38754
-> HTTP Response code: -11
-> e: 38754
->
```

Figure 34 - Negative HTTP Response Codes.

[20] Negative Response Codes (Polykanine, 2021)
[21] Failing Requests (Wolph, 2018)

What could -1 mean? Should it be handled differently from -11?

There is a slight difference between the two, but it is unnecessary to prevent cookies from getting set in either case. It returns -1 if the response is not valid HTTP. It seems that the -11 response times out without responding due to the default time-out of 5000ms (Figure 35).

```
#define HTTPCLIENT_DEFAULT_TCP_TIMEOUT (5000)

/// HTTP client errors
#define HTTPC_ERROR_CONNECTION_REFUSED  (-1)
#define HTTPC_ERROR_SEND_HEADER_FAILED  (-2)
#define HTTPC_ERROR_SEND_PAYLOAD_FAILED (-3)
#define HTTPC_ERROR_NOT_CONNECTED       (-4)
#define HTTPC_ERROR_CONNECTION_LOST     (-5)
#define HTTPC_ERROR_NO_STREAM           (-6)
#define HTTPC_ERROR_NO_HTTP_SERVER      (-7)
#define HTTPC_ERROR_TOO_LESS_RAM        (-8)
#define HTTPC_ERROR_ENCODING            (-9)
#define HTTPC_ERROR_STREAM_WRITE        (-10)
#define HTTPC_ERROR_READ_TIMEOUT        (-11)
```

Figure 35 - Negative Response Codes.

As part of the effort to minimise the number of negative response codes obtained, it was determined that the problem was with resource competition during the implementation of the threads. For the posts to be successful, the implementation of the threads had to be in sync with their release.

5.1.3. Prototype Testing

The process of testing the prototype to validate the development is depicted in this section. The goal is to test that what was built meets users' needs and expectations.

At the program's start, the ESP32 acts as an Acess Point and can be remotely accessed by other devices. The user connects to the ESP32 via Wi-Fi using his cellphone (Figure 36).

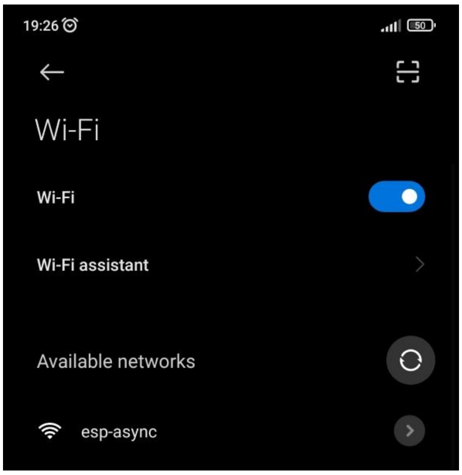


Figure 36 - ESP32 Wi-Fi Access Point.

The HTML webserver form is displayed when accessing the Router address illustrated at the bottom right of Figure 37.

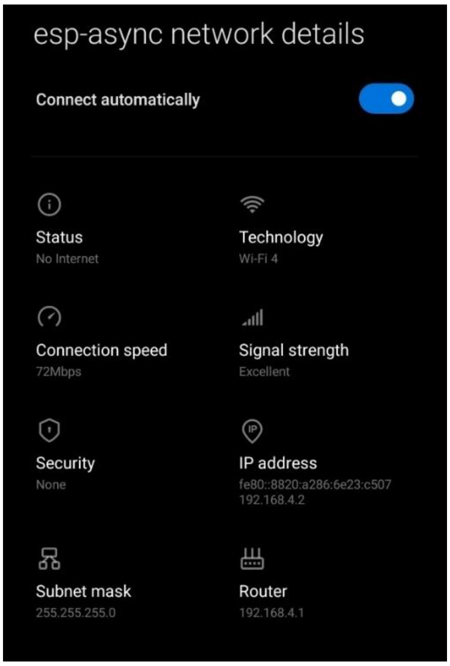


Figure 37 - Access Point Network Details.

When accessing the IP address while connected to the ESP32, the user has a form (Figure 38) where he can change a set of varials such as the information for the scans, namely the Interval scans for Wi-Fi and BLE, as well as the tag Name.

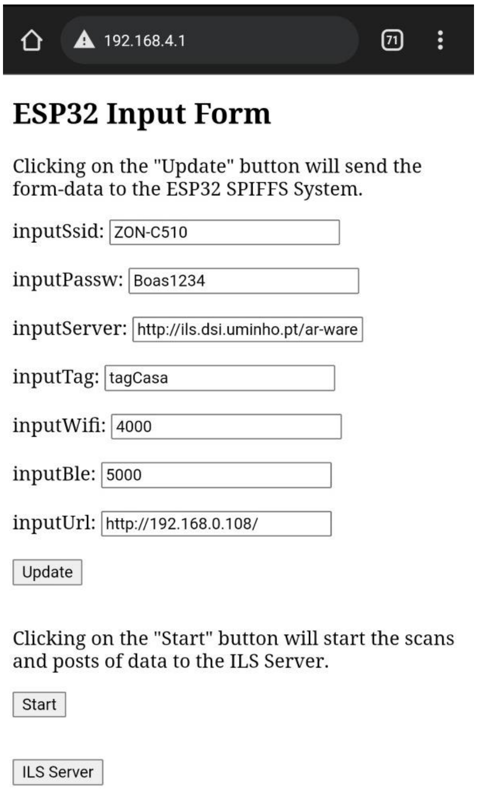


Figure 38 - ESP32 Input Form.

After pressing Start or waiting for one minute, the Tag will start to scan Wi-Fi networks and BLE devices and post the information on the ILS Server interface. The user can access the interface and see the posted scan information, as depicted in Figure 39.

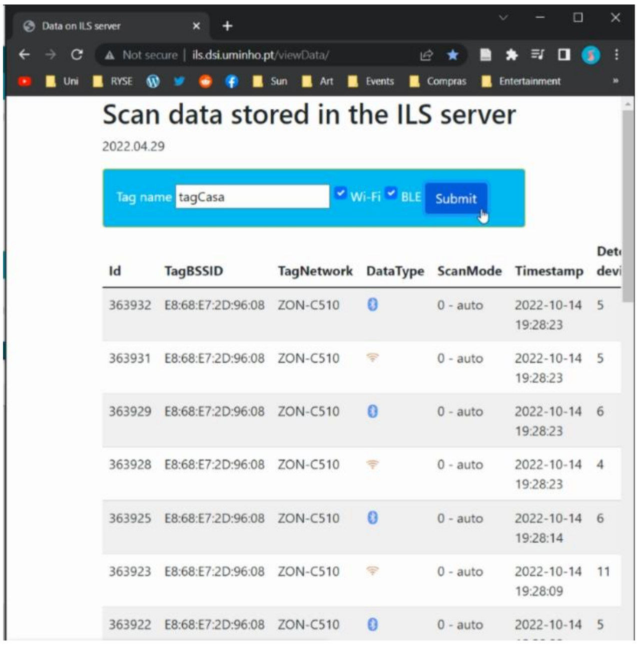


Figure 39 - Posted information.

5.2. Performance Evaluation

Various software versions were built to test the solution - the analysis of these results considered how these models could provide a significant advantage for scanning and posting. Using the tag, evaluating the results derived from the models was paramount. From retrieving data to posting, the software was submitted to many changes. These aimed to reduce the "penalty" times for posting, allowing scans to be performed without problems. The use of threads was optimized, practical and efficient. As a result, it is vital to compare methods, choosing the most effective one out of all available.

5.2.1. Testing Objectives and Procedure

Autonomy tests had to be performed to test the autonomy of the tag. A powerbank was used to evaluate the power consumption. For these tests, firstly, program A would be run. This program allowed the tag to communicate with a local web server to retrieve the parameters' values (Wi-Fi interval scan time, BLE interval scan time, tagName). This information would then be stored in the ESP32 permanent files. Secondly, program B would run, where after the tag connected to the Wi-Fi network, it would scan and post information until the powerbank ran out of battery (Figure 40).

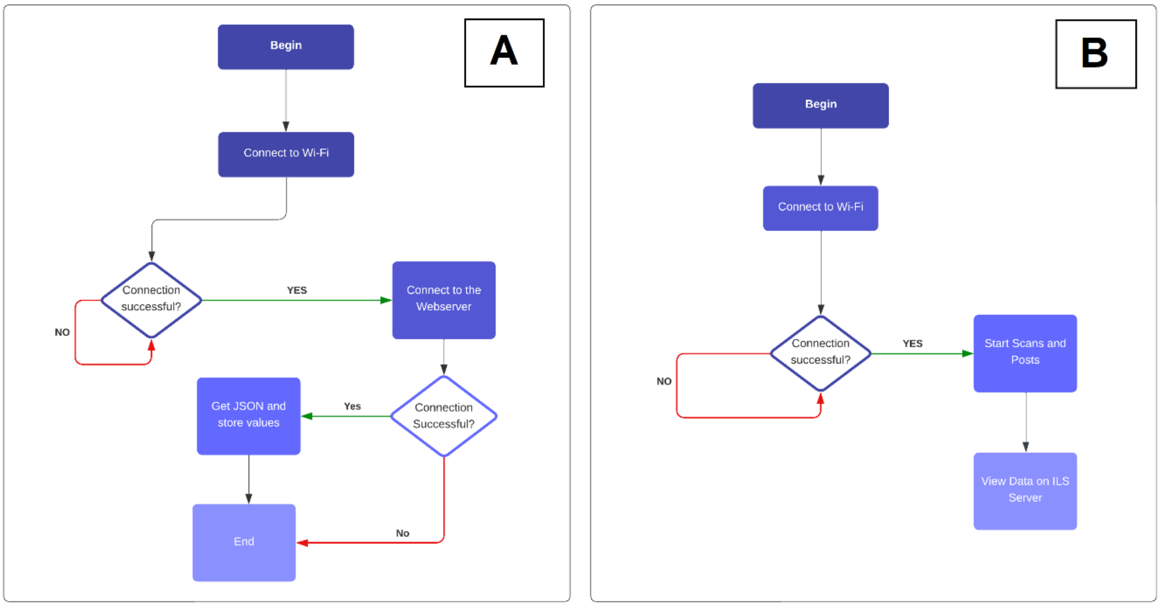


Figure 40 - Testing Program.

Thirty-nine tests were performed in total, with thirteen different parameters. Each parameter would be tested three times in a row to assure test consistency.

5.2.2. Test Environment

An environment for testing is a set of software and hardware that allows the tester to execute test cases. In other words, it will enable the tester to conduct tests with hardware, software and network configurations that support test execution (LaunchDarkly, 2021).

The test environment is configured according to the needs of the test application. In this instance, the test bed comprises the test environment and data. A proper test environment ensures software testing success. An incorrect setup can lead to inconsistent results, resulting in extra costs and delays.

ESP32 firmware is first built in the Arduino IDE before being flashed onto the board (Figure 41). The board is then connected to the powerbank and operates until its battery runs out (Figure 42). Tests were conducted using a home Wi-Fi network.

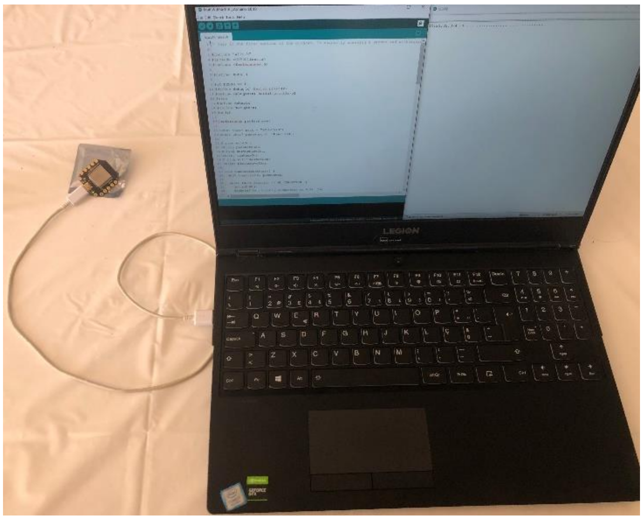
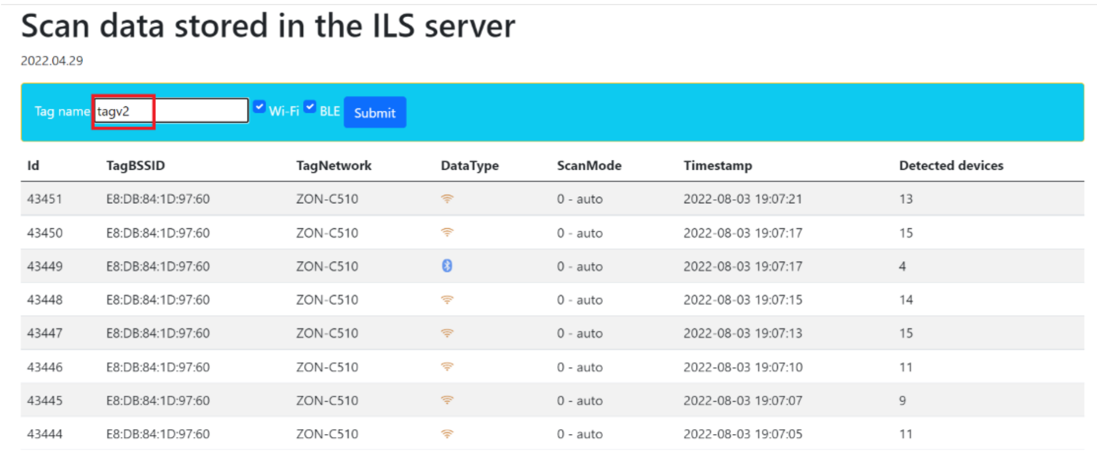


Figure 41 - Firmware Flash.

Figure 42 - ESP32 connected to the Powerbank.

Web Interface

An initial version of a web interface is available to check what has been registered on the server (Figure 43): http://ils.dsi.uminho.pt/viewData/[22]



Scan data stored in the ILS server

2022.04.29

Tag name tagv2 Wi-Fi BLE Submit

Id	TagBSSID	TagNetwork	DataType	ScanMode	Timestamp	Detected devices
43451	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:21	13
43450	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:17	15
43449	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:17	4
43448	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:15	14
43447	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:13	15
43446	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:10	11
43445	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:07	9
43444	E8:DB:84:1D:97:60	ZON-C510		0 - auto	2022-08-03 19:07:05	11

Figure 43 - Scan data stored in the ILS server.

The system only displays the last 30 records.

Apache Server Implementation

The web server is necessary so that instead of having the parameters' values for the scans and posts hard coded, it is possible to change it and obtain the data from an external source. This way, the ESP32 connects

[22] Web Interface (ILS, 2022a)

to it, retrieves its information and proceeds with the scans. The web server is used in test program A, described in section 5.2.1.

Apache[23] (https://www.apachelounge.com/) is the most widely used Web Server application. It can be used on almost all platforms, such as Windows, macOS, Linux and Raspberry. It supports several features; many of them are compiled as separate modules, extend its core functionality, and provide everything from server-side programming language support to authentication mechanisms (Figure 44).

Figure 44 - Apache Lounge.

- **Extract the download files and move them to (C:)**[24]
- **Install Apache using Command Prompt (Figure 45)**

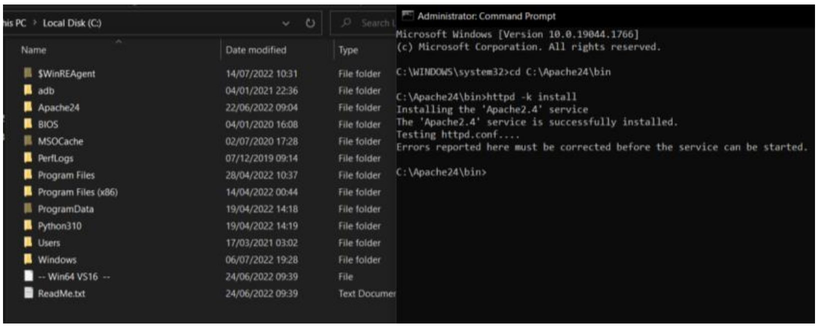


Figure 45 - Apache Command Prompt.

- **Run the APACHE service (Figure 46)**

23 Apache Lounge official website (Apache, 2022)

24 Using the video as a guide (Flavor Of The Month, 2020)

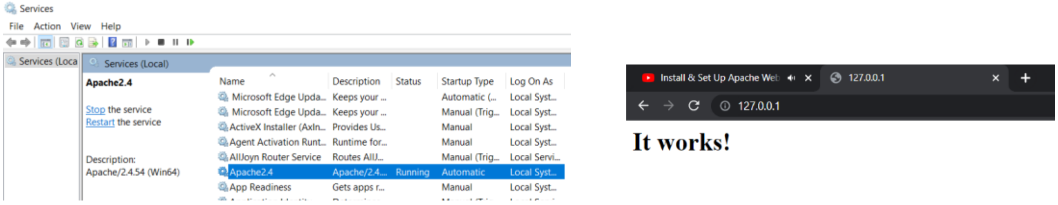


Figure 46 - Apache Service.

- **Find out the local server's IPv4 to access with other devices (Figure 47)**

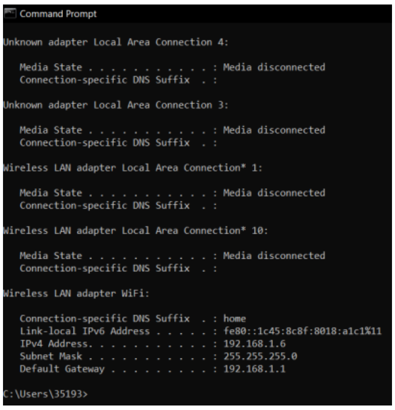


Figure 47 - Local server's IPv4.

- **Edit the INDEX.html file, which is in C:\Apache24\htdocs (Figure 48)**

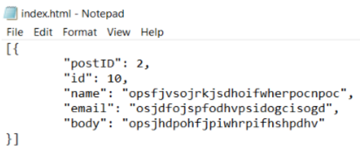


Figure 48 - INDEX.html File.

- **Insert the web server's IPv4 in the Arduino program file (Figure 49)**

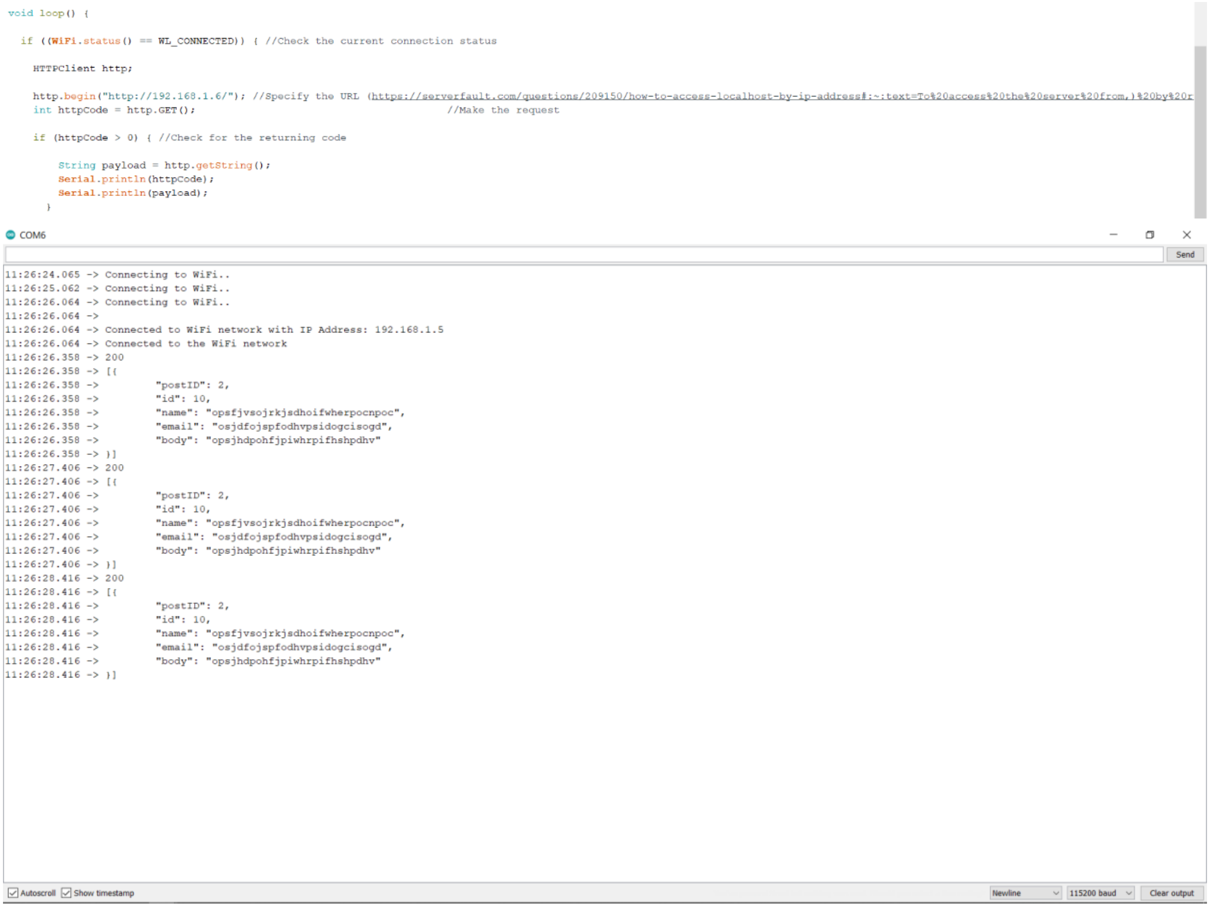


Figure 49 - Inserting the server's IPv4.

Flash Powerbank 2200mAh

ESP32 battery operation is described in this section. To run the module, it is necessary to program the ESP32 with an adapter. Then connecting the module to the battery so that the operation begins.

It is estimated that the ESP32 WROOM module draws around 20 mA of current. Furthermore, extra circuits on the board and the implementation of USB power supply and battery operation are essential factors affecting power consumption. A good battery choice is vital because ESP32 WROOM runs from approximately 2.55 to 3.6 Volts, while external extensions often require at least 3.3 Volts.

ESP32 operation was performed by using a powerbank. The powerbank uses a 3.7V lithium battery and converts the voltage to 5 Volt, which is then reduced to 3.3 Volts by an LDO (low drop-out voltage regulator) (RadioShuttle, 2022).

A battery with 2200 mAh provides enough power to charge a mobile device (Figure 50). The Powerbank charges in 3-5 hours with a USB cable and turns red when charging, off when charged or blue when used. The instructions suggest fully charging up the powerbank before using it (Planetpromo, 2022).

Figure 50 - 2200mAh Powerbank.

In the following table (Table 7), the specifications of the powerbank are listed.

Table 7 - Powerbank Specifications

Specification	Details
Material	Plastic
Dimensions	92 × 26 × 26 mm
Weight	190 g
Potency	2200 mAh
Interface	2.0
Certificates	FCC, RoHS, CE, UL (qualitylogoproducts, 2022)
Input	25V/800mA, Output: 5V/1A
Battery cell Type	Battery cell type: Lithium-ion Grade A Non-Recycled
Input port	Micro USB (A), Output port: USB Type (B)

Specification	Details
Charging time	3-5 hours
Safety protection	Over-charge, over-voltage, short-circuit (eSpares, 2021)

5.2.3. Test Case

A Test Case is a well-designed document to develop and better understand the test case data for a particular scenario (Vijay, 2012). When written in a standard format, all stakeholders can easily understand test cases, lessening the test effort and error rate. (Rajkumar, 2016).

In whichever method of documenting test cases is used, there are specific fields that must be included in any good test case (Table 8):

Table 8 - Test Case Specifications[25]

Test Case Field	Description
Test case ID	A unique ID is required for each test case.
Test Created By	Name of the Tester
Test Execution Date	Date when the test was executed
Description	Describes the test objective in brief
Pre-conditions	Test case execution conditions
Test Steps	Mentions all the test steps in detail and the order in how they could be executed from the user's perspective
Test Data	Input data used for testing. Different data sets are delivered with precise values to be used as an input
Expected Result	The result that is expected once the test cases are executed

[25] In line with the website's description (Hamilton, 2020)

Test Case Field	Description
Actual result	The result that the system shows once the test case is executed. The result is captured after the execution.
Status	Based on the result and the expected result, the status of the test case is set. An actual result that differs from the expected is marked as failed. Otherwise, it is marked as passed

In section A.4 Test Cases, all Test Case files are presented.

5.2.4. Testing Results

After testing, it was concluded that the tag would have an average autonomy of approximately 10 hours and 10 minutes which is considered satisfactory. It was also noted that the powerbank would significantly lose its autonomy as more tests were done. This is due to the lack of quality of the powerbank and significant wear and tear.

The following Table 9 displays the duration of tests, the number of Wi-Fi and BLE records, and the intervals used. The intervals are in milliseconds; the duration is in hours, minutes and seconds and the records are numeric. An interval with a numeric value of -1 means that the technology was not utilized.

Test Case 2 was a failure because it would stop scanning automatically. The author and the advisors could not understand why this occurred. The average values do not include tagT2 data.

Table 9 - Testing Information

Test Case ID	Version	WiFiInterval	BLEinterval	Duration	Wi-Fi Records	BLE Records
tagT1	1	10000	5000	12:35:01	4499	8896
	2			10:37:38	3785	7439
	3			10:18:40	3658	7183
tagT2	1	2000	1000	00:10:47	-	-
	2			06:43:28	2242	2232

Test Case ID	Version	WiFiInterval	BLEinterval	Duration	Wi-Fi Records	BLE Records
tagT3	1	5000	10000	09:19:52	4652	2347
	2			06:57:54	2903	1470
	3			06:57:20	3325	1684
tagT4	1	1000	2000	07:37:43	5757	5056
	2			05:54:16	4589	4032
	3			05:29:32	2648	2326
tagT5	1	8000	10000	11:10:55	4199	3373
	2			07:07:51	2305	1854
	3			07:43:15	2409	1935
tagT6	1	10000	8000	15:46:37	5666	7080
	2			14:21:35	5143	6404
	3			13:52:50	4982	6198
tagT7	1	15000	20000	18:20:41	4386	3300
	2			16:07:42	3865	2895
	3			15:51:20	3794	2837
tagT8	1	20000	15000	15:28:34	2778	3708
	2			14:48:08	2646	3536
	3			14:23:43	2573	3432
tagT9	1	0	20000	06:13:46	7689	966
	2			05:06:12	4817	759
	3			03:30:04	4094	591
tagT10	1	20000	0	10:00:02	1799	18551
	2			08:55:08	1589	13265
	3			08:47:57	1573	12153

Test Case ID	Version	WiFiInterval	BLEinterval	Duration	Wi-Fi Records	BLE Records
tagT11	1	-1	0	09:55:01	0	21172
	2			08:51:51	0	13471
	3			08:31:00	0	14970
tagT12	1	0	-1	06:08:20	8128	0
	2			05:10:22	5591	0
	3			05:01:07	6109	0
tagT13	1	0	0	07:13:41	6660	6736
	2			06:04:37	3178	3119
	3			05:39:31	4222	4161
			Average	09:36:40	3778	5469

Table 10 gives a perspective on the mathematical and statistical data of the first set of tests. This data was calculated with data from the first set of tests. All the attributes are in seconds, except for Total, which is in numeric, and Standard Deviation and Variance, which are in scientific notation.

Table 10 - Statistical Data

		Interval	Total	Mean	Median	Mode	Max	Min	Range	STDEV.S	STDEV.P	Variance.S	Variance.P
TestCase 1	**Wi-Fi**	10	4498	10	10	10	33	00	33	2.49E-05	2.49E-05	6.20E-10	6.20E-10
Version 1	**BLE**	05	8895	05	05	05	26	00	26	2.22E-05	2.22E-05	4.95E-10	4.95E-10
TestCase 3	**Wi-Fi**	05	4651	07	05	05	26	00	26	8.47E-04	8.47E-04	7.18E-07	7.17E-07
Version 1	**BLE**	10	2346	14	10	10	26	01	25	1.19E-03	1.19E-03	1.42E-06	1.42E-06
TestCase 4	**Wi-Fi**	01	5756	05	03	03	09	00	09	1.01E-03	1.01E-03	1.01E-06	1.01E-06
Version 1	**BLE**	02	5055	05	03	03	07	00	07	1.07E-03	1.07E-03	1.15E-06	1.15E-06
TestCase 5	**Wi-Fi**	08	4198	10	08	08	16	00	16	7.96E-04	7.96E-04	6.34E-07	6.33E-07
Version 1	**BLE**	10	3372	12	10	10	15	00	15	8.88E-04	8.88E-04	7.88E-07	7.88E-07
TestCase 6	**Wi-Fi**	10	5665	10	10	10	30	00	30	2.26E-05	2.26E-05	5.09E-10	5.09E-10
Version 1	**BLE**	08	7079	08	08	08	26	00	26	1.83E-05	1.83E-05	3.34E-10	3.34E-10
TestCase 7	**Wi-Fi**	15	4385	15	15	15	34	03	31	1.75E-05	1.75E-05	3.07E-10	3.07E-10
Version 1	**BLE**	20	3299	20	20	20	40	13	27	1.69E-05	1.69E-05	2.86E-10	2.86E-10
TestCase 8	**Wi-Fi**	20	2777	20	20	20	40	17	23	1.36E-05	1.36E-05	1.86E-10	1.86E-10
Version 1	**BLE**	15	3707	15	15	15	27	05	22	1.69E-05	1.69E-05	2.87E-10	2.87E-10

		Interval	Total	Mean	Median	Mode	Max	Min	Range	STDEV.S	STDEV.P	Variance.S	Variance.P
TestCase 9	**Wi-Fi**	00	7688	03	02	02	27	00	27	2.30E-04	2.30E-04	5.31E-08	5.31E-08
Version 1	**BLE**	20	965	23	20	19	31	09	22	6.39E-04	6.39E-04	4.09E-07	4.08E-07
TestCase 10	**Wi-Fi**	20	1798	20	20	20	40	00	40	3.78E-05	3.78E-05	1.43E-09	1.43E-09
Version 1	**BLE**	00	18550	02	02	02	20	00	20	1.72E-05	1.72E-05	2.95E-10	2.95E-10
TestCase 11	**Wi-Fi**	-01	-	-	-	-	-	-	-	-	-	-	-
Version 1	**BLE**	00	21171	02	02	02	23	00	23	1.21E-05	1.21E-05	1.45E-10	1.45E-10
TestCase 12	**Wi-Fi**	00	8127	03	02	02	47	00	47	1.17E-04	1.17E-04	1.37E-08	1.37E-08
Version 1	**BLE**	-01	-	-	-	-	-	-	-	-	-	-	-
TestCase 13	**Wi-Fi**	00	6659	04	03	03	17	00	17	3.35E-04	3.35E-04	1.12E-07	1.12E-07
Version 1	**BLE**	00	6735	04	03	03	17	00	17	3.33E-04	3.33E-04	1.11E-07	1.11E-07
		Average	6244.36364	10	09	09	43	02	40	3.49E-04	3.49E-04	2.92E-07	2.92E-07

When the data collected from the tests was clustered, several histograms were created to visualize the data and retrieve information. These histograms have two variables, Wi-Fi and BLE. The vertical axis is the total number of scans of the three tests made with each parameter.

Test Case 1 (Figure 51) had a Wi-Fi interval of 10 seconds and a BLE interval of 5 seconds. Test Case 6 (Figure 52) had a Wi-Fi interval of 10 seconds and a BLE interval of 8 seconds.

The histogram proves to show that the program was successful. Scans were performed at the corresponding interval. Despite some posts being late and others early, most lived up to expectations.

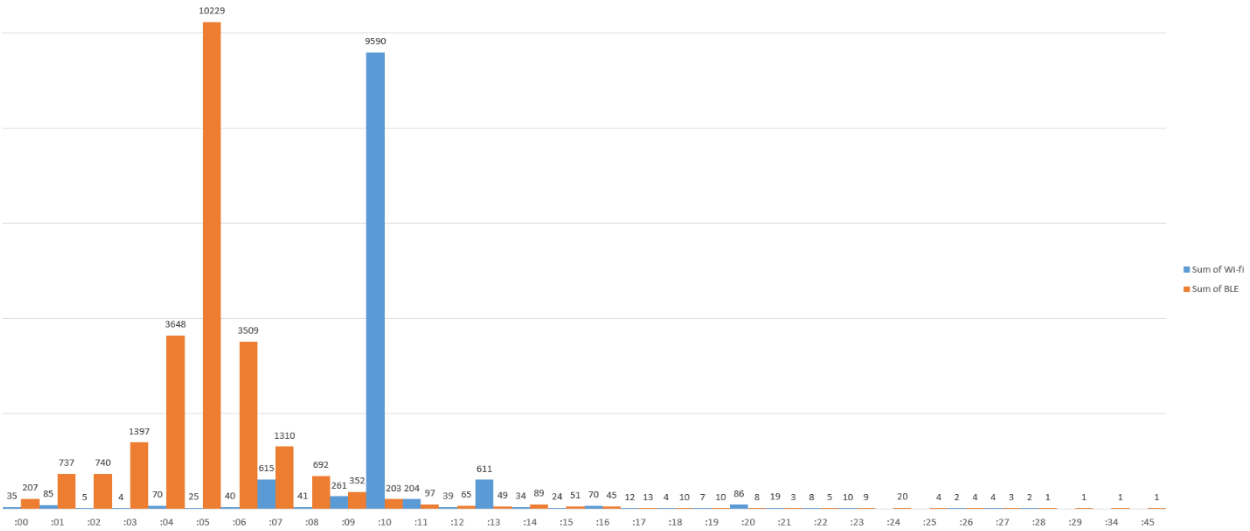


Figure 51 - Test Case 1 Histogram.

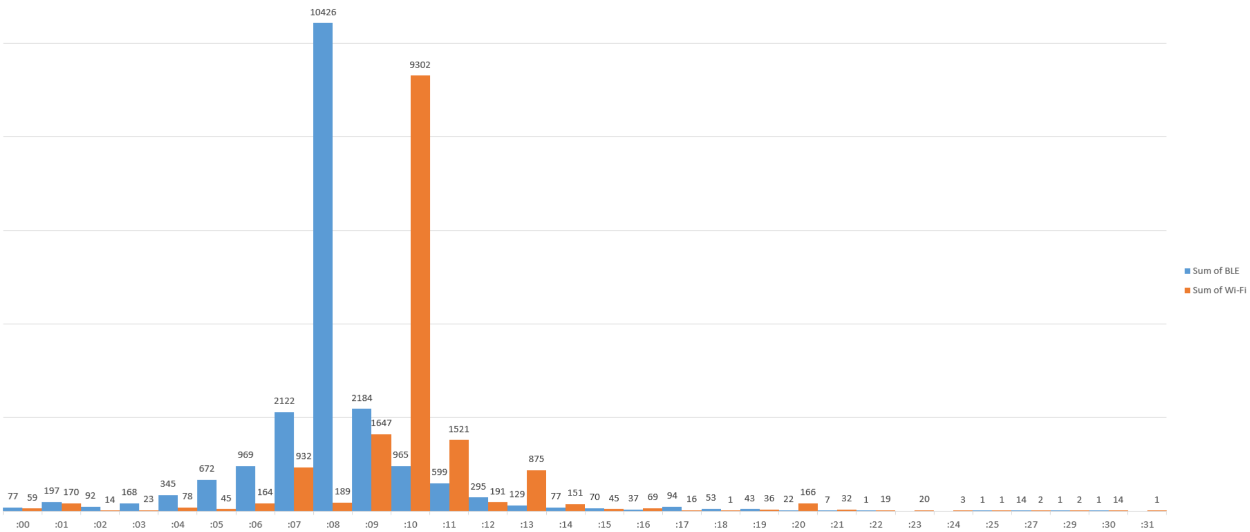


Figure 52 - Test Case 6 Histogram.

All the created histograms are available in section A.4 Test Cases.

6. Conclusion

This project consisted of constructing a solution demonstrating the hardware capabilities of a microcontroller board used for Indoor Localisation. In this chapter, the general and specific objectives of the project are analysed. Difficulties encountered during the development of the system are also presented. Finally, some suggestions for future improvements are proposed.

6.1. Summary

Indoor localisation has already been used to improve customer experience in indoor areas. This work focused on the main problems of the present time, where an extensive system such as the GPS fails in indoor environments. Therefore, it was necessary to think about the dimension of this problem and a way to solve it. A framework is presented on themes that encompass Indoor Localisation systems and what they represent in the functioning of an environment. Once this report was completed, an exploratory analysis was carried out on the quality of the microcontroller boards and an investigation and experience of the same, which was quite a difficult task due to the high knowledge requirements needed. Subsequently, evaluations and research were developed to identify which tags could respond to the problem. Then, the results are shown, identifying the advantages and possibilities of using different requirements. To understand how the developed solution is capable of responding to the problem initially defined, the main results achieved are listed:

- A Beetle ESP32 microcontroller was bought and worked upon because it fitted the requirements;
- Wi-Fi and Bluetooth libraries were incorporated into the code to enable scans;
- HTTP posts were used to integrate the software with an external interface;
- An HTML page containing an Input Form was developed so that external devices could connect and set the parameters for the scans;
- The preferences library was included in the code to allow parameters to be saved permanently in the ESP32;
- Thirty-nine tests were done to test the tag's autonomy.

This project fits industrial positioning designs. It tests the use of these tags for tracking objects in industrial spaces. At the end of the study, it was possible to conclude that the fingerprinting technique presented a high quality and is increasingly used in indoor localisation. However, additional efforts are required to improve these systems' performance so that applications highly dependent on user location can provide better services to their users.

6.2. Risk Table

This artefact comes as a list of known and critical risks to capture the perceived threats to the project's success, sorted in order of importance and associated with specific mitigation and contingency actions (Table 11). It is required to identify the risks' probability and impact (on a scale of 0 to 10) to mitigate them.

Table 11 - Risk Table

ID	Description	Probability (P) [0…10]	Impact (I) [0…10]	Magnitude (P × I)	Mitigation Strategies
1	Inexperience with the tools	8	10	80	The tools used were studied timely, and assistance was asked in case of doubts.
2	Project development delay	5	8	40	The tasks were established at the outset of the project plan.
3	Project requirements change	4	7	28	Reshaping the project was discussed to conform to the new specifications with the supervisors.
4	Unreliable ScanData	4	7	28	ScanData was thoroughly analysed to produce consistent, high-quality scans.

ID	Description	Probability (P) [0…10]	Impact (I) [0…10]	Magnitude (P × I)	Mitigation Strategies
5	Data Transmission errors	5	6	30	Data redundancy was reduced, and primary sources of inaccuracy were identified while regularly updating the system.
6	ESP32 or Powerbank malfunction	2	8	16	Leaving the hardware on prolonged charge/use was avoided as this may cause it to overheat, and it was kept in a safe and dry place away from heat sources, moisture and dust.
7	Server Crashes	2	6	12	Server-side optimisation prevented the server from being overwhelmed and crashing under normal operating conditions.
8	Integration issues	5	7	35	Operating system updates disabled the database engine, so the process supervisor was quickly notified.
9	Fatal firmware	4	8	32	Programs were written well-defined so that

ID	Description	Probability (P) [0...10]	Impact (I) [0...10]	Magnitude (P × I)	Mitigation Strategies
					executions would not be interrupted.

6.3. Future Research

To conclude this study, it is possible to identify a set of skills and relate them to the work to be carried out in the future. The program was expected to have even more autonomy. For that, it is required to be significantly improved. Attaching an accelerometer could be helpful if scans were only performed when the tag is moving. This start & stop system would allow the tag to have a hibernate function with the help of the accelerometer. When it is verified that the object in which the tag is inserted is not moving, the tag will be able to transition to a "hibernate" state that limits its operation to save energy. Likewise, it must transition from this state to "active" as soon as the object starts to move.

Shifting communication protocols could be instrumental if the tag could transition from UDP mode to TCP mode (and vice versa). In UDP mode, it communicates through the UDP protocol that uses less energy and resources in exchange for less reliability. In TCP mode, it communicates via the TCP/IP protocol, which has more excellent information reliability at the cost of higher energy costs.

Both of these requirements are expected to increase the system's autonomy significantly.

REFERENCES

[1] Apache. (2022). Apache VS16. https://www.apachelounge.com/download/

[2] Arduino. (2022). Software. https://www.arduino.cc/en/software

[3] Dev, M. N. (2022). Exception Stack Trace Decoder [Java]. https://github.com/me-no-dev/EspExceptionDecoder (Original work published 2016)

[4] Dev, M. N. (2022). AsyncTCP [C++]. https://github.com/me-no-dev/AsyncTCP (Original work published 2017)

[5] Dev, M. N. (2022). ESPAsyncWebServer [C++]. https://github.com/me-no-dev/ESPAsyncWebServer (Original work published 2015)

[6] DFRobot. (2022a). Arduino Boards Distributor. https://www.dfrobot.com/topic-277.html

[7] DFRobot. (2022b). ESP32(ESP-WROOM-32). https://www.dfrobot.com/product-1559.html

[8] Eisa, S., Peixoto, J., Meneses, F., & Moreira, A. (2013). Removing useless APs and fingerprints from WiFi indoor positioning radio maps. International Conference on Indoor Positioning and Indoor Navigation, 1–7. https://doi.org/10.1109/IPIN.2013.6817919

[9] eSpares. (2021). Portable Power Bank 2200mAh. https://www.espares.co.uk/product/es1742456/portable-power-bank-2200mah

[10] Espressif. (2022). Additional Boards Manager. https://dl.espressif.com/dl/package_esp32_index.json

[11] fear, moe. (2021, September 5). Answer to 'Task watchdog got triggered'. Stack Overflow. https://stackoverflow.com/a/69065654

[12] Flavor Of The Month (Director). (2020, November 2). Install & Set Up Apache Web Server. https://www.youtube.com/watch?v=oJnCEqeAsUk

[13] FreeRTOS. (2022). FreeRTOS queue API. FreeRTOS. https://www.freertos.org/a00018.html

[14] G6EJD - David (Director). (2018, May 30). Runtime Error Debugging. https://www.youtube.com/watch?v=323CS87h6WU

[15] Hamilton, T. (2020, January 22). Test Case Template. https://www.guru99.com/download-sample-test-case-template-with-explanation-of-important-fields.html

[16] ILS. (2022a). Data on ILS server. http://ils.dsi.uminho.pt/viewData/

[17] ILS. (2022b). ILS. http://ils.dsi.uminho.pt/ar-ware/S02/i2a/i2aSamples.php

[18] Khanh, T. T., Nguyen, V., Pham, X.-Q., & Huh, E.-N. (2020). Wi-Fi indoor positioning and navigation: A cloudlet-based cloud computing approach. Human-Centric Computing and Information Sciences, 10(1), 32. https://doi.org/10.1186/s13673-020-00236-8

[19] LaunchDarkly. (2021, April 6). Test Environments 101: Definition, Types, and Best Practices. LaunchDarkly. https://launchdarkly.com/blog/test-environments-101-definition-types-and-best/

[20] Liu, F., Liu, J., Yin, Y., Wang, W., Hu, D., Chen, P., & Niu, Q. (2020). Survey on WiFi-based indoor positioning techniques. IET Communications, 14(9), 1372–1383. https://doi.org/10.1049/iet-com.2019.1059

[21] Maker Tutor (Director). (2018, December 9). Installing the ESP32 Board in Arduino IDE. https://www.youtube.com/watch?v=hjJx6QOWVkU

[22] Marques, N., Meneses, F., & Moreira, A. (2012). Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning. 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1–9. https://doi.org/10.1109/IPIN.2012.6418937

[23] Planetpromo. (2022). Flash power bank 2200mAh. Promotional Items and Corporate Gifts - Planetpromo.Eu. https://www.planetpromo.eu/en/12357101.html

[24] Polykanine, A. (2021). Request failed, httpCode: -11. https://community.blynk.cc/t/request-failed-httpcode-11/26544/5

[25] Postman. (2022). Postman API Platform. https://www.postman.com/

[26] qualitylogoproducts. (2022). Customized Charge-It-Up Power Bank Chargers (2200 mAh, UL Listed). Www.Qualitylogoproducts.Com. https://www.qualitylogoproducts.com/tradeshow-promotions/charge-it-up-power-bankcharger.htm

[27] RadioShuttle. (2022). Battery-Powered ESP32. https://www.radioshuttle.de/en/media-en/tech-infos-en/battery-powered-esp32/

[28] Rajkumar. (2016, February 10). How to Write Test Cases. Software Testing Material. https://www.softwaretestingmaterial.com/test-case-template-with-explanation/

[29] Ralph S Bacon (Director). (2021, September 3). STOP using Serial.print. https://www.youtube.com/watch?v=–KxxMaiwSE

[30] Random Nerd Tutorials. (2018, October 4). ESP32 Dual Core with Arduino IDE. https://randomnerdtutorials.com/esp32-dual-core-arduino-ide/

[31] Random Nerd Tutorials. (2019a, May 16). ESP32 Bluetooth Low Energy. https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/

[32] Random Nerd Tutorials. (2019b, August 29). Input Data on HTML Form ESP32/ESP8266 Web Server Arduino IDE. https://randomnerdtutorials.com/esp32-esp8266-input-data-html-form/

[33] Random Nerd Tutorials. (2020, April 8). ESP32 HTTP GET and HTTP POST with Arduino IDE. https://randomnerdtutorials.com/esp32-http-get-post-arduino/

[34] Random Nerd Tutorials. (2021a, February 12). ESP32 Wi-Fi Library Functions. https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/

[35] Random Nerd Tutorials. (2021b, March 2). ESP32 Save Data Permanently using Preferences Library. https://randomnerdtutorials.com/esp32-save-data-permanently-preferences/

[36] Sakpere, W., Adeyeye Oshin, M., & Mlitwa, N. (2017). A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies. South African Computer Journal, 29, 145. https://doi.org/10.18489/sacj.v29i3.452

[37] Sam. (2021, February 19). Task watchdog got triggered [Forum post]. Stack Overflow. https://stackoverflow.com/q/66278271

[38] Subedi, S., & Pyun, J.-Y. (2020). A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies. Sensors, 20(24), Article 24. https://doi.org/10.3390/s20247230

[39] Tarmo. (2021, February 19). Answer to 'Task watchdog got triggered'. Stack Overflow. https://stackoverflow.com/a/66279325

[40] techtutorialsx. (2017, August 20). FreeRTOS Queues. https://techtutorialsx.com/2017/08/20/esp32-arduino-freertos-queues/

[41] uPesy. (2022). Create a Wi-Fi access point with an ESP32. UPesy. https://www.upesy.com/blogs/tutorials/how-create-a-wifi-acces-point-with-esp32

[42] Vijay. (2012, December 12). Sample Test Case Template. Software Testing Help. https://www.softwaretestinghelp.com/test-case-template-examples/

[43] Wolph. (2018). What does response code -3 mean? Cookiebot Support. http://support.cookiebot.com/hc/en-us/community/posts/4410841627154-What-does-response-code-3-mean-

[44] XTronical (Director). (2020, April 14). ESP32—Out of memory—How to solve this compiler error and get more of the memory you paid for! https://www.youtube.com/watch?v=5VoXNIoOwZE

[45] Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., & Raulefs, R. (2016). Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. IEEE Communications Surveys & Tutorials, PP, 1–1. https://doi.org/10.1109/COMST.2016.2632427

[46] Zafari, F., Gkelias, A., & Leung, K. (2017). A Survey of Indoor Localization Systems and Technologies. IEEE Communications Surveys & Tutorials, PP. https://doi.org/10.1109/COMST.2019.2911558

[47] Zghair et. al. (2019). Indoor Localization System Using Wi-Fi Technology. Iraqi Journal of

Computer, Communication, Control and System Engineering, 69–77. https://doi.org/10.33103/uot.ijccce.19.2.8

7. ANNEXE

A.1.Preliminary Research

An integrated microcontroller board was selected to minimise the hardware development effort. Initially, several microcontroller boards were contemplated for use as the tag's main component, bearing in mind that they would be discounted later based on functional requirements. This table lists all the specifications of the first set of microcontroller boards.

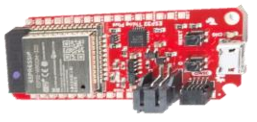
Product	Brand	Wi-Fi	Bluetooth	Cost	Size	Weight	Board	Microcontroller	USB Connector	PINS	Connectivity	Sensors	Communication	Clock Speed	Memory
Nano RP2040 Connect	Arduino	802.11b/g/n	Bluetooth® and BLE v4.2	21 €	45mmx18mm	6g	Nano RP2040 Connect	Raspberry Pi RP2040	MicroUSB	22 digital, 20 with PWM and eight analog	ATECC608A-MAHDA-T Crypto IC	LSM6DSOXTR (6-axis)	UART I2C SPI	133 MHz	16MB Flash IC
Black Wireless	BeagleBone	802.11b/g/n 2.4GHz	Bluetooth 4.1 plus BLE	100 €	-	-	Nano RP2040 Connect	2x PRU 32-bit	USB client/USB host	2x 46-pin headers	-	-	-	800MHZ	4GB 8-bit eMMC on-board flash storage

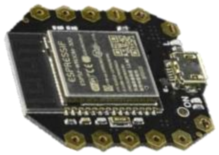
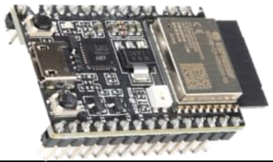
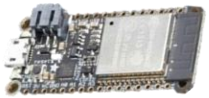
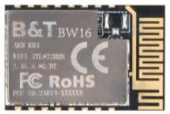
Product	Brand	Wi-Fi	Bluetooth	Cost	Size	Weight	Board	Microcontroller	USB Connector	PINS	Connectivity	Sensors	Communication	Clock Speed	Memory
Blue	BeagleBone	802.11bgn	Bluetooth 4.1 and BLE	118 €	-	-	AM335x 1GHz ARM® Cortex-A8	2x PRU 32-bit 200MHz microcontrollers	USB2client/USB2host	2x 46-pin headers	-	Sensors: 9-axis IMU (accels, gyros, magnetometer), barometer, thermometer	GPS, DSM2 radio, UARTs, SPI, I2C, 1.8V analog, 3.3V GPIOs	2×32-bit 200-MHz programmable real-time units (PRUs)	4GB 8-bit eMMC flash storage
AI	BeagleBone	2.4/5GHz Wi-Fi	Bluetooth	113 €	8.9cm x 5.4cm x 1.5cm	48 g	-	-	USBtypeC USBtypeA	-	-	-	-	533 MHz	16GB on-board eMMC flash
Green Wireless	BeagleBone	WiFi 802.11 b/g/n 2.4GHz	Bluetooth 4.1 with BLE	56 €	-	-	AM335x 1GHz ARM® Cortex-A8	2x PRU 32-bit microcontrollers	USB client/USB host	2x 46-pin headers	-	-	2x Grove connectors (I2C and UART)	-	4GB 8-bit eMMC on-board flash storage
Enhanced	BeagleBone	?	?	100 €	-	-	TI AM3358 ARM® Cortex A8	2x PRU 32-bit microcontrollers	USB client/USB host	2x 46-pin headers	-	-	-	-	4GB 8-bit eMMC on-board flash storage
Thing Plus	SparkFun	802.11 BGN	Integrated dual-mode Bluetooth (classic and BLE)	$20.95	2.30 x 0.90 Inches	-	Dual-core Tensilica LX6	-	MicroUSB	38 pins	-	-	high-speed SPI, UART, I^2S and I^2C	Up to 240MHz clock frequency	16MB of flash storage

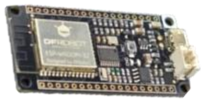
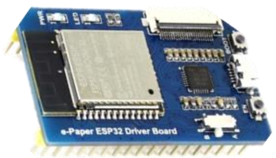
Product	**Brand**	**Wi-Fi**	**Bluetooth**	**Cost**	**Size**	**Weight**	**Board**	**Microcontroller**	**USB Connector**	**PINS**	**Connectivity**	**Sensors**	**Communication**	**Clock Speed**	**Memory**
C3 WROOM Module	ESP32	IEEE 802.11 b/g/n-compliant	Bluetooth LE: Bluetooth 5, Bluetooth mesh	$3.50	18.0 × 20.0 × 3.2 mm	-	-	32-bit RISC-V single-core processor, up to 160 MHz	-	-	-	-	GPIO SPI UART I2C I2S	160 MHz	4 MB SPI flash
C3 WROOM Development Board	ESP32	WiFi IEEE 802.11 b/g/n-compliant	Bluetooth LE: Bluetooth 5, Bluetooth mesh	$10.00	-	-	-	32-bit RISC-V single-core processor, up to 160 MHz	MicroUSB	-	-	-	-	160 MHz	4 MB SPI flash
ESP32	GeekCreit	2.4 GHz dual-mode	?	8.25 €	-	-	Tensilica Xtensa® Dual-Core 32-bit LX6	-	-	48 pins	-	-	ADC UART PWM DAC SPI I2C I2S	-	-

A.2. Outlining Possibilities

The initial consideration of microcontroller boards led to several additional ones being examined, and some previously considered were disregarded due to functional demands. In this section, an enlarged analysis of the data is presented to get a better picture based on the size and weight of the boards.

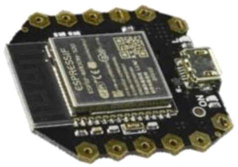
Name	Picture	Number of pins	Length	Width	Module	Wi-Fi	Bluetooth	IDE	Soldering	Price	Link	DataSheet
SparkFun Thing Plus		28	59mm	24mm	ESP-WROOM-32	Yes	Yes	Arduino	No	18,50 €	Link1	Datasheet 1
Arduino Nano RP2040 Connect		30	45mm	18mm	Nina W102 uBlox	Yes	Yes	Arduino	No	21 €	Link2	Datasheet 2
Arduino Nano 33 IoT		30	45 mm	18 mm	Nina W102 uBlox	Yes	Yes	Arduino	No	16 €	Link3	Datasheet 3
Arduino MKR Wi-Fi 10		28	61.5 mm	25 mm	Nina W102 uBlox	Yes	Yes	Arduino	No	27,90 €	Link4	Datasheet 4

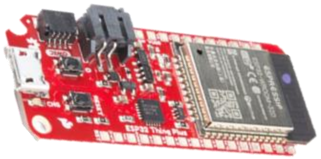
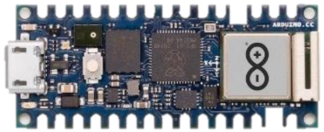
Name	Picture	Number of pins	Length	Width	Module	Wi-Fi	Bluetooth	IDE	Soldering	Price	Link	DataSheet
BEETLE		14	35mm	34mm	ESP-WROOM-32	Yes	Yes	Arduino	No	15,95 €	Link5	Datasheet 5
ESP32 Development Board		36	57mm	28mm	-	Yes	Yes	Arduino	No	15	Link6	Datasheet 6
Adafruit HUZZAH32		28	51.0mm	22.7mm	ESP-WROOM-32	Yes	Yes	Arduino	No	17.62	Link7	Datasheet 7
Realtek		16	24mm	16mm	RTL8720DN	Yes	Yes	Arduino	Yes	7,10 €	Link8	Datasheet 8
TTGO		32	40.27mm	31.07mm	-	Yes	Yes	Arduino	No	15,87 €	Link9	Datasheet 9

Name	Picture	Number of pins	Length	Width	Module	Wi-Fi	Bluetooth	IDE	Soldering	Price	Link	DataSheet
FireBeetle		36	53mm	24mm	ESP-WROOM-32	Yes	Yes	Arduino	No	19,90 €	Link10	Datasheet 10
E-paper		32	48.25mm	29.46mm	ESP-WROOM-32	Yes	Yes	Arduino	No	13,95 €	Link11	Datasheet 11
LuaNode		30	51.45mm	23.37mm	ESP-WROOM-32	Yes	Yes	Arduino	No	9,95 €	Link12	Datasheet 12
NodeMCU		36	48mm	26mm	ESP-WROOM-32	Yes	Yes	Arduino	No	12	Link13	Datasheet 13
ESP32 Module		36	25.2mm	18mm	-	Yes	Yes	Arduino	Yes	10	Link14	Datasheet 14

A.3. Final Decision

The board had to be evaluated for its functional features and extras that would be useful for the work. All four boards listed in this table are deemed the most appropriate and optimal for the project.

Preference	Name	Picture	Number of pins	Length	Width	Module	Wi-Fi	Bluetooth	IDE	Soldering?	Price	Link	DataSheet	Justification
1	**BEETLE**		14	35 mm	34 mm	ESP-WROOM-32	Yes	Yes	Arduino	No	15,95 €	Link1	Datasheet1	A small-scale electronic device with a great range of benefits such as low power consumption, an enticingly small form factor, a surprising amount of functionality and much computational power

2	**SparkFun Thing Plus**		28	59 mm	24 mm	ESP-WROOM-32	Yes	Yes	Arduino	No	18,50 €	Link2	Datasheet2	An easy-to-use, robust, scalable and adaptable microcontroller unit enabled with the Qwiic Connect System. Soldering or shielding is not required.
3	**Arduino Nano RP2040 Connect**		30	45 mm	18 mm	Nina W102 uBlox	Yes	Yes	Arduino	No	21 €	Link3	Datasheet3	A versatile plug&play board with dual-core, connectivity, audio and machine learning capabilities. It also takes less time to compile with the benefits of a full Arduino Cloud support

4	**ESP32 Module**		36	25.2mm	18 mm	-	Yes	Yes	Arduino	Yes	10	Link4	Datasheet4	Based on its value for money, wide range of features, small size and relatively low power consumption, it is well suited to the IoT application. A good understanding of soldering is required.

A.4. Test Cases

Each test case represents one set of parameters that were tested. Each set of parameters was tested three times. The histogram with clustered data is available for each parameter at the end of every third (final) test case.

Test Case ID	tagT1	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	8-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT1
2	WIFIinterval = 10000
3	BLEinterval = 5000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-08 13:14:14	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-09 01:49:15	Pass
9	Calculate tag’s execution time	-	12 hours 35 minutes 0 seconds	Pass

Test Case ID	tagT1	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	14-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

Prerequisites:
Powerbank fully charged
Tag data saved in the flash memory
Internet Access
Webserver online

S #	Test Data
1	TagName = tagT1
2	WIFIinterval = 10000
3	BLEinterval = 5000

A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
Enter TagName	Credential can be entered	As Expected	Pass
Click Submit	Initial tag data is displayed	As Expected	Pass
Record initial tag data	-	2022-10-14 10:34:19	Pass
Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
Click Submit	Final tag data is displayed	As Expected	Pass
Record final tag data	-	2022-10-14 21:11:57	Pass

Calculate tag's execution time	-	10 hours 37 minutes 38 seconds	Pass

Test Case ID	tagT1	**Test Case Description**	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	**Version**	3.0

Date Tested	15-Oct-2022	**Test Case (Pass/Fail/Not Executed)**	Pass

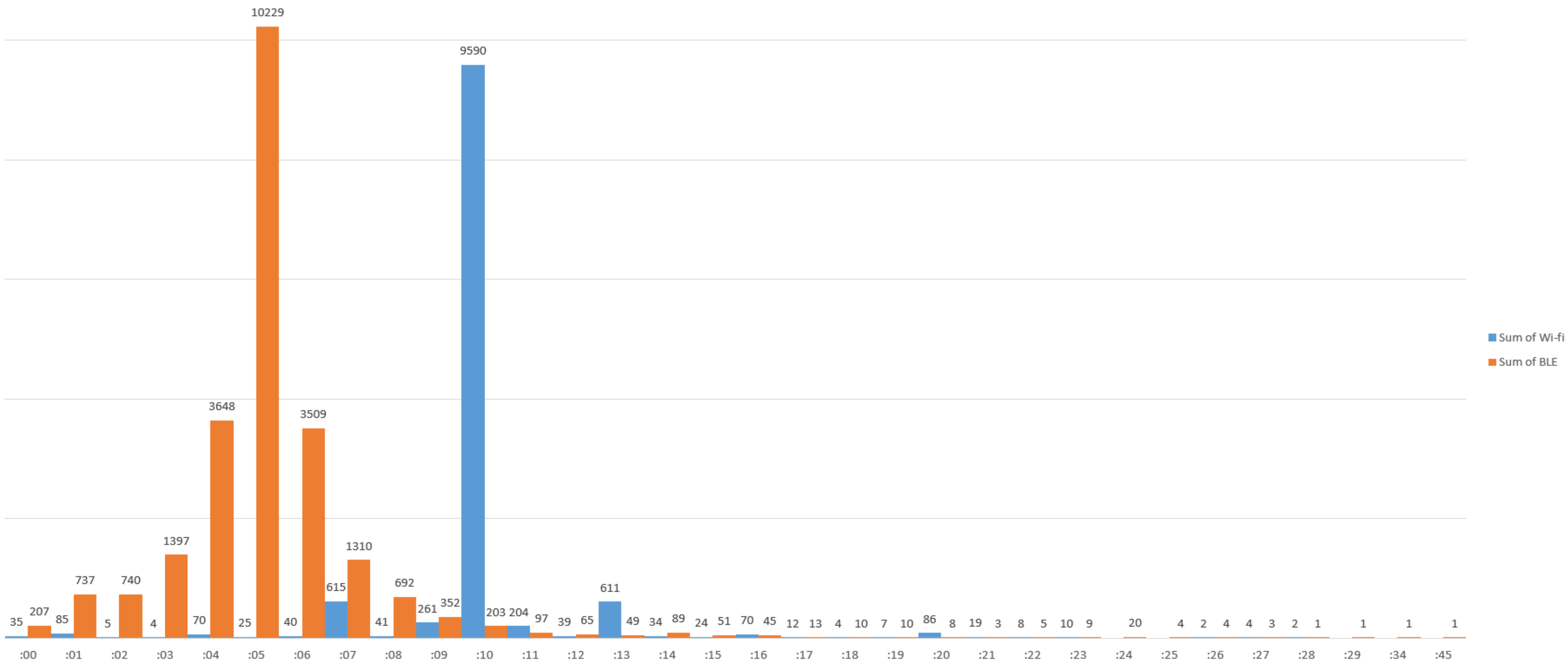
S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT1
2	WIFIinterval = 10000
3	BLEinterval = 5000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-15 09:54:31	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass

8	Record final tag data	-	2022-10-15 20:13:11	Pass
9	Calculate tag’s execution time	-	10 hours 18 minutes 40 seconds	Pass



Sum of Wi-fi
Sum of BLE
:00 :01 :02 :03 :04 :05 :06 :07 :08 :09 :10 :11 :12 :13 :14 :15 :16 :17 :18 :19 :20 :21 :22 :23 :24 :25 :26 :27 :28 :29 :34 :45
35 207 85 737 5 740 4 1397 70 3648 25 10229 40 3509 615 1310 41 692 261 352 9590 203 204 97 39 65 611 49 34 89 24 51 70 45 12 13 4 10 7 10 86 8 19 3 8 5 10 9 20 4 2 4 4 3 2 1 1 1 1

Test Case ID	tagT2	**Test Case Description**	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	**Version**	1.0

Date Tested	10-Aug-2022	**Test Case (Pass/Fail/Not Executed)**	Fail

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT2
2	WIFIinterval = 2000
3	BLEinterval = 1000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-10 13:53:03	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-10 14:03:50	Pass
9	Calculate tag's execution time	-	0 hours 10 minutes 47 seconds	Fail

Test Case ID	tagT2	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	21-Sep-2022	Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT2
2	WIFIinterval = 2000
3	BLEinterval = 1000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-21 14:02:35	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-21 20:46:03	Pass
9	Calculate tag’s execution time	-	6 hours 43 minutes 28 seconds	Fail

Test Case ID	tagT3	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	15-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT3
2	WIFIinterval = 5000
3	BLEinterval = 10000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-15 13:13:22	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-15 22:33:14	Pass
9	Calculate tag's execution time	-	9 hours 19 minutes 52 seconds	Pass

Test Case ID	tagT3	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	16-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT3
2	WIFIinterval = 5000
3	BLEinterval = 10000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-16 09:56:26	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-16 16:54:20	Pass
9	Calculate tag's execution time	-	6 hours 57 minutes 54 seconds	Pass

Test Case ID	tagT3	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	3.0

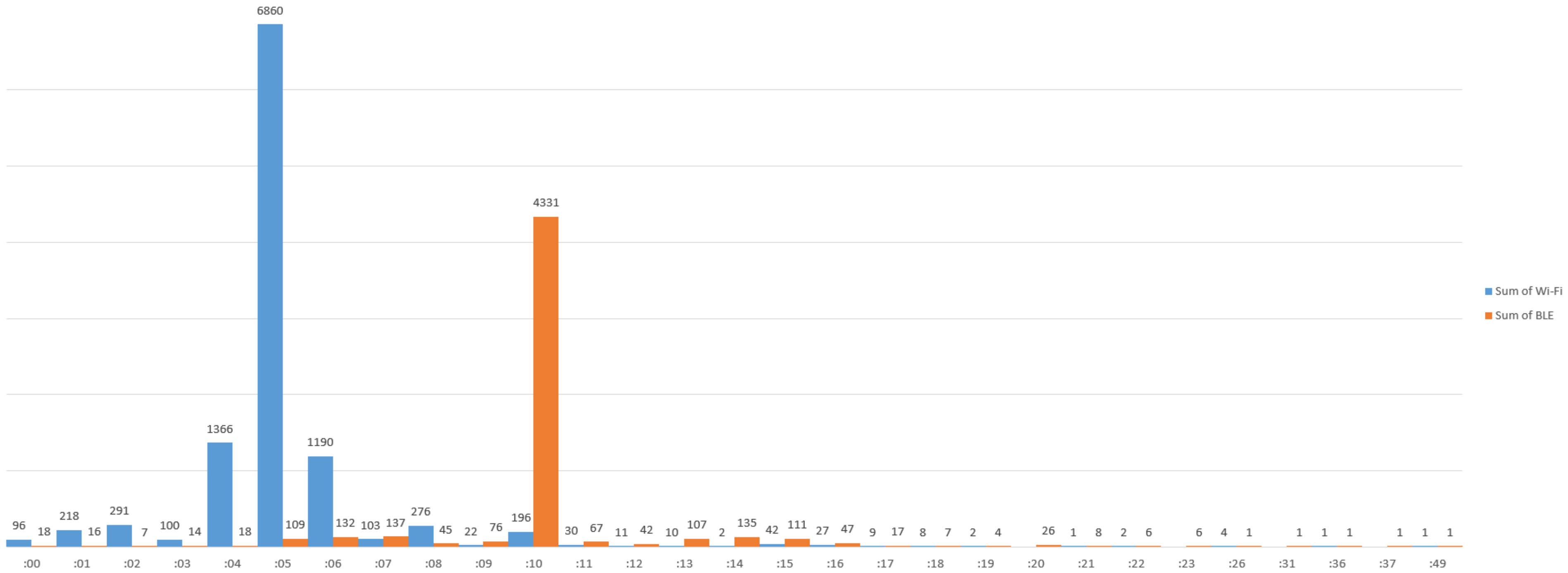
Date Tested	17-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT3
2	WIFIinterval = 5000
3	BLEinterval = 10000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-17 11:55:10	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-17 18:52:30	Pass
9	Calculate tag's execution time	-	6 hours 57 minutes 20 seconds	Pass



Sum of Wi-Fi
Sum of BLE
96
18
218
16
291
7
100
14
1366
18
6860
109
1190
132
103
137
276
45
22
76
196
4331
30
67
11
42
10
107
2
135
42
111
27
47
9
17
8
7
2
4
26
1
8
2
6
6
4
1
1
1
1
1
1
1
:00
:01
:02
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:26
:31
:36
:37
:49

Test Case ID	TagT4	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	16-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT4
2	WIFIinterval = 1000
3	BLEinterval = 2000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-16 16:50:46	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-17 00:28:29	Pass
9	Calculate tag’s execution time	-	7 hours 37 minutes 43 seconds	Pass

Test Case ID	TagT4	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	18-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT4
2	WIFIinterval = 1000
3	BLEinterval = 2000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-18 14:53:00	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-18 20:47:16	Pass
9	Calculate tag's execution time	-	5 hours 54 minutes 16 seconds	Pass

Test Case ID	TagT4	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	3.0

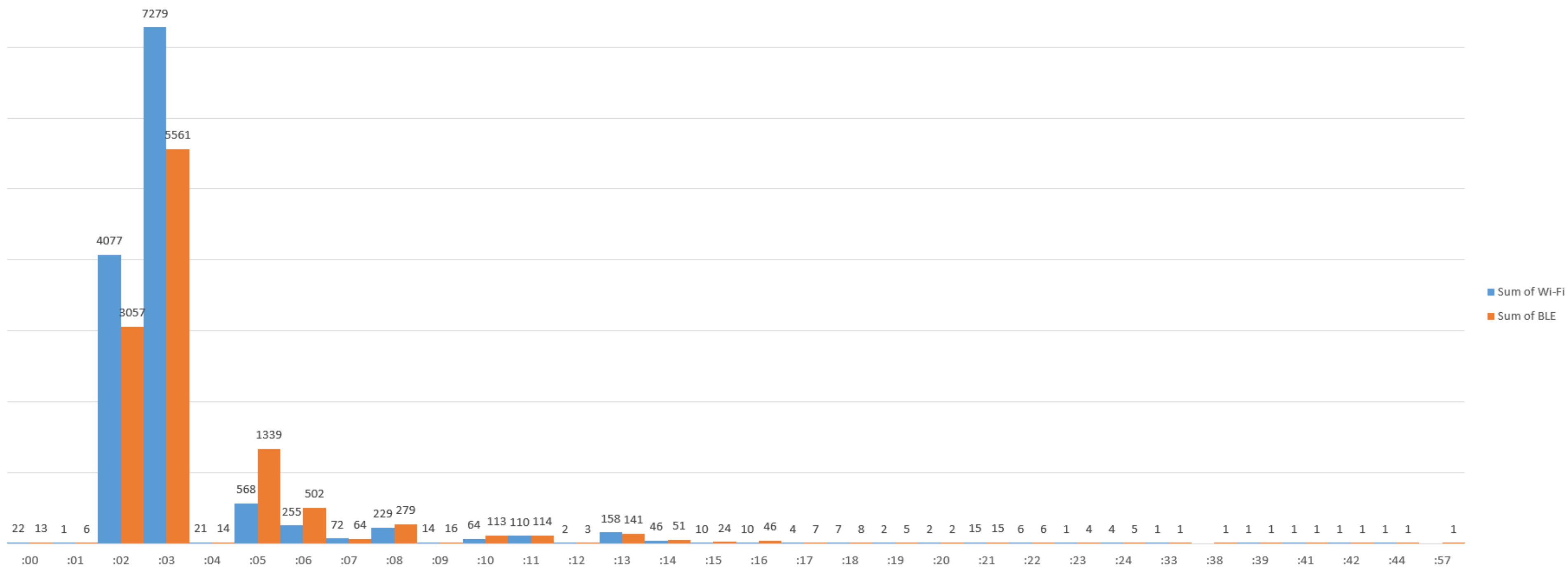
Date Tested	19-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT4
2	WIFIinterval = 1000
3	BLEinterval = 2000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-19 10:34:04	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-19 16:03:36	Pass
9	Calculate tag's execution time	-	5 hours 29 minutes 32 seconds	Pass



22
13
1
6
4077
3057
7279
5561
21
14
568
1339
255
502
72
64
229
279
14
16
64
113
110
114
2
3
158
141
46
51
10
24
10
46
4
7
7
8
2
5
2
2
15
15
6
6
1
4
4
5
1
1
1
1
1
1
1
1
1
1
1
1
:00
:01
:02
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:24
:33
:38
:39
:41
:42
:44
:57
Sum of Wi-Fi
Sum of BLE

Test Case ID	TagT5	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	17-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT5
2	WIFIinterval = 8000
3	BLEinterval = 10000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-17 11:22:24	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-17 22:33:19	Pass
9	Calculate tag's execution time	-	11 hours 10 minutes 55 seconds	Pass

Test Case ID	TagT5	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	20-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT5
2	WIFIinterval = 8000
3	BLEinterval = 10000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-20 11:14:13	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-20 18:22:04	Pass
9	Calculate tag’s execution time	-	7 hours 7 minutes 51 seconds	Pass

Test Case ID	TagT5	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	3.0

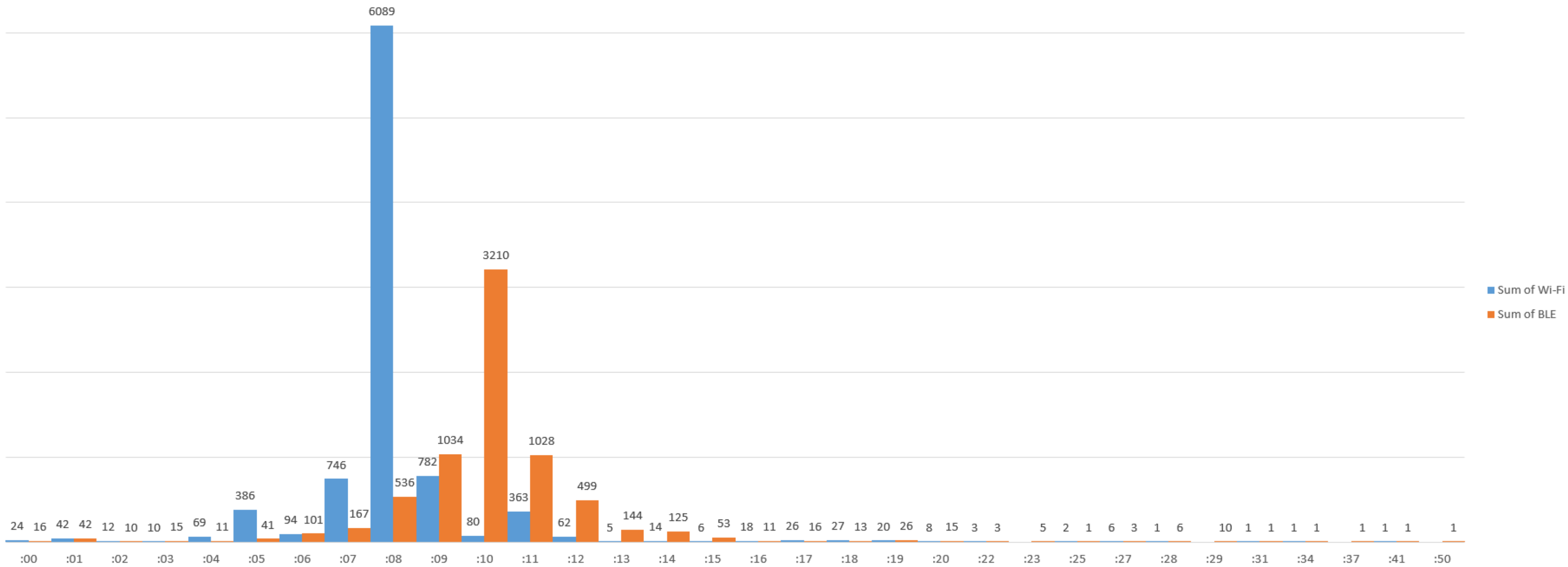
Date Tested	21-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT5
2	WIFIinterval = 8000
3	BLEinterval = 10000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-21 10:08:00	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-21 17:51:15	Pass
9	Calculate tag's execution time	-	7 hours 43 minutes 15 seconds	Pass


Sum of Wi-Fi
Sum of BLE
6089
3210
1034
1028
782
746
536
499
386
363
167
144
125
101
94
80
69
62
53
41
24 16 42 42 12 10 10 15 69 11 386 41 94 101 746 167 6089 536 782 1034 80 3210 363 1028 62 499 5 144 14 125 6 53 18 11 26 16 27 13 20 26 8 15 3 3 5 2 1 6 3 1 6 10 1 1 1 1 1 1 1 1
:00 :01 :02 :03 :04 :05 :06 :07 :08 :09 :10 :11 :12 :13 :14 :15 :16 :17 :18 :19 :20 :22 :23 :25 :27 :28 :29 :31 :34 :37 :41 :50

Test Case ID	TagT6	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	18-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT6
2	WIFIinterval = 10000
3	BLEinterval = 8000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-18 14:54:01	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-19 06:40:38	Pass
9	Calculate tag's execution time	-	15 hours 46 minutes 37 seconds	Pass

Test Case ID	TagT6	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	23-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT6
2	WIFIinterval = 10000
3	BLEinterval = 8000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-23 12:04:32	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-24 02:26:07	Pass
9	Calculate tag’s execution time	-	14 hours 21 minutes 35 seconds	Pass

Test Case ID	TagT6	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	3.0

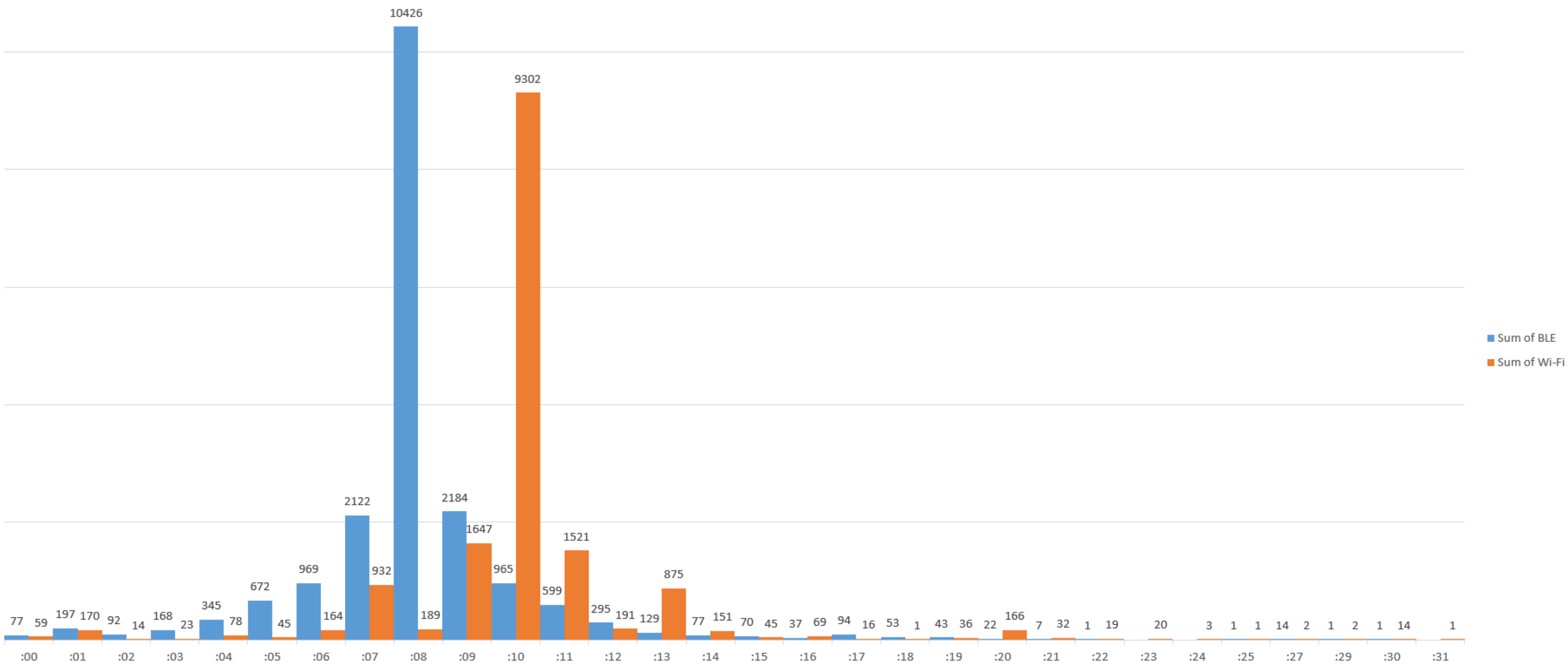
Date Tested	24-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT6
2	WIFIinterval = 10000
3	BLEinterval = 8000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-24 16:51:39	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-25 06:44:29	Pass
9	Calculate tag’s execution time	-	13 hours 52 minutes 50 seconds	Pass


77
59
197
170
92
14
168
23
345
78
672
45
969
164
2122
932
10426
189
2184
1647
965
9302
599
1521
295
191
129
875
77
151
70
45
37
69
94
16
53
1
43
36
22
166
7
32
1
19
20
3
1
1
14
2
1
2
1
14
1
:00
:01
:02
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:24
:25
:27
:29
:30
:31
Sum of BLE
Sum of Wi-Fi

Test Case ID	TagT7	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	19-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT7
2	WIFIinterval = 15000
3	BLEinterval = 20000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-19 18:50:28	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-20 13:11:09	Pass
9	Calculate tag's execution time	-	18 hours 20 minutes 41 seconds	Pass

Test Case ID	TagT7	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	25-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT7
2	WIFIinterval = 15000
3	BLEinterval = 20000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-25 15:03:22	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-26 07:11:04	Pass
9	Calculate tag’s execution time	-	16 hours 7 minutes 42 seconds	Pass

Test Case ID	TagT7	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	3.0

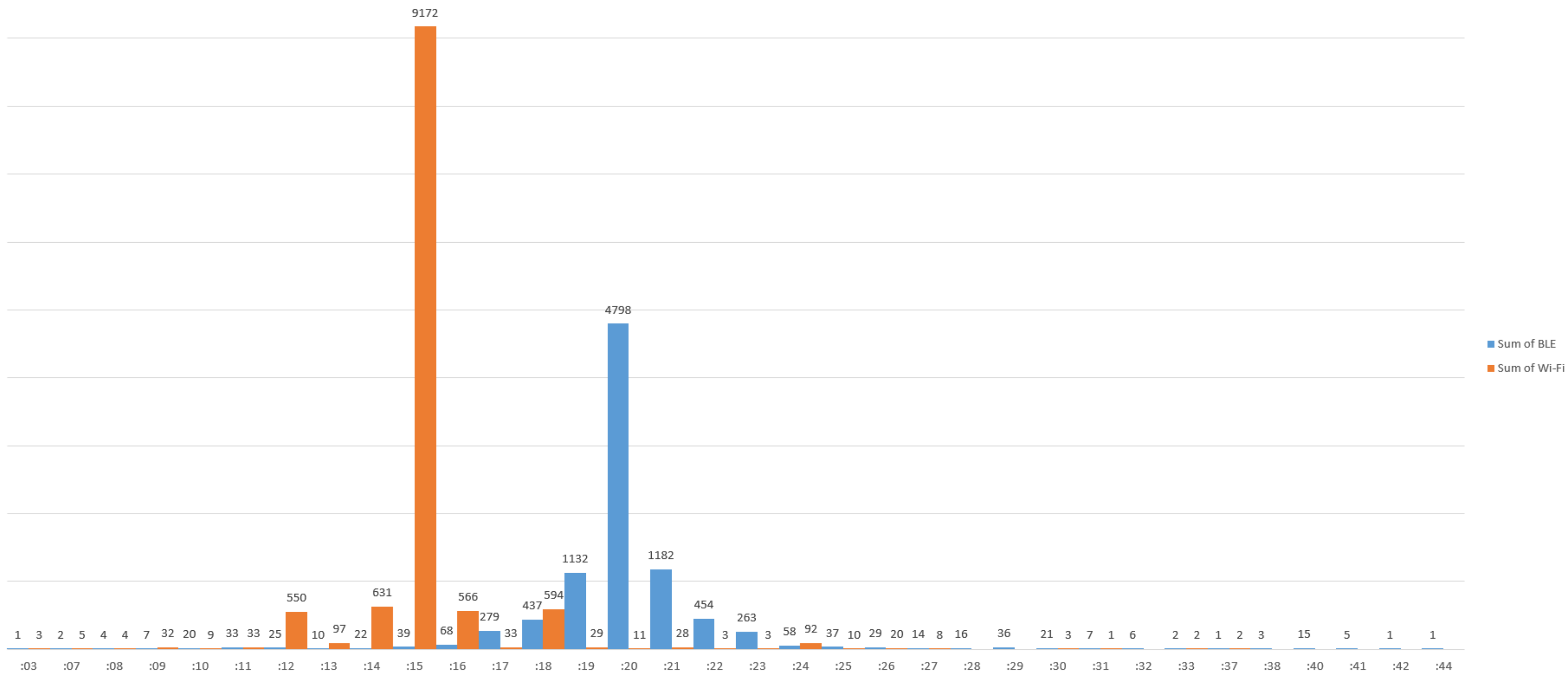
Date Tested	26-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT7
2	WIFIinterval = 15000
3	BLEinterval = 20000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-26 14:11:21	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-27 06:02:41	Pass
9	Calculate tag’s execution time	-	15 hours 51 minutes 20 seconds	Pass


9172
4798
1132
1182
550
631
566
594
437
454
279
263
1
3
2
5
4
4
7
32
20
9
33
33
25
10
97
22
39
68
33
29
11
28
3
3
58
92
37
10
29
20
14
8
16
36
21
3
7
1
6
2
2
1
2
3
15
5
1
1
:03
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:24
:25
:26
:27
:28
:29
:30
:31
:32
:33
:37
:38
:40
:41
:42
:44
Sum of BLE
Sum of Wi-Fi

Test Case ID	TagT8	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	21-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT8
2	WIFIinterval = 20000
3	BLEinterval = 15000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-21 21:24:01	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-22 12:52:35	Pass
9	Calculate tag's execution time	-	15 hours 28 minutes 34 seconds	Pass

Test Case ID	TagT8	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	1-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT8
2	WIFIinterval = 20000
3	BLEinterval = 15000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-01 23:14:34	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-02 14:02:42	Pass
9	Calculate tag’s execution time	-	14 hours 48 minutes 8 seconds	Pass

Test Case ID	TagT8	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	3.0

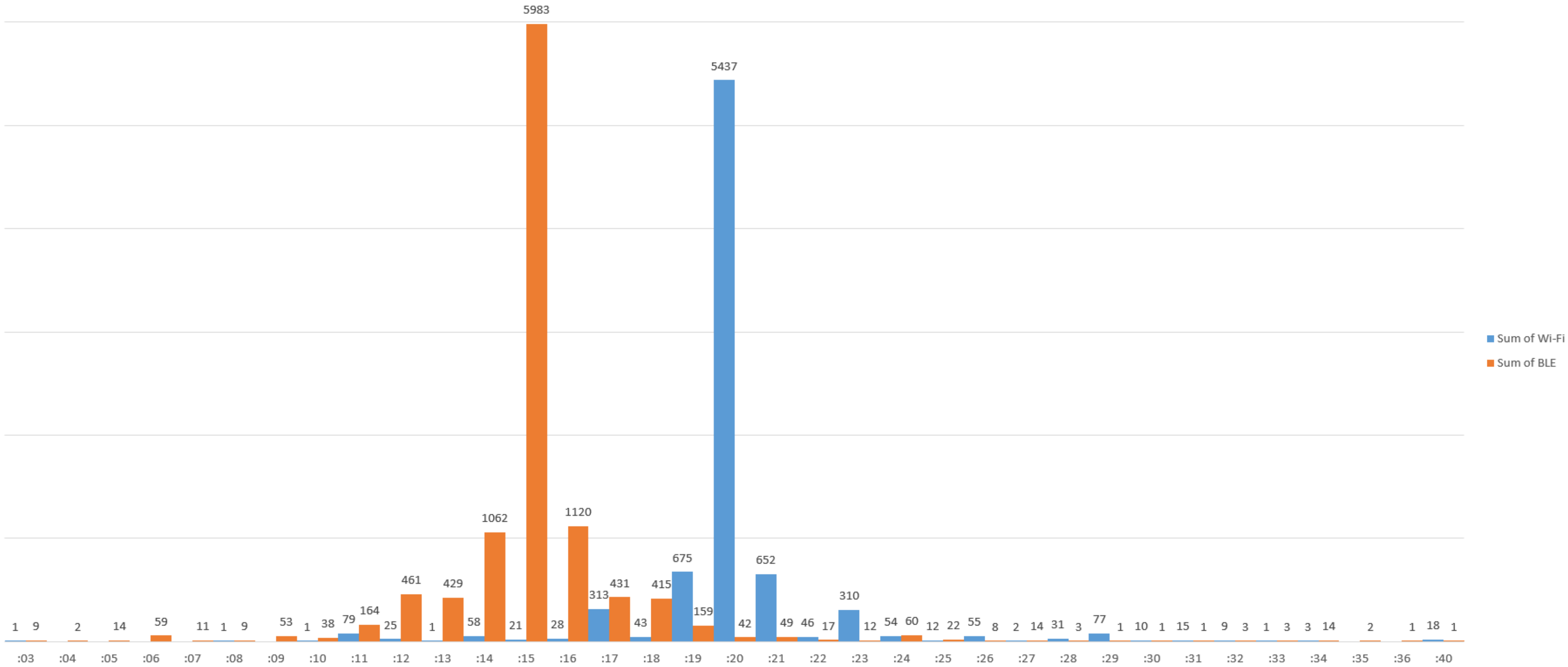
Date Tested	3-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT8
2	WIFIinterval = 20000
3	BLEinterval = 15000

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-03 09:54:48	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-04 00:18:31	Pass
9	Calculate tag's execution time	-	14 hours 23 minutes 43 seconds	Pass



5983
5437
1062
1120
675
652
461
429
431
415
313
310
164
159
1
9
2
14
59
11
1
9
53
1
38
79
25
1
58
21
28
43
42
49
46
17
12
54
60
12
22
55
8
2
14
31
3
77
1
10
1
15
1
9
3
1
3
3
14
2
1
18
1
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:24
:25
:26
:27
:28
:29
:30
:31
:32
:33
:34
:35
:36
:40
Sum of Wi-Fi
Sum of BLE

Test Case ID	TagT9	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	25-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT9
2	WIFIinterval = 0
3	BLEinterval = 20000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-25 12:40:00	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-25 18:53:46	Pass
9	Calculate tag’s execution time	-	6 hours 13 minutes 46 seconds	Pass

Test Case ID	TagT9	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	4-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT9
2	WIFIinterval = 0
3	BLEinterval = 20000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-04 14:17:02	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-04 19:23:14	Pass
9	Calculate tag’s execution time	-	5 hours 6 minutes 12 seconds	Pass

Test Case ID	TagT9	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	3.0

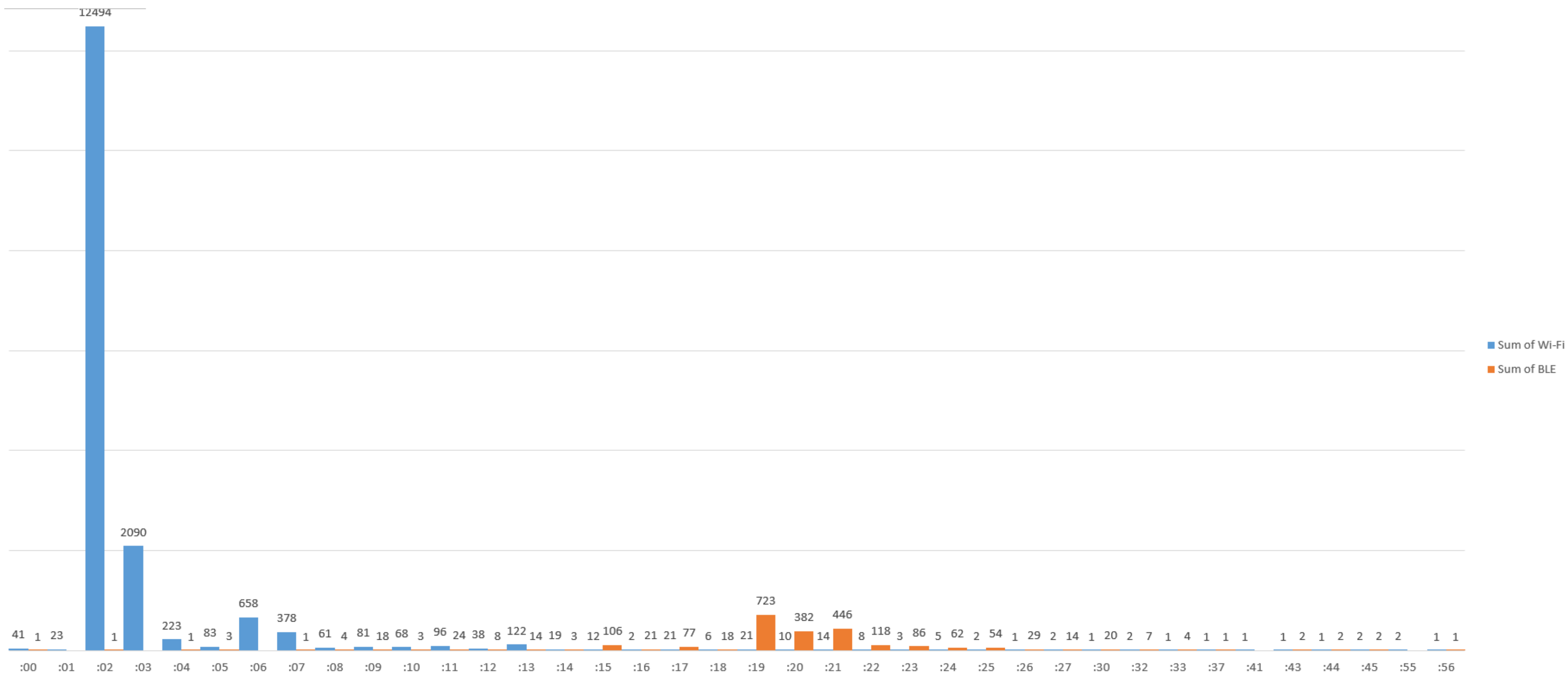
Date Tested	5-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT9
2	WIFIinterval = 0
3	BLEinterval = 20000

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-05 11:25:08	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-05 14:55:12	Pass
9	Calculate tag’s execution time	-	3 hours 30 minutes 4 seconds	Pass


Sum of Wi-Fi
Sum of BLE
41 1 23 12494 1 2090 223 1 83 3 658 378 1 61 4 81 18 68 3 96 24 38 8 122 14 19 3 12 106 2 21 21 77 6 18 21 723 10 382 14 446 8 118 3 86 5 62 2 54 1 29 2 14 1 20 2 7 1 4 1 1 1 1 2 1 2 2 2 2 1 1
:00 :01 :02 :03 :04 :05 :06 :07 :08 :09 :10 :11 :12 :13 :14 :15 :16 :17 :18 :19 :20 :21 :22 :23 :24 :25 :26 :27 :30 :32 :33 :37 :41 :43 :44 :45 :55 :56

Test Case ID	TagT10	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	27-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT10
2	WIFIinterval = 20000
3	BLEinterval = 0

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-27 11:39:13	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-27 21:39:15	Pass
9	Calculate tag’s execution time	-	10 hours 0 minutes 2 seconds	Pass

Test Case ID	TagT10	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	27-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT10
2	WIFIinterval = 20000
3	BLEinterval = 0

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-27 14:15:11	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-27 23:10:19	Pass
9	Calculate tag's execution time	-	8 hours 55 minutes 8 seconds	Pass

Test Case ID	TagT10	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	3.0

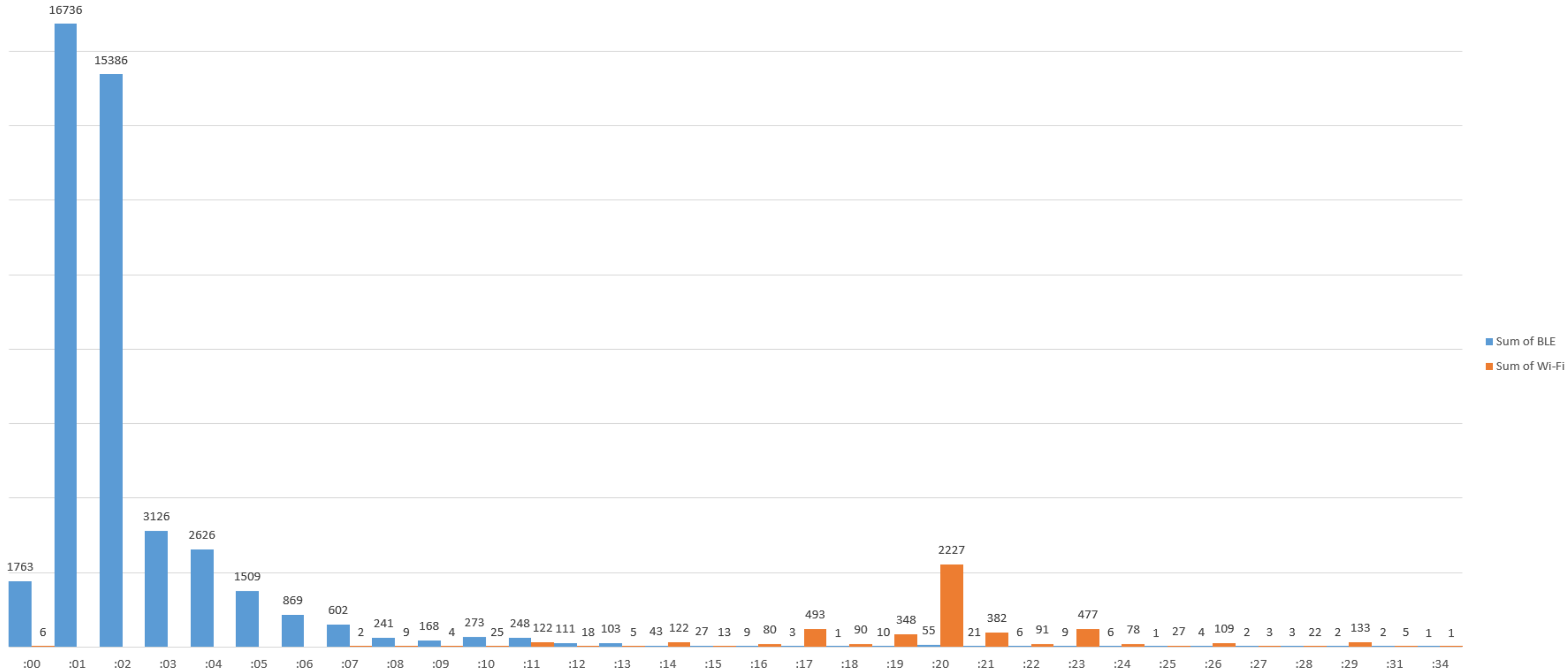
Date Tested	28-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT10
2	WIFIinterval = 20000
3	BLEinterval = 0

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-28 09:27:42	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-28 18:15:39	Pass
9	Calculate tag’s execution time	-	8 hours 47 minutes 57 seconds	Pass



16736
15386
3126
2626
1763
1509
869
602
2227
Sum of BLE
Sum of Wi-Fi
:00
:01
:02
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:24
:25
:26
:27
:28
:29
:31
:34

Test Case ID	TagT11	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	28-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT11
2	WIFIinterval = -1
3	BLEinterval = 0

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-28 18:58:21	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-29 04:53:22	Pass
9	Calculate tag’s execution time	-	9 hours 55 minutes 1 second	Pass

Test Case ID	TagT11	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	29-Sep-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT11
2	WIFIinterval = -1
3	BLEinterval = 0

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-29 09:26:44	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-29 18:18:35	Pass
9	Calculate tag's execution time	-	8 hours 51 minutes 51 seconds	Pass

Test Case ID	TagT11	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	3.0

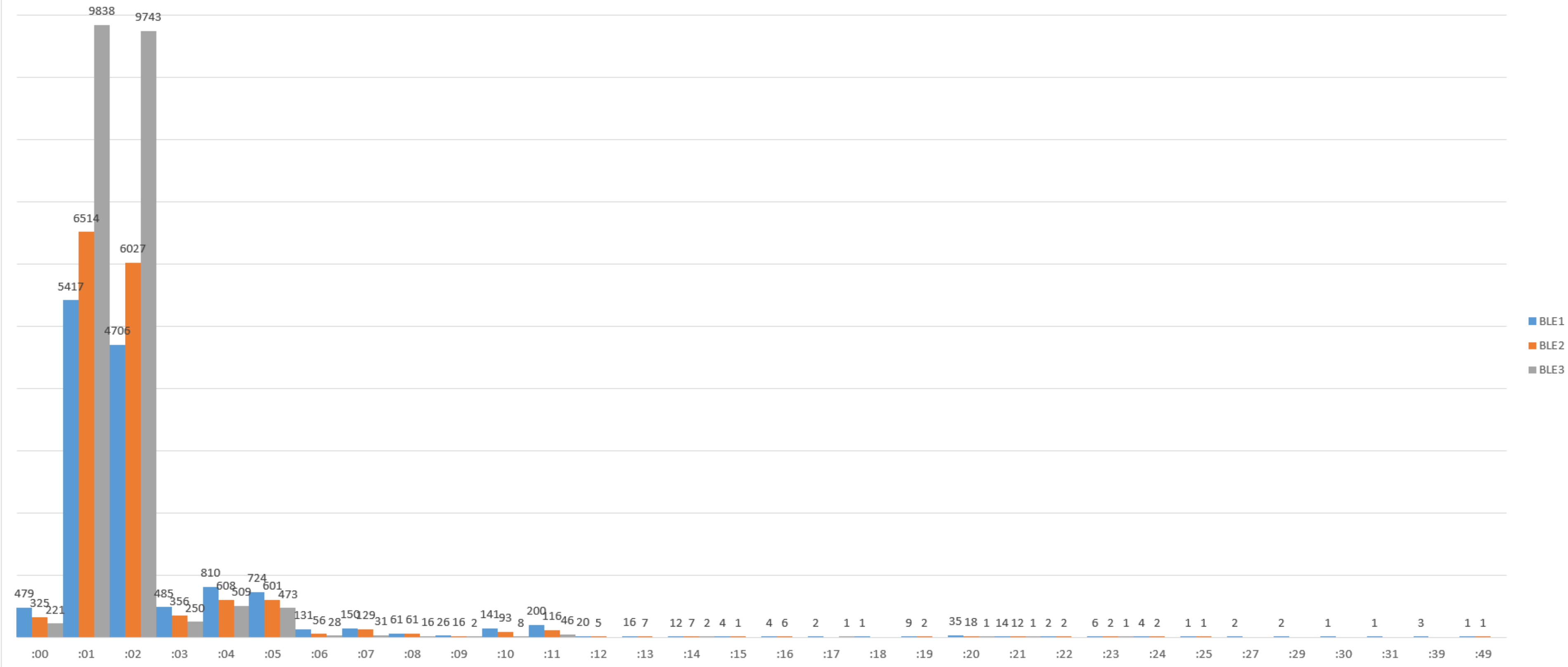
Date Tested	30 Sep	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT11
2	WIFIinterval = -1
3	BLEinterval = 0

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-09-30 06:42:40	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-09-30 15:13:40	Pass
9	Calculate tag’s execution time	-	8 hours 31 minutes 0 seconds	Pass



BLE1
BLE2
BLE3
:00
:01
:02
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:18
:19
:20
:21
:22
:23
:24
:25
:27
:29
:30
:31
:39
:49

Test Case ID	TagT12	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	30-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT12
2	WIFIinterval = 0
3	BLEinterval = -1

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-30 10:43:11	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-30 16:51:31	Pass
9	Calculate tag's execution time	-	6 hours 8 minutes 20 seconds	Pass

Test Case ID	TagT12	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	06-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT12
2	WIFIinterval = 0
3	BLEinterval = -1

Test Scenario A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-06 12:59:22	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-06 18:09:44	Pass
9	Calculate tag's execution time	-	5 hours 10 minutes 22 seconds	Pass

Test Case ID	TagT12	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	3.0

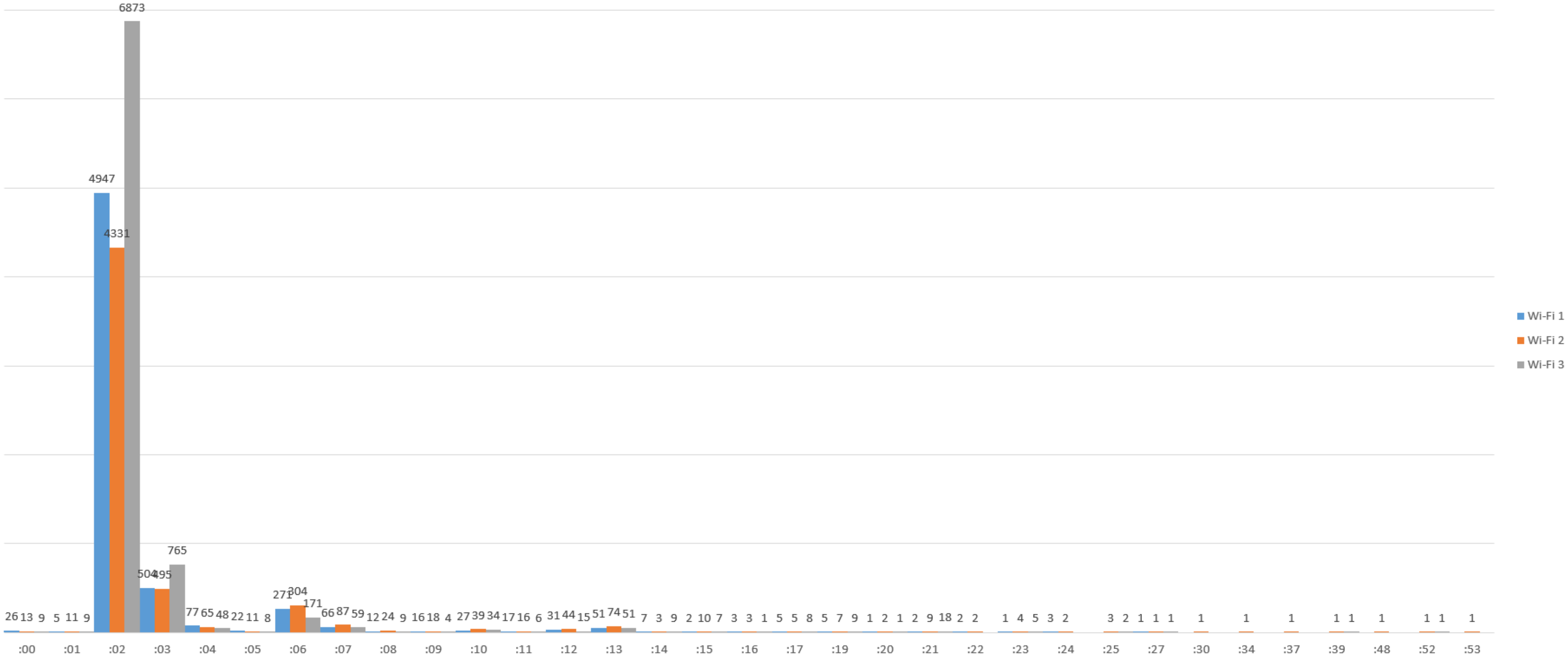
Date Tested	7-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT12
2	WIFIinterval = 0
3	BLEinterval = -1

Test Scenario: A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-07 10:49:39	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-07 15:50:46	Pass
9	Calculate tag’s execution time	-	5 hours 1 minute 7 seconds	Pass



Wi-Fi 1
Wi-Fi 2
Wi-Fi 3
:00
:01
:02
:03
:04
:05
:06
:07
:08
:09
:10
:11
:12
:13
:14
:15
:16
:17
:19
:20
:21
:22
:23
:24
:25
:27
:30
:34
:37
:39
:48
:52
:53

Test Case ID	TagT13	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	1.0

Date Tested	31-Aug-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT13
2	WIFIinterval = 0
3	BLEinterval = 0

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-08-31 13:59:32	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-08-31 21:13:13	Pass
9	Calculate tag’s execution time	-	7 hours 13 minutes 41 seconds	Pass

Test Case ID	TagT13	Test Case Description	Test the duration of the tag’s execution time powered with a powerbank
Created By	Pedro	Version	2.0

Date Tested	12-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT13
2	WIFIinterval = 0
3	BLEinterval = 0

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-12 08:11:51	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-12 14:16:28	Pass
9	Calculate tag’s execution time	-	6 hours 4 minutes 37 seconds	Pass

Test Case ID	TagT13	Test Case Description	Test the duration of the tag's execution time powered with a powerbank
Created By	Pedro	Version	3.0

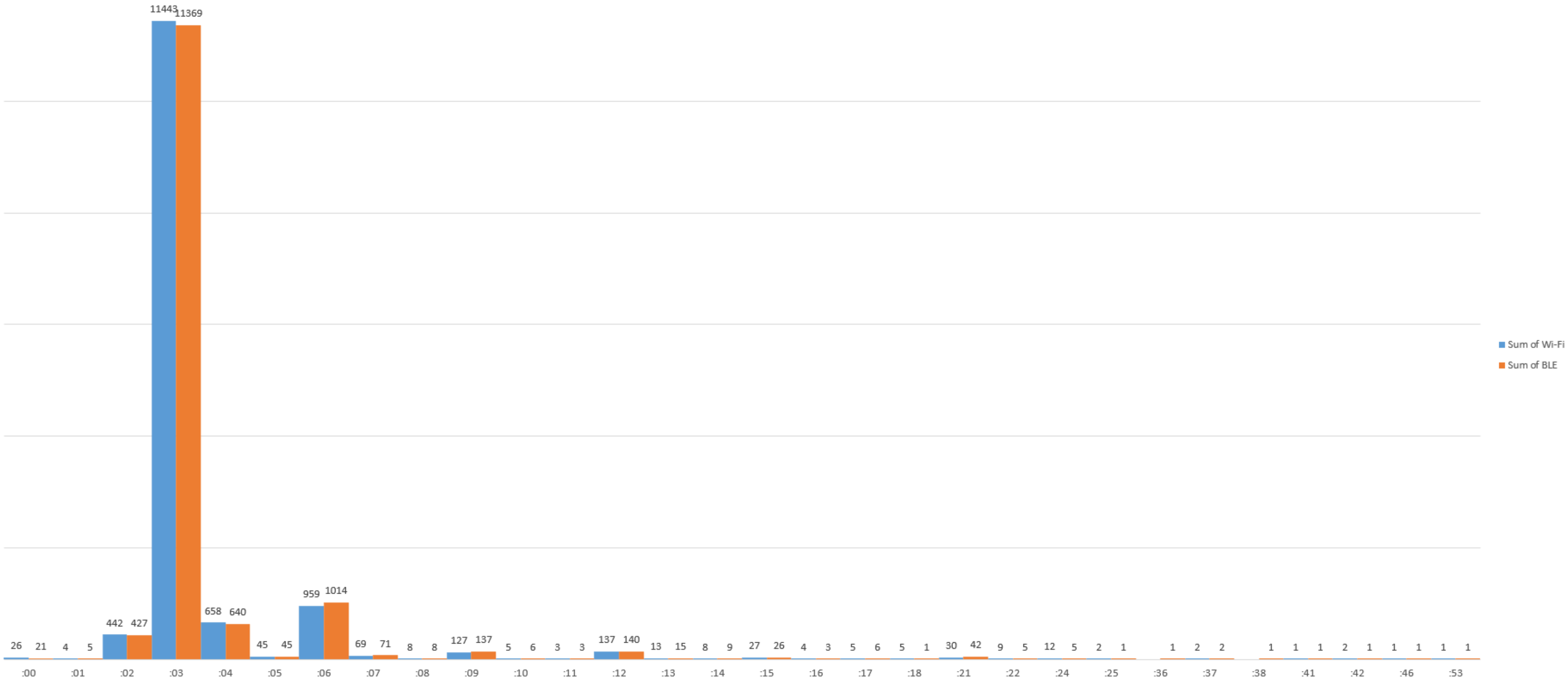
Date Tested	13-Oct-2022	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Powerbank fully charged
2	Tag data saved in the flash memory
3	Internet Access
4	Webserver online

S #	Test Data
1	TagName = tagT13
2	WIFIinterval = 0
3	BLEinterval = 0

Test Scenario	A powerbank-powered tag is left in an open environment with Internet Access, while the webserver is monitored for when the tag starts to post data and when it last posted data.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Connect the Tag to the Powerbank	The powerbank turns blue when it is working	As Expected	Pass
2	Navigate to http://ils.dsi.uminho.pt/viewData	Site should open	As Expected	Pass
3	Enter TagName	Credential can be entered	As Expected	Pass
4	Click Submit	Initial tag data is displayed	As Expected	Pass
5	Record initial tag data	-	2022-10-13 15:51:37	Pass
6	Wait until the powerbank is completely discharged	Powerbank stops displaying blue colour	As Expected	Pass
7	Click Submit	Final tag data is displayed	As Expected	Pass
8	Record final tag data	-	2022-10-13 21:31:08	Pass
9	Calculate tag's execution time	-	5 hours 39 minutes 31 seconds	Pass



11443
11369
442
427
658
640
959
1014
26 21 4 5 45 45 69 71 8 8 127 137 5 6 3 3 137 140 13 15 8 9 27 26 4 3 5 6 5 1 30 42 9 5 12 5 2 1 1 2 2 1 1 1 2 1 1 1 1 1
:00 :01 :02 :03 :04 :05 :06 :07 :08 :09 :10 :11 :12 :13 :14 :15 :16 :17 :18 :21 :22 :24 :25 :36 :37 :38 :41 :42 :46 :53
Sum of Wi-Fi
Sum of BLE