Improving accessibility for pedestrian crosswalks using digital fencing

Facundo Manuel Bustos Candisano - a52614

Dissertation submitted to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering.

As part of the double diploma program with the Universidad Tecnológica Nacional (UTN)

Scientific supervision:

Prof. João Paulo Coelho

Prof. José Alexandre Gonçalves

José Gonçalves, VALLED

Prof. Claudio J. Paz, UTN

Bragança

2022-2023

Improving accessibility for pedestrian crosswalks using digital fencing

Facundo Manuel Bustos Candisano - a52614

Dissertation submitted to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering
As part of the double diploma program with the Universidad Tecnológica Nacional (UTN) Scientific supervision:

Prof. João Paulo Coelho
Prof. José Alexandre Gonçalves
José Gonçalves, VALLED
Prof. Claudio J. Paz, UTN

Bragança
2022-2023

Dedication

I dedicate this work to my partner, Maria Luz Barone, to my mother, Ines Maria Candisano, my father, Rodrigo Francisco Javier Bustos Lourido, and my family. You have been my pillars and the foundation upon which my success is built. Your unconditional love, sacrifice, and unwavering belief in me have been the driving force behind my accomplishments. Your guidance, wisdom, and support have shaped me into the person I am today.

With love and gratitude,

Facundo

Acknowledgment

I would like to express my gratitude to everyone who have played a role in the completion of this work. Their guidance, support, and encouragement have been invaluable, and I am forever grateful.

I acknowledge the role played by my supervisor, Dr. João Paulo Coelho. Your feedback, patience, and willingness to share knowledge have been essential in broadening my perspectives and striving for excellence. Working under your supervision has been a valuable opportunity.

I would also like to thank the Polytechnic Institute of Bragança (IPB), where I conducted this research. The academic environment and resources provided by the university have been crucial in shaping my understanding and enhancing my research capabilities.

To my home university in Argentina, Technological National University (UTN), for providing me with a solid foundation and nurturing my intellectual curiosity. Also for giving me the opportunity to participate in the Dual Diplomacy program.

Last, but certainly not least, I would like to acknowledge the unwavering support of my family, my girlfriend and my friends. Their constant love, encouragement, and belief in my abilities have been a source of strength throughout this work. I am grateful for their patience during the countless hours I spent immersed in my research, and for their understanding during moments when I needed to prioritize my work. Their presence in my life has brought joy, balance, and inspiration, reminding me of the importance of pursuing my passions.

Abstract

The exponential growth of urban populations has put in agenda the need of cities to become more sustainable. The concept of *Smart Cities* can be an important part for the solution to this problem. One of the elements that can be found in this paradigm are smart crosswalks. Smart crosswalks can improve the access to such information by providing a way for the user to share data with them. Although they have many advantages, it is still difficult to adequately support people who are blind or visually impaired when they are crossing the street. The VALLPASS project developed a smart pedestrians crosswalk that, besides other design requirements, aims to promote accessibility by sharing local traffic data with the user. This thesis addresses this problem by presenting the development of a system that aims to provide a solution to promote accessibility of visually impaired citizens through the implementation of a digital fencing based on the user location obtained from the Received Signal Strength Indicator (RSSI) values between two beacons and the user's smartphone. That in a future could be integrated into the VALLPASS solution. In order to explore this solution it was developed a custom-made application in Android, and it was used an ESP32 as a Bluetooth Low Energy (BLE) Beacon.

Keywords: Pedestrians crosswalks accessibility; Digital fencing; Smart Cities ; Bluetooth Beacons; VALLPASS.

Resumen

El crecimiento exponencial de las poblaciones urbanas ha puesto en la agenda la necesidad de que las ciudades se vuelvan más sostenibles. El concepto de *Ciudades Inteligentes* puede ser una parte importante de la solución a este problema. Uno de los elementos que se pueden encontrar en este paradigma son los cruces peatonales inteligentes. Los cruces de peatones inteligentes pueden mejorar el acceso a dicha información al proporcionar una forma para que el usuario comparta datos con ellos. Aunque tienen muchas ventajas, todavía es difícil brindar un apoyo adecuado a las personas ciegas o con discapacidad visual cuando cruzan la calle. El proyecto VALLPASS desarrolló un cruce de peatones inteligente que, además de otros requisitos de diseño, tiene como objetivo promover la accesibilidad al compartir datos de tráfico local con el usuario. Esta tesis aborda este problema al presentar el desarrollo de un sistema que tiene como objetivo brindar una solución para promover la accesibilidad de los ciudadanos con discapacidad visual a través de la implementación de un cercado digital basado en la ubicación del usuario obtenida a partir de los valores de RSSI entre dos balizas y el teléfono del usuario. Para explorar esta solución, se desarrolló una aplicación personalizada en Android y se utilizó un ESP32 como baliza BLE.

Palabras clave: Accesibilidad de los cruces peatonales; Cercado digital; Ciudades inteligentes; Balizas Bluetooth; VALLPASS.

Contents

List of Tables

List of Figures

Acronyms

API Application Programming Interface.

APS Accessible Pedestrian Signals.

BLE Bluetooth Low Energy.

CPS Cyber-Physical Systems.

GAP Generic Access Profile.

GATT Generic Attribute.

GNSS Global navigation satellite system.

GPIO General Purpose Input/Output Interface.

GPS Global Positioning System.

ICT Information and Communication Technology.

IDE Integrated Development Environment.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

PDU Protocol Data Unit.

RF Radio frequency.

RSSI Received Signal Strength Indicator.

RTC Real-Time Clock.

SDK Software Development Kit.

SoC System-on-Chip.

UART Universal Acrychronous Receiver/Transmitter.

UI User Interface.

USB Universal Serial Bus.

UUID Universally Unique Identifier.

XML Extensible Markup Language.

Chapter 1

Introduction

This chapter presents the motivation of the work, objectives and the document structure adopted. Also it makes a quick overview of the *smart cities* paradigm.

1.1 Overview

Urban populations have been increasing since the 1st Industrial Revolution as more people started working in factories. But it was not until the 3rd Industrial Revolution that this phenomena has taken drastic contours. Over the past 50 years, the population living in cities has grown quickly, and by 2050, it is predicted that almost two-thirds of the world's population will live in urban areas [1]. This occurs simultaneously with what it is now called the 4th Industrial Revolution. This revolution is altering the paradigm of the world we live in as it introduces the concept of Cyber-Physical Systems (CPS). This approach integrates computational components (information processing) with physical processes, which exchange data through a network infrastructure. The technological advances in the Internet of Things (IoT), robotics, and autonomous vehicles are the foundation for making CPS possible [2].

Smart Cities has emerged as a solution to address the pressing need for cities to become more sustainable, efficient, and livable. *Smart cities* refers to a conceptual idea where urban areas use Information and Communication Technology (ICT) services or products that enhance the social and ethical well-being of its citizens, provide life quality, and interactivity of urban services. Promotion of sustainability is a key factor in *smart cities* by reducing costs, resource consumption and human footprint in the planet. Moreover, it also aims to promote contact between citizens and government [3]. To accomplish this, governments must apply different policies that optimize resource management, enhance sustainability, and improve the quality of life of their citizens, some of this solution would require the use of various types of sensors to gather precise data. The value of a *smart city* is not solely determined by the amount of technology it may have, but also by what it chooses to do with it.

Smart cities frequently rely on a network of interconnected devices that can communicate and exchange data. At the present, internet is ubiquitous and is the data network that supports the data collected by the devices and delivered it to cloud servers, where the data is analyzed in real time through the use of different algorithms. The data processing add intelligence to the system because it enables it to learn from the data and adjust to changing conditions.

The installation of smart crosswalks is one of the key elements in a *smart city*. Using a variety of technologies, smart crosswalks are intelligent traffic management systems that aims to improve pedestrian safety, reduce traffic congestion, and enhance overall pedestrian experience.

Although Smart Crosswalks have many advantages, it is still difficult to adequately support people who are blind or visually impaired when they are crossing the street. According to the most recent national censuses, in Portugal about 890,000 people have vision problems, of which 27,000 are totally blind [4]. The lack, or significant deprivation of the sense of sight, translates into challenges that many people must overcome and that are normally not perceived by the rest of the population. Public transportation plays an important role in providing ways for visually impaired, or blind persons, to travel independently. However, to take the bus, train or just move around the city on foot, it is necessary to walk through streets where, usually, both people and vehicles coexist. The sharing of space between cars and people poses safety challenges that must be overcome in order to avoid accidents that often end in fatalities. Security is even more critical on crosswalks since these are places where the probability of collision between vehicles and pedestrians is substantially higher. Knowing when and where it is safe to cross a given road is fundamental. Moreover, that information must be easily perceived by any pedestrian. In most cases, being able to see or hear greatly simplifies the process of crossing a road since, besides being able to easily grasp the traffic conditions, existent road signs, color or graphic codes require visual perception.

At the present, in Portugal, there is a normative that promotes the inclusion of passive solutions to help visually impaired persons to walk safely [5]. In particular, the addition of floor tactile clues to facilitate the location of the crosswalk. Other Accessible Pedestrian Signals (APS), including audible signals and crosswalk tactile contrasts, are also conveniences that greatly help the visually impaired navigate in urban scenarios. However, even if this normative can be seen as a positive point toward the democratization of road access, passive solutions, such as the ones enumerated above, lacks vital information to help the user in unfamiliar environments. Indeed, it is worth noticing that memory and familiarity with the surrounding environment is a fundamental assets used for navigation. When in familiar places, visually impaired people generally know the layout and memorize the location of obstacles. For example, the number of lanes, the crosswalk extension or even the traffic pattern. However, when in unusual places or with high and complex traffic variability, prior knowledge is unavailable or inapplicable. Moreover, even audible clues that exist in some crosswalks may not be easily perceived due to several causes such as the traffic and surrounding noise.

1.2 Motivation and work overview

It is in the context provided in Section 1.1 that this thesis is being carried out. In this work it is explored the inclusion of a digital fence based on Bluetooth signals in the VALLPASS solution. The VALLPASS project, funded by the Portuguese program Norte2020 under the grant NORTE-01-0247-FEDER-113439, developed a pedestrian crosswalk that could be integrated into the smart cities paradigm. In addition, it intends to include active systems that could promote an increase in crosswalks safety targeting a broader pedestrian universe of users which includes visually impaired persons. The VALLPASS project consortium is headed by the company VALLED and has as technological partners the research centres CeDRI and MORE.

To be able to fulfill this challenge , it is needed the development of an application with an interface capable of communication with the visually impaired that exchange data with the VALLPASS solution. In particular, using triangulation based on the signal strength of each transceiver existing in each pole, and through a dedicated smartphone application, the user will have the ability to assess the moment when it is safest to cross the road.

The situation illustrated on Figure 1.1 presents the typical disposition of the VALLPASS solution. In the situation illustrated in the figure, two VALLPASS units are installed sideways of a crosswalk. Each one of the VALLPASS elements includes a Bluetooth beacon that provides an Radio frequency (RF) signature able to be detected to a distance of up to 100 m. Assuming that a given pedestrian has a smartphone with the application installed, the RSSI of each Bluetooth beacon will be decoded by the software and the relative position of the pedestrian, regarding the crosswalk, is estimated.

1.3 Objectives

The aim of this thesis is to include active systems in addition to the existing work presented by the VALLPASS project, that could promote an increase in crosswalks safety targeting particularly visually impaired persons.

This work addresses the architecture of a protection digital fencing system based on Bluetooth signals that will be integrated into the VALLPASS smart crosswalk solution.

The system will implement Bluetooth beacons, and with the integration of a custom-made mobile application that will retrieve information of the beacons, such as the RSSI, it will be able to steer the pedestrian toward the crosswalk center and provide acoustic,

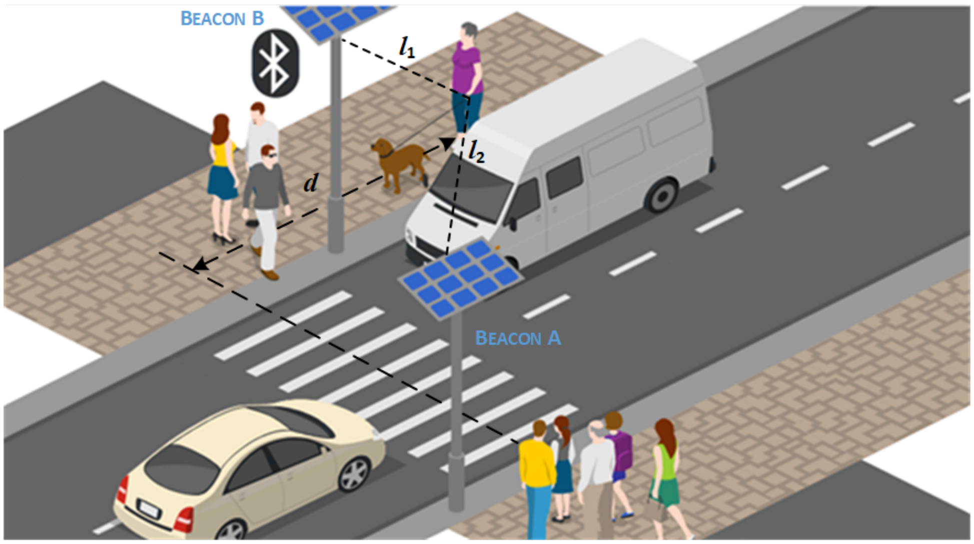

Figure 1.1: User localization using two Bluetooth beacons embedded on the VALLPASS smart pedestrian crosswalk.

voice or vibration codes according to the actual degree of security for traversing the road.

It is expected that the system provides a useful and reliable solution that could be integrated in the VALLPASS project in the future, and ensuring the accessibility for all pedestrians.

In the course of this work, one paper has been published and presented at the SASYR, and a second paper has been submitted for its approval.

1.4 Support

The project VALLPASS (vigilância ativa e inteligente com suporte LoRa para passadeiras), NORTE-01-0247-FEDER-113439 and to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021), for supporting this work.

1.5 Document Structure

This document is organized in 6 chapters, where the present chapter presents the contextualization, proposal and objectives of the work.

Chapter 2 presents a bibliographic review, which address fundamental contents necessary to understand the concepts and understanding of this work.

Chapter 3 provides an overview of Bluetooth Beacons, demonstrating the method adopted for distance measurement, and some basic characteristics about android application development.

Chapter 4 presents with details the proposed solution for this work, and it describes how it was developed.

Chapter 5 presents the measurements made to obtain the parameters of the system. Also, it presents the test made to evaluate the performance of the system and a discussion of the obtained results.

Finally, the last chapter outlines the main conclusions and points out future work directions.

Chapter 2

State of the art

This chapter presents the state-of-art, after a bibliographic review concerning the different scientific and commercial approaches for crosswalks accessibility aiming the visually impaired. It describes the most common approaches found in the literature that deal with this challenge.

2.1 Introduction

The idea of promoting safety for the visually impaired in urban environments is not a new topic. Accessibility systems in cities typically include a combination of infrastructure and technology-based solutions to help them navigate safely and independently.

The special difficulties experienced by visually impaired people in urban environments, where congested streets, busy junctions, and different impediments can represent significant hurdles to their movement and safety, are the focus of these accessibility systems. Cities may design spaces that are more inclusive and accommodative for people with vision impairments by combining infrastructure and technology.

Infrastructure is essential in fostering safety for people who are blind or visually impaired. For instance, the design of the sidewalks must be taken into account. Cities may make sure that sidewalks are wide, well-maintained, and devoid of any hazards or mobility-impairing obstructions like parked automobiles, garbage bins, or construction materials. Visually impaired people can navigate their environment more confidently and easily with the help of clear signage and tactile pavement.

Technology-based solutions have become effective instruments for improving accessibility for the blind in addition to infrastructure. The creation of intelligent crosswalks with auditory signals and vibrating surfaces is one example of this innovation. These technologies give visually challenged pedestrians real-time information so they can decide when it is safe to cross the roadway. These crosswalks, which are linked to traffic signals, make it possible for people with vision impairments to safely cross intersections even when there are no visual clues.

Collaboration between the visually impaired community, city planners, architects, and technology developers is necessary to advance safety for the blind and visually impaired in urban settings. Cities can design spaces that are more inclusive, safe, and accessible for everyone—regardless of visual abilities—by combining infrastructural upgrades with technological solutions. Urban environments must continually innovate and take into account different viewpoints to make sure that they actually meet the requirements of everyone, promoting a sense of autonomy, equality, and belonging.

2.2 Accessibility Systems

The most commonly used approach that can be find around the world is the one that can be defined as passive solutions. Examples of such approach are:

- Tactile cues: Sidewalks, and pedestrian crossings fitted with special paving designed to alert visually impaired people of their presence and guide them safely.
- Audio traffic signals: Traffic signals that are equipped with audio cues that provide audible indications of when it is safe to cross the road.

The main problem with this solutions is that the unfamiliarity with the environment where the person is located and even the traffic noise could make these approaches insufficient to give independence and safety to them.

Taking in consideration the drawbacks of the mentioned strategies, researchers is aiming for active solutions, that try to tackle down the main problems of them. Most of these solutions are found to fall into two categories: satellite based positioning (Global navigation satellite system (GNSS) or Global Positioning System (GPS)), and image processing.

The use of GNSS/GPS allows gathering both the global pedestrian location and where he or she is heading. This information, in conjunction with the map with the crosswalk locations, will be used to steer the person in the urban landscape. However, the use of GNSS/GPS as a guidance system must take into consideration the overall system precision which is around five meters under open sky [6]. Besides this error, the accuracy is even worse around buildings and bridges as satellite signals can be shadowed by them. Some of those drawbacks can be bypassed by using hybrid approaches that merge more than just one technology. For example, [7], designed a smartphone application that combines a digital map of the intersections and the user location and orientation obtained from the GPS receiver and the compass sensor build into the phone. For the cases where the GPS signal is weak, they placed Bluetooth beacons with a geo-ID tag to improve the information on the user's location. After arriving near the crosswalk, the application exchange messages with the traffic light controller, through Wi-Fi or LTE, asking how long the user must wait until is safe to cross the road. This information will then be relayed to the user using audio messages. It is worth pointing out that this approach requires the location of Bluetooth beacons since, in general, they are not integrated into conventional crosswalk signaling system. Moreover, it relies on the availability of an API to query the traffic system about its status which is not usually the case.

Another paper that explores the use of GPS for increasing the security of visually challenged persons was published by [8]. The authors have developed an application, named "Virtual Guide Dog" (VGD), which resorts to a GPS-based localization method to check when the user is near an intersection. If this is the case, the app will ask the pedestrian if he or she wants to cross the road and, if this is the case, the app will give orientations to steer the user. When the user is in position, the app connects via Bluetooth to the traffic light controller and acts as the physical process of pressing the crosswalk button. When it is safe to cross, the application indicates the user to start traversing. This solution suffers from the same issues as the previous solutions by relying on third-party technology to operate.

Other researchers approached this question through computer vision and image processing. That is the case of the Crosswatch project [9] which is a computer vision-based smartphone system developed to help visually impaired to detect traffic intersections. They first obtain the user location through the GPS sensor of the smartphone, although the GPS may not have a good precision for them is good enough to identify the intersection the user is standing in but not which corner, and certainly not the relative location to the crosswalk. Then they ask the user to rotate 360° to take pictures every 20° and obtain an panoramic image of the location, while they are doing that, they also take measurements of the accelerometer and magnetometer sensor for each photo to unwrap the panorama into an overhead view image centered at the estimated location of the user feet and then rotate them to align it with the geographic North and scale to obtain a relation in meters per pixel. Once the image is ready they obtain a second image from Google Maps of the intersection also aligned to the geographic North and with the same scale in meters per pixel of the previous one. With this two images, they run a computer vision algorithm to do a comparison between them and obtain the exact location of the user. That information later could be used as a base for future work to help to guide to user go through the intersection. However this image capture method can be too complicated to be performed. Another problem with this kind of solution is that they rely on the user smartphone and its capabilities of performing fast image processing to be able to guide the user in real-time.

The study of integration of wearable sensors instead of the user phone, to obtain real-time images, is another approach that can be found in the literature. For instance [10], proposed the design of a smartphone application where the user needs to wear an especial sunglasses equipped with a camera. First they use the GPS sensor to locate the user. Once it detects that he or she is around a crosswalk it triggers the camera and send a picture to a cloud server. In there, they run an image processing algorithm that determines whether is safe to cross. If the result is positive the application notify the user to start crossing. If not, it repeats the algorithm until it gets a positive result. This method heavily relies in the response time of the cloud server and the quality of the service available by the visually impaired.

In 2017, [11] designed an algorithm in conjunction based on adaptive extraction and consistency analysis for real-time image processing in order to detect crosswalks at urban intersections. This algorithm is feed with real time video taken from a wearable system that is composed of a camera, a pair of bone-conducting earphones, and a portable computer. Once the algorithm detected the crosswalk, it instructs the user to align himself with the crosswalk. Further advances in the same work, in 2020 [12] proposed an unified system where the algorithm not only detect crosswalks and guides the user to them, but also read traffic light status to guarantee the safety of the user. It also integrate a feature called "scene recognition" to tell the user where he or she is located on the environment that the googles are able to see. In 2017, they released a commercial version of this product [13].

Also on the computer vision area, [14] proposed the implementation of a wearable system for crosswalk navigation. The system operates in real-time and utilizes prior maps, similar to how autonomous vehicles are guided. Computer vision algorithms locates the user in the map and orient it toward the crosswalk. The state of the crosswalk signal is detected using a convolutional neural network and the user is notified when it is safe to cross. Then the user is guided along a path towards a destination on the prior map. The system continually updates the user position relative to the path and corrects the user's trajectory with simple verbal commands. This solution rely on the use of a RGB-D camera and a LiDAR sensor to provide the real-time data to be processed by the portable computer with the detection algorithm.

The use of wearables always poses problems that span from the need to acquire a specific item that, due to its particular application and target audience, can be very expensive or even not within the person's sense of fashion and personal style. Moreover, at night, the cameras may not be able to correctly detect the crosswalk.

Following on the wearable technology, [15] in his master thesis developed an integrated system called "Smart-Cane" to increase safety for the visually impaired. He proposes the deployment of RFID tags on the crosswalk, each one contains information about its location relative to a crosswalk. When the smart-cane reads them, it sends a message through the Raspberry-Pi to the user smartphone giving them instructions about how to position themselves to safely cross. A second subsystem consists of a driver alert system. When the smartphone app detects that the user is crossing it sends a message through LTE to a Message Queue Telemetry Transport (MQTT) server, where nearby drivers can access this information and receive an alert message. Moreover, the "green time system" which communicates with the traffic lights controller, asks permission for allocating pedestrians green light time. In addition, it provides extra green light time for pedestrians to complete crossing safely and accordingly when needed. This system may be too expensive to do a successful installation as it used 272 tags for a single crosswalk, also the need of the drivers to have the application to receive the alert it's not a plausible option as the drivers can't be forced to install it.

Kiyoung et al. [16] presented the "Crossing Assistance System" where a location is performed through machine learning using the Received Signal Strength Indication (RSSI) from eight Bluetooth beacons located at a four-corner car intersection with four crosswalks (two beacons at each side of a crosswalk). In particular, a smartphone app takes the measured RSSI of each beacon and then the machine learning algorithm computes the user location. In this setup, the beacons send data every half second and the smartphone app is built in order to receive two RSSI signals per second provided by the eight beacons. As they have large amounts of data, and the signals can be sensitive to noise due to the traffic, they implemented the moving average filter to achieve better results. According to the authors, if they give the algorithm a three seconds windows to process the data they can provide the user location with a $99,8\%$ accuracy. However, their work doesn't implement any guidance system as their main focus was developing the location algorithm. Although they mention it as a potential future work. In addition, Bluetooth beacons must be installed on third party systems which can be challenging.

Besides academic solutions that address this problems, there are commercial approaches for this challenge.

Sinowatcher [17], developed two two distinct products with this objective. One is called "Smart Pedestrian Warning Bollards", and the other is identified as "Smart Pedestrian Crossing Warning Totem". The first one consists of a pair of stakes that, together, create a photoelectric barrier using pairs of infrared emitters/receivers, responsible for detecting pedestrians' crossing attempts. When the pedestrian interrupts the barrier after a certain amount of time it enables the pedestrian crossing mode, where visual and acoustic information is provided to pedestrians. Additionally, it also provides visual indication to vehicles of the presence of people on the crosswalk. The second solution is significantly more advanced as it integrates image processing and analysis technology, allowing for the detection of road code violations, detection of pedestrian flow direction, and integration with existing traffic light systems.

Euroasfalt [18] is a Polish company that, among other products and services, they offer a smart crosswalk solution called "APC Zebra". This device enhances the functionalities of the passive signage typically found on crosswalks by incorporating motion-reactive elements. The system comes into operation when a pedestrian approaches the crossing area and activates the motion sensors. At that moment, both the pavement lighting and the vertical signs are activated. They also mention the existence of a device called "SmartPass", which, in addition to pedestrian detection, enables the generation of sound messages, measurement of vehicle speeds approaching the crosswalk, and transmission of information and statistics via GSM.

All the solutions explored in this chapter show the ongoing interest in developing systems that can be inclusive for all the pedestrians, but still there is a room for improvement and further development in this subject.

Chapter 3

Materials and Methods

This chapter presents the technological tools used for the development of this work. Section 3.1 explores the concept of the BLE technology and the Bluetooth beacons. Then, in Section 3.3, it shows how to calculate using the RSSI. Later Section 3.4 introduces the filtering method used for eliminate the uncertainty in the measurements. Section 3.5 presents the developing of an Android application and the programming languages which can be used for building it.

3.1 Bluetooth Low Energy

Today smartphones come equipped with Bluetooth technology, making it widely accessible to nearly everyone. The project aims to utilize this technology for communication between the beacons installed in the VALLPASS crosswalk, and the pedestrians phones.

According to the Institute of Electrical and Electronics Engineers (IEEE):

> "Bluetooth is a standard for short range, low power, and low cost wireless communication that uses radio technology."[19]

Over 2100 companies around the world already support Bluetooth technology, and the wireless personal area network (WPAN) technology, based on the Bluetooth specification, is now an IEEE standard under the denomination of 802.15 WPANs.

This work relies on the Bluetooth Low Energy (BLE), which is an improvement of the standard Bluetooth technology. It allows the communication between two devices that does not need large amount of data to be transmitted between them. This technology also supports a broadcast mode which allows one device to transmit data to an unlimited number of receivers simultaneously.

Table 3.1 shows a comparative between this two technologies.

Specifications	**Bluetooth Classic**	**BLE**
Range	100 m	$> 100\ m$
Data Rate	$1-3\ Mbps$	1 $Mbps$
Frequency	$2,4\ GHz$	$2,4\ GHz$
Latency	100 ms	6 ms
Power Consumption	1 W	$0,01$ to $0,5\ W$
Peak Current	$< 30\ mA$	$< 15\ mA$

Table 3.1: Comparison between Bluetooth Classic and BLE.

Some uses of this technology are:

- Health monitors
- Fitbit devices
- Industrial monitoring sensors
- IoT applications

Its power consumption is very efficient as the device remains in sleep mode unless a connection has been made. As the data transfer rates are high and there isn't large amount of data to send, this connections usually last a few milliseconds allowing devices which ran on batteries for days or even weeks of autonomy [20].

3.2 Bluetooth Beacons

Bluetooth beacons are transmitter devices that broadcast a BLE signal to the nearby mobile devices. This technology enables smartphones, computer, tablets, and other devices to perform actions when in close proximity to one of them.

The Bluetooth beacons transmits an Universally Unique Identifier (UUID) signal that can be picked up by a compatible device. This can be used to determine the device's physical location, track objects, or trigger an event on the device based on the location of some points of interest such as a bus stop or a crosswalk.

One of the main differences between Bluetooth beacons and other location-based technologies is that the broadcasting device (beacon) is an unidirectional transmitter: it can not receive data. Thus, it is needed that the receiving devices have a specific application or system to interact with the beacon. This way the beacons can not collect data about the nearby users, thus they are unable to track the user location[21].

The hardware of a beacon is compact and usually consist of: a microcontroller with a BLE radio chip, and a power source. Additionally, there are beacons that have some sensors and general peripherals. There are multiple manufacturers of BLE transceivers such as Texas Instruments, Nordic Semiconductors, Dialog Semiconductors, and Cypress among others. The power source used on those beacons varies between vendors and applications. There are coin cell batteries allowing the beacon device to have very small dimensions, at the cost of reduced battery life. On the other hand, standard AA batteries can be used to drastically improve battery life, at the cost of larger dimensions. The beacons installed in an outdoor area commonly uses solar-power to recharge their batteries. When external power is required, power outlet and Universal Serial Bus (USB) are popular choices.

Each beacon has a specific firmware that makes use of the available hardware. The firmware controls the characteristics that impact the total power consumption:

1. **Transmission power (Tx Power):** The beacons advertises with a fixed transmission power. It is represented in dBm (decibel-milliwatts) that is a unit used to indicate that a power level is expressed in decibels (dB) with reference to $1mW$. As the signal travels trough the air the signal strength decreases. With a higher Tx power, the signal can travel longer distances, but this impact in a shorter battery

life [22].

2. **Advertising interval:** This is the frequency in which the beacon advertises. An interval of 100ms means that the signal broadcasts 10 times in a second. Longer interval will lead to a slower battery drain, at the cost of a higher latency on the receptor device.

Figure 3.1 exemplifies how the energy consumption changes over three transmission power levels and three advertising intervals for a BLE beacon. It is clear that the power consumption is proportional to the transmission power and inversely proportional to the advertising interval[23].

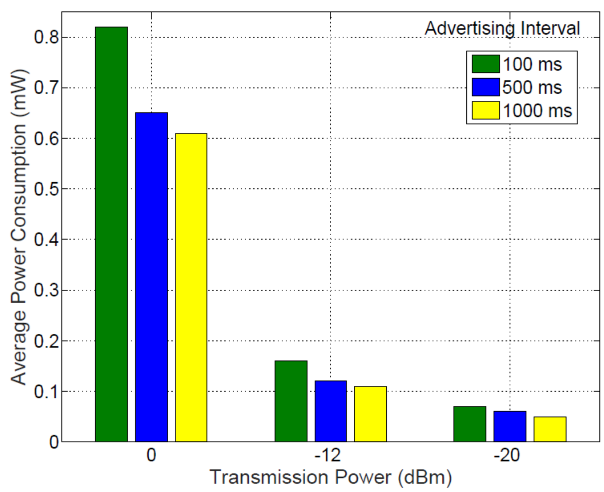

Figure 3.1: Average power consumption under different advertising intervals and transmission power levels [23].

The BLE protocol allows two operating modes: connected and advertising mode. The connected mode uses the Generic Attribute (GATT) profile to transfer data [24]. On the other hand, the advertising mode uses the Generic Access Profile (GAP) to send data to anyone who is listening. The BLE beacons takes advantage of the advertising GAP mode to broadcast data periodically[25]. The advertising signals contains a small data payload, also know as PDU. Each type of beacon uses a custom specification to partition the advertising data. Below are some of the existing protocol:

1. **iBeacon:** It is a protocol created by Apple. This packet structure enables convenient identification of individual beacon devices, using the UUID, major and minor values. Figure 3.2(a) shows the advertising PDU of iBeacon [26].

2. **Eddystone:** Is the Google standard for BLE beacons. It supports three types of packets, Eddystone-UID, Eddystone-URL, and Eddystone-TLM. Eddystone-UID functions in a very similar way to Apple's iBeacon. The working principle of a URL frame is similar to the conventional QR code, whereas a TLM frame allows developers to provide additional data regarding the deployed beacon. Figure 3.2(b) shows the advertising PDU of Eddystone. [27]

<table>
<tr><td colspan="4">(a) Adv PDU</td><td colspan="5">Payload defined by iBeacon Standard</td></tr>
<tr><td>1 byte</td><td>4 bytes</td><td>2 bytes</td><td>6 bytes</td><td>9 bytes</td><td>16 bytes</td><td>2 bytes</td><td>2 bytes</td><td>1 byte</td></tr>
<tr><td>Preamble</td><td>Access Address</td><td>Header</td><td>MAC</td><td>iBeacon Prefix</td><td>Universally Unique Identifier (UUID)</td><td>Major</td><td>Minor</td><td>Tx Power</td></tr>
</table>

<table>
<tr><td colspan="4">(b) Adv PDU</td><td colspan="7">Payload defined by Eddystone Standard</td></tr>
<tr><td>1 byte</td><td>4 bytes</td><td>2 bytes</td><td>6 bytes</td><td rowspan="2">UID</td><td>1 byte</td><td>1 byte</td><td colspan="2">16 bytes</td><td colspan="2">2 bytes</td></tr>
<tr><td rowspan="5">Preamble</td><td rowspan="5">Access Address</td><td rowspan="5">Header</td><td rowspan="5">MAC</td><td>Frame Type</td><td>Ranging</td><td colspan="2">UID</td><td colspan="2">Reserve</td></tr>
<tr><td rowspan="2">URL</td><td>1 byte</td><td>1 byte</td><td colspan="4">18 bytes</td></tr>
<tr><td>Frame Type</td><td>Ranging</td><td colspan="4">URL</td></tr>
<tr><td rowspan="2">TLM</td><td>1 byte</td><td>1 byte</td><td>2 bytes</td><td>2 bytes</td><td>4 bytes</td><td>4 bytes</td></tr>
<tr><td>Frame Type</td><td>TLM Version</td><td>Battery Level</td><td>Temperature</td><td>ADV_CNT</td><td>SEC_CNT</td></tr>
</table>

Figure 3.2: Advertising PDU of (a) iBeacon and (b) Eddystone [28].

3.3 Distance measurement

The average signal power between a transmitter and a receiver decreases logarithmically with distance. The RSSI is a measurement of the strength of the incoming (received) signal in a receiver. The smaller this value, the less the attenuation in the signal transmission. One of the most commons methods of RSSI ranging, and the one used in this thesis, is the logarithmic normal (shadowing model) [29].

The logarithmic normal distribution describes the random shadowing effects which occur due to the environment along the propagation path [29].

$$P_L(d) = P_L(d_0) + 10n\log(\frac{d}{d_0}) + \chi_\sigma \tag{3.1}$$

In the Equation 3.1 we have the following therms:

- $P_L(d)$ is the path loss value (in dBm) of the received signal at the distance d in meters.
- $P_L(d_0)$ is the path loss value (in dBm) at a reference distance d_0.
- n is the path loss exponent for a specific environment (see Table 3.2).
- χ_σ is a cover factor for the uncertainties of the model. Is a zero-mean Gaussian distributed random variable (in dB) with standard deviation value of σ.

Environment	**Path Loss Exponent** n
Free Space	2
Urban area cellular radio	$2,7$ to $3,5$
Shadowed urban cellular radio	3 to 5
Inside a building - Line of Sight	$1,6$ to $1,8$
Obstructed in building	4 to 6
Obstructed in factory	2 to 3

Table 3.2: Path loss exponent for different environments [29].

The signal strength at the receiving node is equal to the difference between the signal transmission power and the losses on the path at a distance d:

$$RSSI = P_t - P_L(d) \tag{3.2}$$

In (3.2):

- P_t is the signal transmission power (in dBm).
- $RSSI$ indicates the signal strength which is received from reference nodes at the distance d.

In (3.3), it is presented the reference RSSI value "A" from the reference nodes at the distance d_0

$$A = P_t - P_L(d_0) \tag{3.3}$$

Then replacing (3.2) and (3.3) in (3.1):

$$RSSI = A - 10n\log(\frac{d}{d_0}) - \chi_\sigma \tag{3.4}$$

If we take as reference $d_0 = 1m$, and knowing that χ_σ has a mean value of 0, if we consider $\overline{RSSI}$ as the RSSI of multiple times of measurements we can despise this value [30].

$$\overline{RSSI} = A - 10n\log(d) \tag{3.5}$$

And solving the equation for the distance d as is the desired value that it is needed:

$$d = 10^{(A-RSSI)/10n} \tag{3.6}$$

3.4 RSSI filtering

As discussed in Section 3.3, to obtain the distance to a beacon using (3.6) we need to eliminate the uncertainty generated by the random variable χ_σ. Various authors have approached different methods to achieve this goal. The two main methods are:

1. Moving Average: The moving average filter is an statistic filter that works by gathering n RSSI samples and averaging the values to create a new value. Mathematically, is the convolution between the signal and the filter impulse response and works as a low-pass filter.

2. Kalman Filter: The Kalman filter is a recursive algorithm that uses a series of measurements to estimate the next value based on its previous values.

Based on the work done by [31] where they used BLE beacons for indoor positioning, they tested the moving average and Kalman filtering methods and reached the conclusion that both filters produced similar results. Based on their results, in this work it was implemented the moving average filter as it is quicker to implement.

The implementation of this filter can be seen in the Equation 3.7:

$$\overline{RSSI} = \frac{1}{n} \sum_{i=k-n+1}^{k} RSSI_i \tag{3.7}$$

3.5 Application development

When it comes to operating systems for smartphones, there are several options available, each with its own strengths and weaknesses. Two popular choices are Android and iOS. Android, developed by Google, is an open-source mobile operating system based on the Linux kernel, designed to runs on touchscreen devices such as smartphones and tablets. On the other hand, iOS, developed by Apple, is a closed-source operating system exclusive to Apple devices. Both operating systems offer a range of features and advantages that cater to different preferences and needs. Android, being open-source, provides greater flexibility, customization options, and compatibility with a wide range of devices. Its open nature allows developers to access low-level system features and hardware. In contrast, iOS offers a seamless user experience, polished interface, and strict security measures. Its closed ecosystem ensures compatibility across Apple devices, providing a cohesive and optimized user experience. However, the closed nature of iOS may limit customization options and restrict access to certain system features.

The choice of Android as development platform is because it has more flexibility when it comes to programming hardware features, and its larger market share means that more users are likely to have compatible devices.

The development of applications on Android is made using the Android Software Development Kit (SDK) and the **Kotlin** programming language, which replaced Java as Google's preferred language [32]. The SDK includes a comprehensive set of development tools, such as: a debugger, software libraries, an emulator, documentation, sample code, and tutorials. Google's supported Integrated Development Environment (IDE) is Android Studio.

Android Studio is an IDE that is specifically designed for Android app development. The IDE provides a full suite of tools for developing, testing, and deploying Android applications. Android Studio uses the **Kotlin** or **JAVA** programming language and provides advanced features such as code completion, debugging, and version control.

App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter the app.

There are four types of app components:

1. **Activities:** Is the entry point for interacting with the user. It represents a single screen with a user interface.

2. **Services:** Is a general-purpose entry point for keeping an app running in the background. A service does not provide a user interface.

3. **Broadcast receivers:** Is a component that lets the system deliver events to the app outside of a regular user flow so the app can respond to system-wide broadcast announcements.

4. **Content providers:** A content provider manages a shared set of app data that can be stored in the file system, in a SQLite database, on the web, or on any other persistent storage location that the app can access. Through the content provider, other apps can query or modify the data, if the content provider permits it.

Each type serves a distinct purpose and has a distinct lifecycle that defines how a component is created and destroyed.

A unique aspect of the Android system is that any app can start another app's component. For example, if the user want to capture a photo with the device camera, there's probably another app that does that—and the app can use it instead of developing an activity to capture a photo. For that, it can start the activity of the camera and when the task is completed, the photo is returned so it can be used in the app. From the user perspective, it seems as if the camera is actually a part of the app.

When the system starts a component, it starts the process for that app, if it's not already running, and instantiates the classes needed for the component. For example, if the app starts the activity in the camera app that captures a photo, that activity runs in the process that belongs to the camera app, not in the app's process. Therefore, unlike apps on most other systems, Android apps don't have a single entry point: there's no main() function [33].

The *Activity* class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms, in which apps are launched with a *main()* method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle.

The user interaction with a mobile-app it's non-deterministic, differing from a desktop app, as for example the user can open an email app from the home screen, he or she might see a list of emails. By contrast, if they are using a social media app that then launches the email app, they might go directly to the email app's screen for composing an email.

The Activity class is designed to facilitate this paradigm. When one app invokes another, the calling app invokes an activity in the other app, rather than the app as an atomic whole. In this way, the activity serves as the entry point for an app's interaction with the user.

As a user navigates through, out of, and back to the app, the Activity instances in the app transition through different states in their lifecycle. The Activity class provides a number of callbacks that let the activity know when a state changes or that the system is creating, stopping, or resuming an activity or destroying the process the activity resides in [34].

Most apps contain multiple screens, which means they comprise multiple activities. Typically, one activity in an app is specified as the main activity, which is the first screen to appear when the user launches the app [35].

To use activities in the app, information about them must be registered in the app's manifest, and the activity lifecycles must be managed appropriately.

Figure 3.3 shows a diagram of the activity lifecycle and it's callback methods.

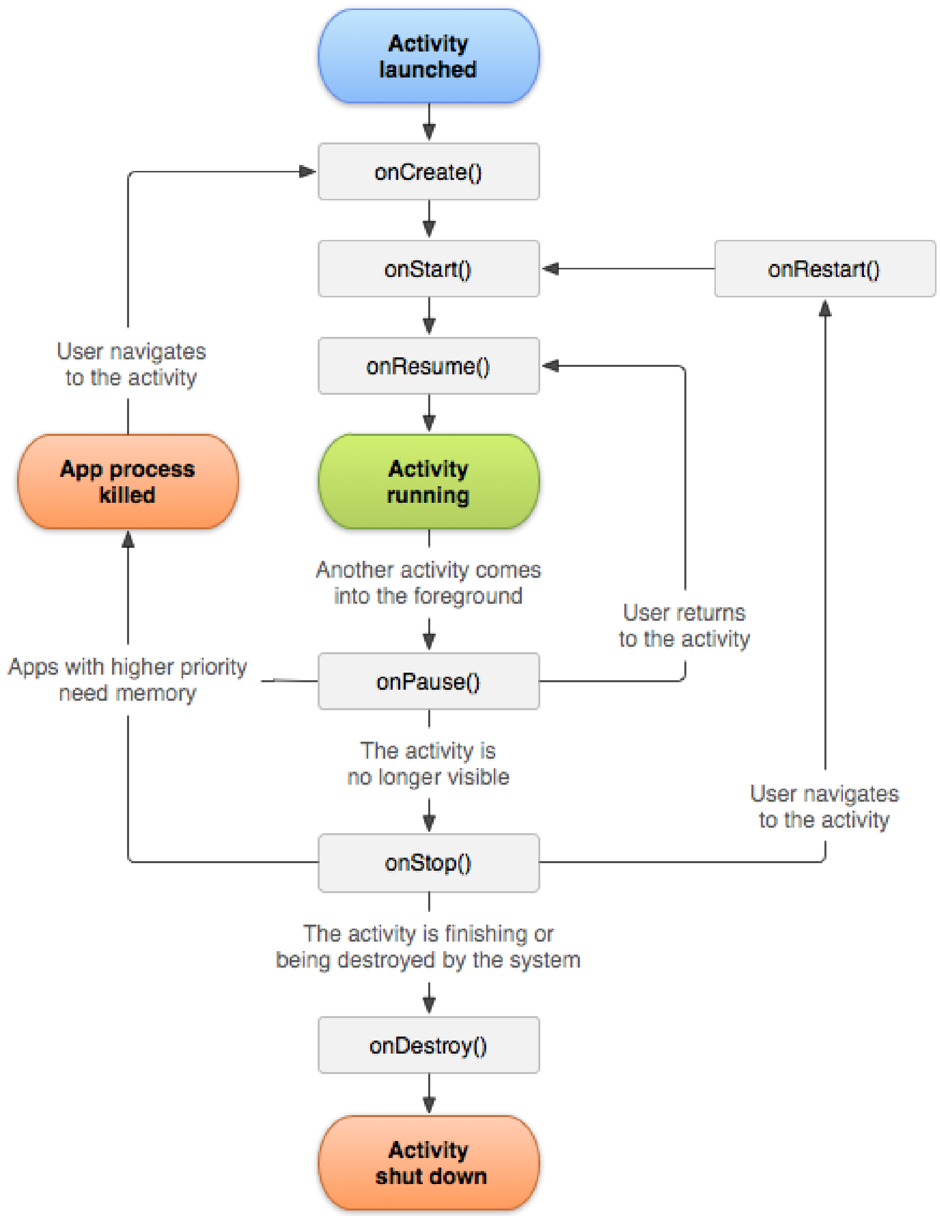

Figure 3.3: The activity lifecycle [34].

Chapter 4

The Developed System

This chapter presents the approach taken along this work. Starting with the Section 4.1, the solution adopted for the problem is presented. Then, in Section 4.2, an overview of the system hardware will be provided. Section 4.3 explores the software architecture for the BLE beacons. Lastly, Section 4.4 describes the development for the smartphone application in Android.

4.1 Proposed solution

In the VALLPASS smart crosswalk solution, there are two VALLPASS units installed sideways of a crosswalk. Each one of the VALLPASS elements includes BLE beacons, wich will be configured with the *iBeacon* protocol that broadcast a RF signal able to be detected by every device that is near to it. Assuming that a given pedestrian has a smartphone with the VALLPASS mobility application installed, the advertised PDU and the RSSI of each beacon will be decoded by the software and the relative position of the pedestrian, regarding the crosswalk, is estimated. Figure 4.1 shows a VALLPASS unit that has been installed in the Avenida Cidade de León, Bragança.

It is worth noticing that, in practice, location by triangulation requires, at least, three points. However, in a typical crosswalk application, only two VALLPASS units will be available. Hence, there is an intrinsic uncertainty if the pedestrian is upstream

Figure 4.1: VALLPASS units installed in the Avenida Cidade de León, Bragança.

or downstream of the crosswalk. However, this issue is not fundamentally a problem since the location algorithm will be concerned with the relative peak power and disregard if the pedestrian is approaching the center of the crosswalk from the left or from the right. Moreover, positioning is not the main objective of this system. Indeed, pedestrian positioning can be fine-tuned by himself with the help of the tactile cues embedded on the sidewalk. Being able to provide accurate information on the actual crosswalk and traffic status while providing decision support regarding the most secure time interval to cross the road is the key feature of the VALLPASS accessibility system.

The estimated distance between two Bluetooth devices using the RSSI value is obtained in (3.6). With this approach, the distance l_1 between "Beacon B" of Figure 4.2

and the user smartphone can be estimated. In the same way, the distance l_2 between "Beacon A" and the user's handheld device can also be inferred. Once the distance between the pedestrian and each of the two beacons is found, the distance of the user to the center of the crosswalk can be estimated by establishing a circle for each beacon with a radius equal to its distance to the user. Then two intersecting points can be obtained, Figure 4.2 presents this concept.

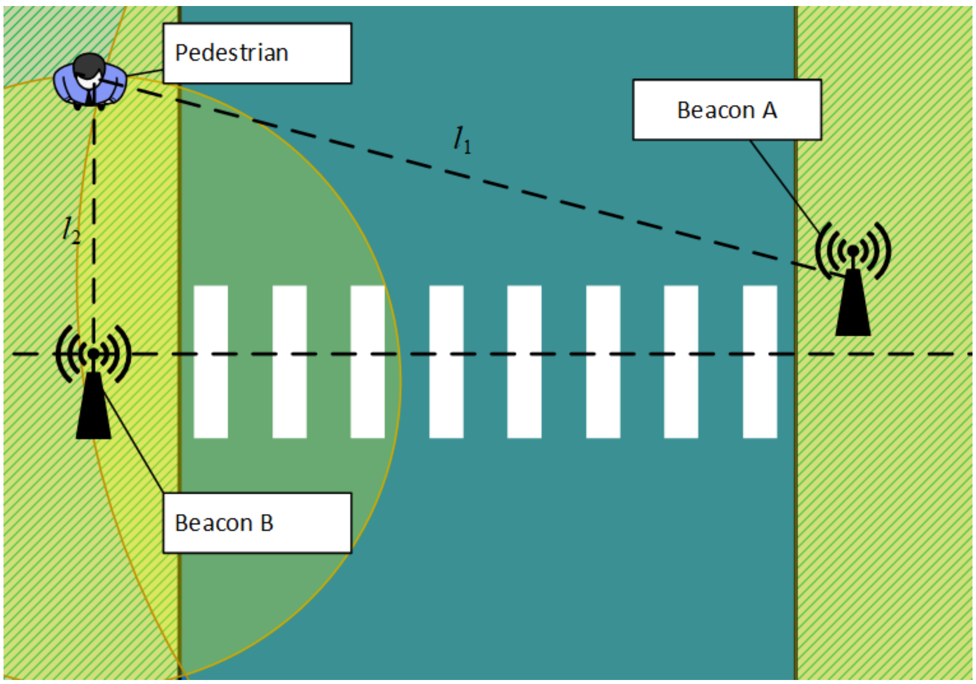

Figure 4.2: Pedestrian localization from the distances estimation to each beacon.

Besides the RSSI, each VALLPASS element in its PDU share with the application the UUID, and the major and minor fields. This will be used to narrow down the location uncertainty.

4.2 Hardware Architecture

The electronic architecture overview of the system is based on two modules responsible for keeping the system operating. As shown in Figure 4.3, the modules are divided in in the microcontroller and the power supply.

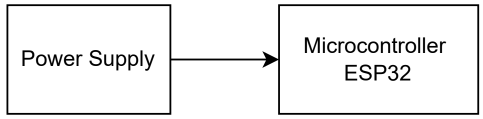

Figure 4.3: Basic electronic architecture overview.

The power supply will be provided by the VALLPASS unit.

Microcontroller ESP32

For the development of the beacon it was used the ESP32 Development Board (see Figure 4.2). Although other modules like the HM-19 that also have BLE capability have been considered, the ESP32 was selected due to its availability. The ESP32 is a System-on-Chip (SoC) created by Espressif Systems with integrated Wi-Fi and BLE.

Figure 4.4: ESP32 Development Kit Board [36].

A dual-core processor and different peripherals, such as, Digital to Analog Converter (DAC), Ultra Low Power (ULP), Pulse Width Modulation (PWM), etc., are features of the ESP32. The Wi-Fi and BLE use the 2.4 GHz band, and the device features 34 General Purpose Input/Output Interface (GPIO) pins that may be programmed with a variety of tasks. The ESP32 module was powered for testing and development using the Micro USB port, which doubles as a power source and an inbuilt USB to Universal Acrychronous Receiver/Transmitter (UART) communication converter. The ESP32 development board also comes with an antenna and a circuit for matching antennas [36].

4.3 Setting up the ESP32 as a BLE Beacon

This section shows the software development for the configuration of the ESP32 as a BLE Beacon. The development environment used to program the ESP32 Board is the Arduino IDE, the installation of the board in the software can be done by following the steps described in [37].

The source code for the configuration of the ESP32 as a BLE beacon was taken from the examples codes found in the Arduino IDE, the code can be found on [38].

The code turns the ESP32 on a BLE beacon which periodically sends out BLE advertisements to allow devices in the vicinity to detect its presence and approximate location. The beacon advertisement contains a UUID, major and minor values, and other information that can be used by devices to identify the beacon.

The code use libraries that work with the system time to configure the advertising interval and libraries to configure the BLE server.

To configure the PDU of the beacon, the major and minor values of the beacon are defined using preprocessor directives. These values can be used to distinguish between different beacons and to provide additional information about the beacon's location or purpose. Also the UUID is declared and generated using a random UUID generator website.

A constant is defined for the duration of deep sleep, which specifies how long the ESP32 will remain in sleep mode between advertisements. The duration is configured to be one second.

Two static variables are determined for storing the last boot time and boot count in Real-Time Clock (RTC) memory. These variables are used to keep track of how long the ESP32 has been running and how many times it has been booted.

Once all the variables are defined, it declares the *setBeacon()* function. This function is defined to set up the advertisement data. It creates a *BLEBeacon* object and sets its manufacturer ID, UUID, major and minor values. The advertisement data is then created using a *BLEAdvertisementData* object and the beacon data is added to it.

In the *setup()* function, the serial communication is initiated and the current system time is obtained, and the boot count and time since the last boot are printed to the serial monitor. The BLE server is initialized, and an advertising object is created. The beacon advertisement data is set up calling the previously defined *setBeacon()* function, and then the advertising is started. After a 100ms delay, the advertising is stopped and deep sleep is initiated.

The Figure 4.5 illustrates a the setting up of the ESP32 as BLE beacon. As can be seen, the system advertises continuously with a deep sleep interval of one second.

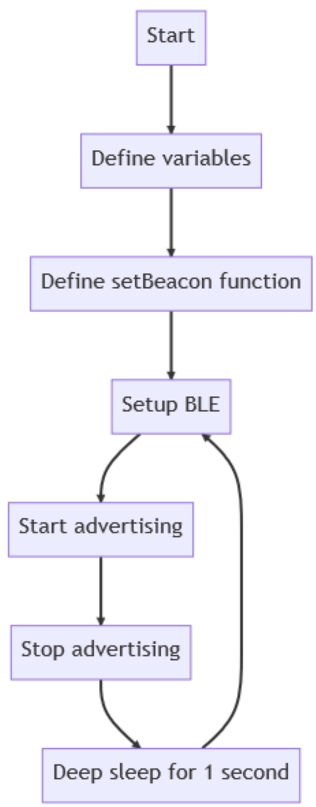

Figure 4.5: ESP32 configuration as a BLE beacon.

4.4 Android Application

This last section explores the design of the Android Application for interfacing between the user and the beacons. It was developed using the Android Studio IDE, and coded using the *Kotlin* programming language, they were selected as they are Google standards for new apps.

You can summarize its functionality as a BLE scanner that calculates the distance to the beacon. The Figure 4.6 shows the basic flowchart of how this application works.

The application consists of four main files, the *MainActivity*, *Beacon*, *ScannerFragment* and, *BeaconAdapter*.

Each file contains primordial aspects to perform the required task.

MainActivity

This code defines the main activity of the application, which is responsible for displaying the user interface and managing the application's lifecycle. This is the entry point of the application.

The *onCreate()* method is called when the activity is first created, in there the layout for the activity is set. The layout is defined in the Extensible Markup Language (XML) files of the project. Then it loads the *ScannerFragment* into the activity. The *ScannerFragment* is a fragment that is used to perform BLE scanning.

Beacon

It defines a Beacon object with properties and methods that enable the detection and handling of BLE beacons.

The main properties this class defines are the following:

- *macAddress*: a string that represents the MAC address of the beacon.
- *deviceName*: a string that represents the name of the device associated with the beacon.
- *rssi*: an integer that represents the RSSI of the beacon.
- *distance*: a double that represents the estimated distance of the beacon from the device

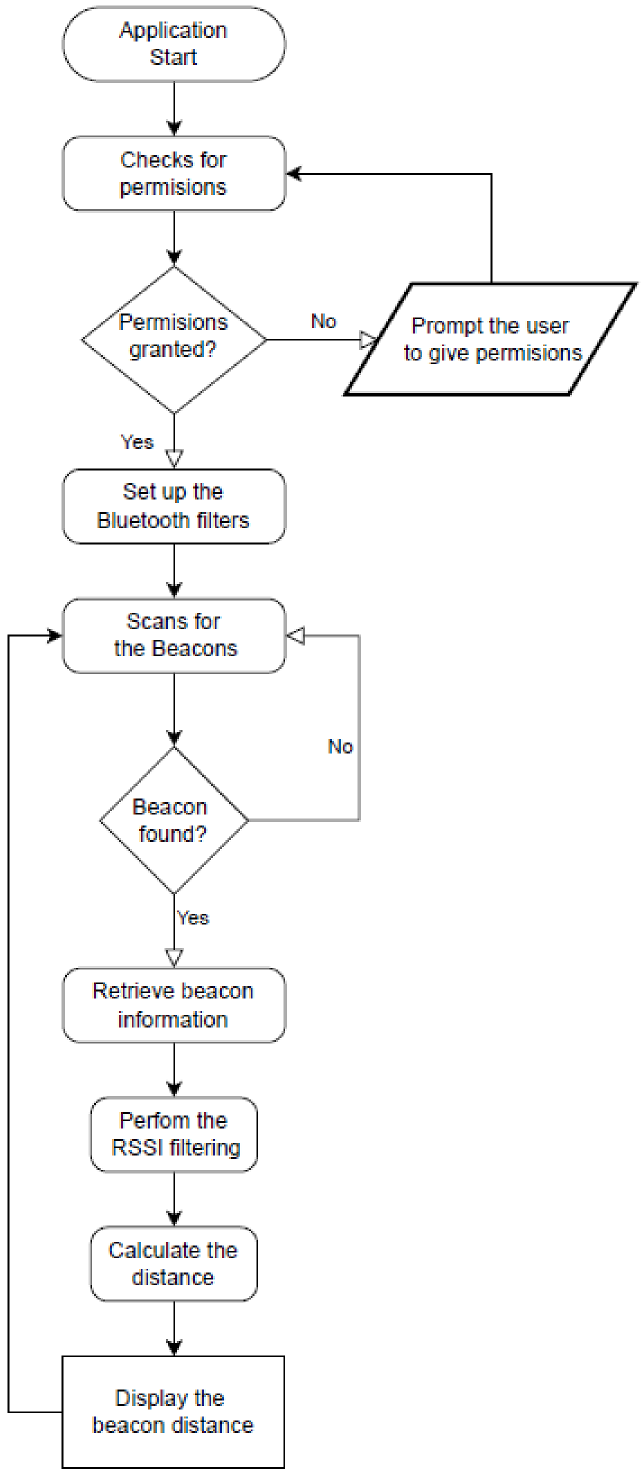

Figure 4.6: Structure of the Android APP

This class also defines a method that is used to compare two Beacon objects to determine if they are equal, this is done by checking if the MAC address of both beacons is the same. It is used to avoid to display the same beacon multiple times on the screen.

BeaconAdapter

It is a class which is used to populate a *RecyclerView* with a list of Beacon objects. The *RecyclerView* is a commonly used User Interface (UI) component in Android applications that displays a list of items that can be scrolled and interacted with.

It defines a class called *BeaconHolder* that is responsible for defining the behavior of individual items in the *RecyclerView*. This class contains references to the UI objects that are used to display the data for each Beacon object in the screen. It also defines a bind method responsible for populating the View objects with data from the *Beacon* object that is passed as a parameter.

Finally, it declares the *updateData* method, which is responsible for updating the data displayed in the *RecyclerView* by taking a new list of *Beacon* objects as a parameter.

ScannerFragment

It is responsible for setting up a Bluetooth connection and scanning for BLE beacons, it uses the Android Bluetooth Application Programming Interface (API) to scan for the devices.

The code begins by initializing the views and setting up the Bluetooth Manager. It then checks for location permissions and requests them if they are not already granted.

Then it uses the *bleFilters* method to exclusively scan for the desired beacons and ignore any other beacon that could be found around, and the *bleSettings* method to configure the scanner.

It implements the *onStart()* and *onStop()* methods to start and stop the BLE scan, respectively. It also uses a Handler to execute a text-to-speech every 10 seconds to read the distance aloud.

The *leScanCallback* object is responsible for handling the scan results. It adds the scan results to the *Beacon* object and updates the *RecyclerView* with the new data. It also calculates the moving average of the RSSI and with that value, the distance from the beacon.

The UI of this application, it contains the main information of the nearby Beacons. It can be seen in Figure 4.7.

The implementation of a text-to-speech of the distance to the crosswalk allows the interfacing with the visually impaired.

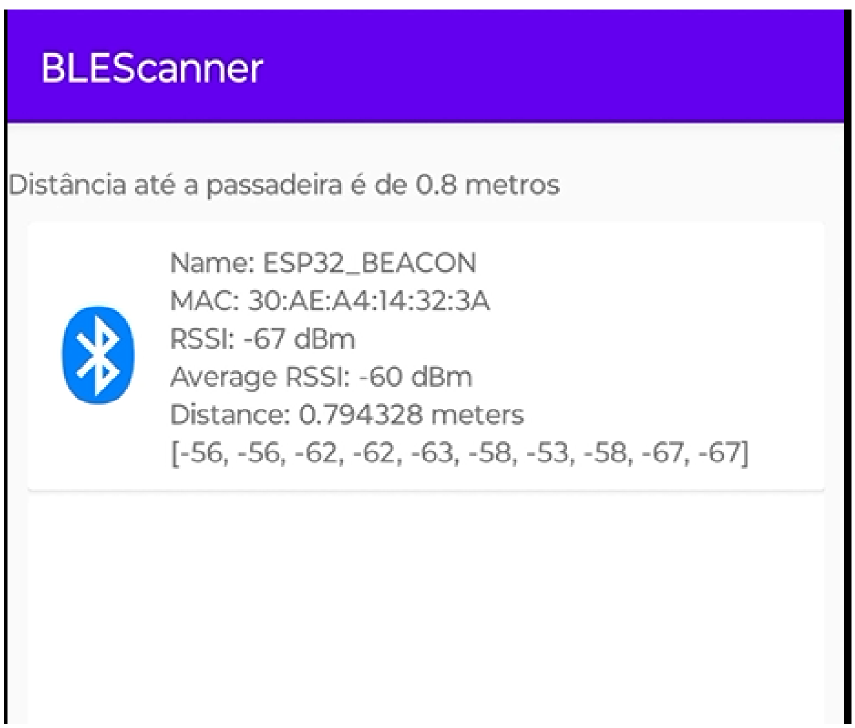

Figure 4.7: UI of the Android APP

In Appendix C there is a link to a screen capture of the system working.

Chapter 5

Results and Discussion

This chapter presents the overall system behavior. Starting with Section 5.1, it is presented the setup of the parameters needed for the application to work. Then Section 5.2 presents the experimental tests used to characterize the performance of the system.

5.1 Setting up the system parameters

First, it must be taken into consideration the parameters needed to perform the distance measurement. From (3.6) it is required to measure the value of the RSSI at one meter, and to determine the path loss exponent (n), its value is determined from Table 3.2

To be able to determine this value, the beacon is set up with a smartphone running the application at one meter of distance to take the corresponding RSSI values. Then, a total of one hundred samples were taken. Figure 5.1 presents the results.

Analyzing the gathered data, it is possible to observe the following parameters:

- Median $= -62dBm$
- Mode $= -62dBm$
- Mean $= -64dBm$

With this in consideration it was chosen that the value for the RSSI at one meter is:

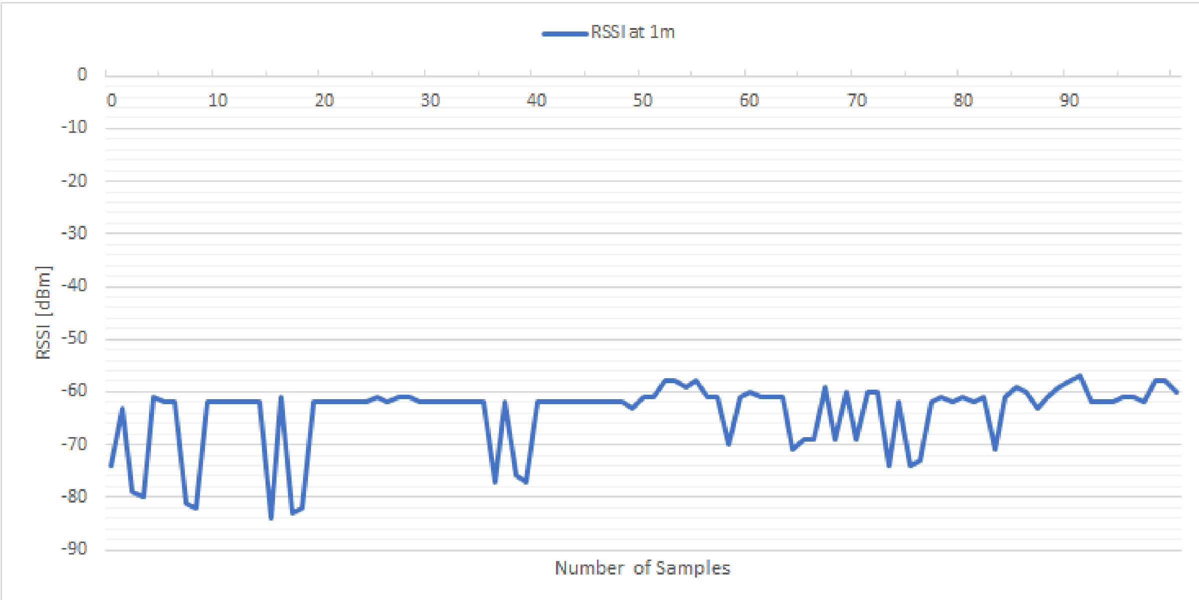

Figure 5.1: Results of the RSSI measurements at one meter.

$$A = -62dBm \tag{5.1}$$

The value of the path loss exponent can be obtained by estimation. This is done by equating the observed values over several samples to the established theoretical values. For this, a measurement taken at two meters of distance was done.

The obtained RSSI at two meters is $-71dBm$. Then, if the path loss exponent (n) is taken as the unknown in (3.6), the result is:

$$n = \frac{(A - RSSI)}{10 * \log(d)} \tag{5.2}$$

Leading to:

$$n = \frac{(-62 - (-71))}{10 * \log(2)} \approx 3 \tag{5.3}$$

This is coherent with the range of possible results that can be seen in the Table 3.2 as this trials were done in an urban area.

5.2 Experimental Testing

Once the application is configured with all its parameters, some tests on the system were performed in an urban landscape with little signal interference's (such as cars and pedestrians). For this purpose, the device was submitted to take one hundred measurements, and four tests were performed with three different distances, within ranges from 2m to 10m. For each test, the window size of the moving average filter was varied from one to ten.

Table 5.1 shows the different configurations for each test carried out. It is important to point out that these tests were performed in a controlled environment where the signal interference was not too high.

Tests	Distances (m)	Filter Size
A	2/5/10	1
B	2/5/10	3
C	2/5/10	5
D	2/5/10	10

Table 5.1: Summary of the experimental tests with different distances being measured.

5.2.1 Analysis of the Experimental Results

The performed tests will be presented in this subsection.

In Figures 5.2, 5.3, and 5.4 it is presented the measurement results with the beacon placed at 2/5/10 meters respectively from the phone. The first thing that can be noticed is the high value of variation presented in the distance. This high value is mostly influenced due to the signal noise present in the area, even though the tests were performed in a controlled environment.

One factor that it is needed to understand the behavior of the system is the coefficient of variation (CV) which is a statistical measure that represents the relative variability or dispersion of a dataset, expressed as a percentage. It is calculated as the ratio of the standard deviation to the mean of the dataset. The Figures 5.5, 5.6, and 5.7 show the

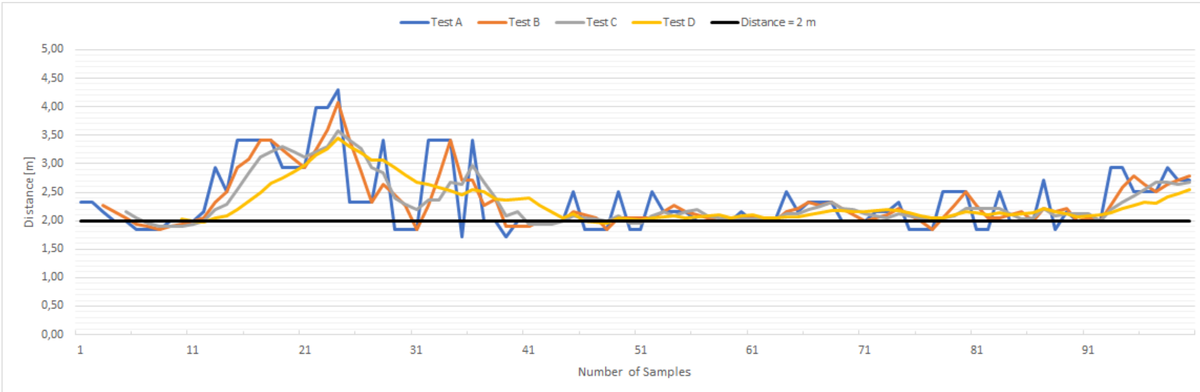

Figure 5.2: Analysis with the beacon at 2 m from the phone.

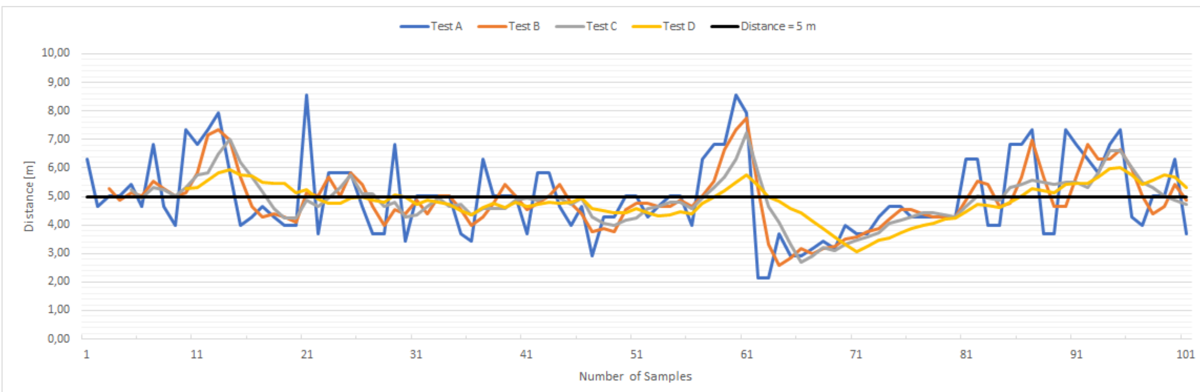

Figure 5.3: Analysis with the beacon at 5 m from the phone.

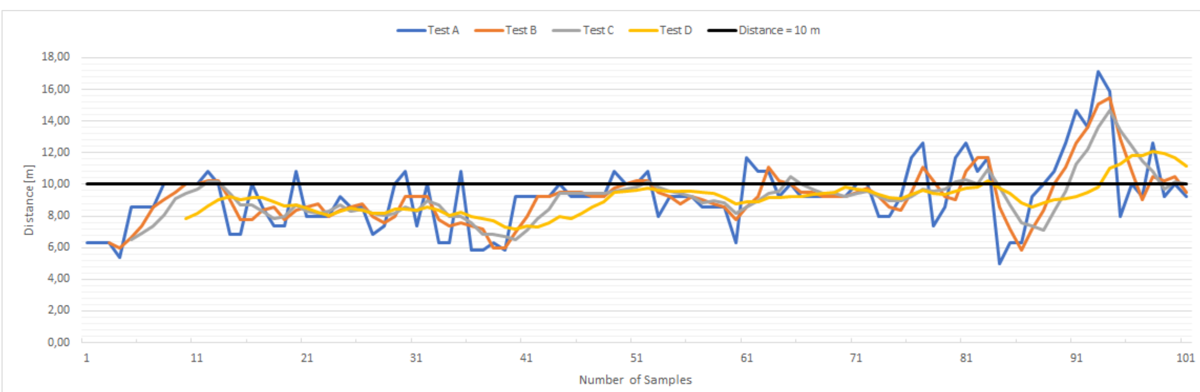

Figure 5.4: Analysis with the beacon at 10 m from the phone.

coefficients of variation for the measurement with the beacon placed at 2/5/10 meters from the phone, respectively. It is seen that the Test D shows a better performance than the others. This implies when the moving average filter has a greater window to calculate the RSSI value the values tend to be less dispersed around the mean of the dataset.

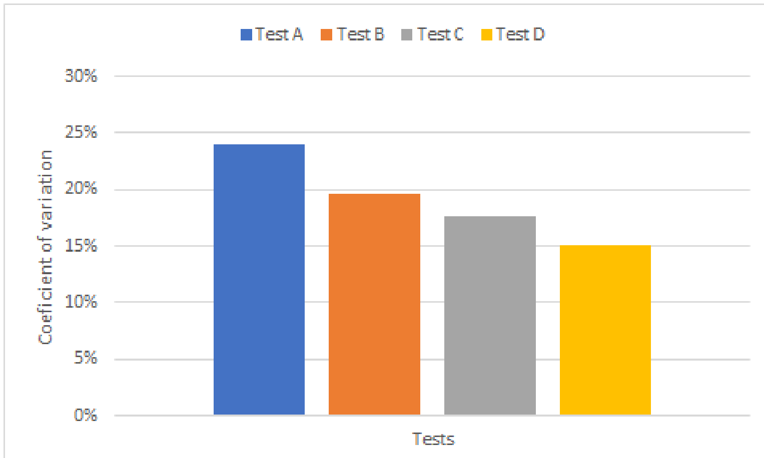

Figure 5.5: Coefficient of variation from the beacon at 2 m from the phone.

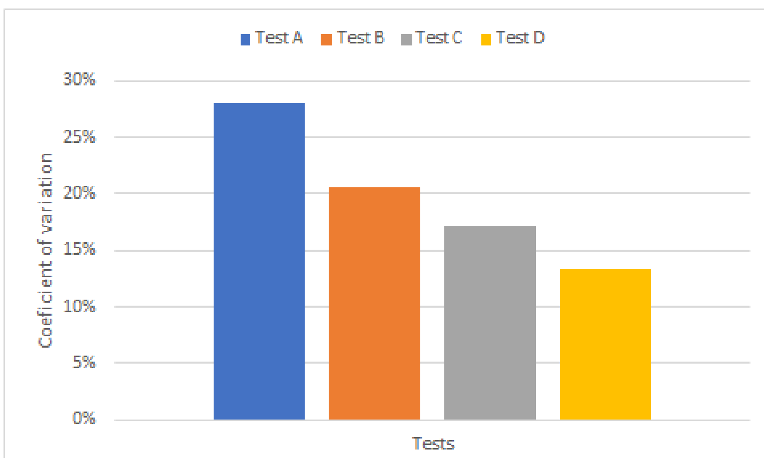

Figure 5.6: Coefficient of variation from the beacon at 5 m from the phone.

Figures 5.8, 5.9, 5.10 present the systematic error from all tests performed at 2/5/10 meters of distance, respectively. It can be noticed that test A has demonstrated highest values, in which decrease successively until test D.

From the analysis, the test C and D has demonstrated promising results, principally for the test D, in which it has shown the lowest systematic error and standard deviation. But these results can mislead the true performance of the system. It cannot be left unnoticed that these tests were performed in a condition where the distance between the

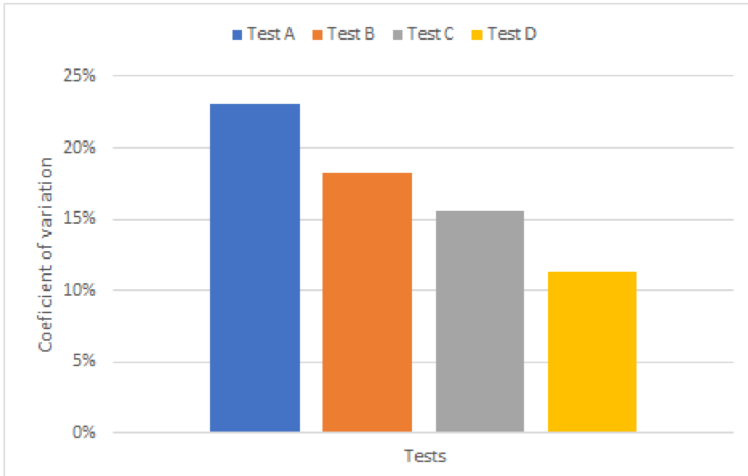

Figure 5.7: Coefficient of variation from the beacon at 10 m from the phone.

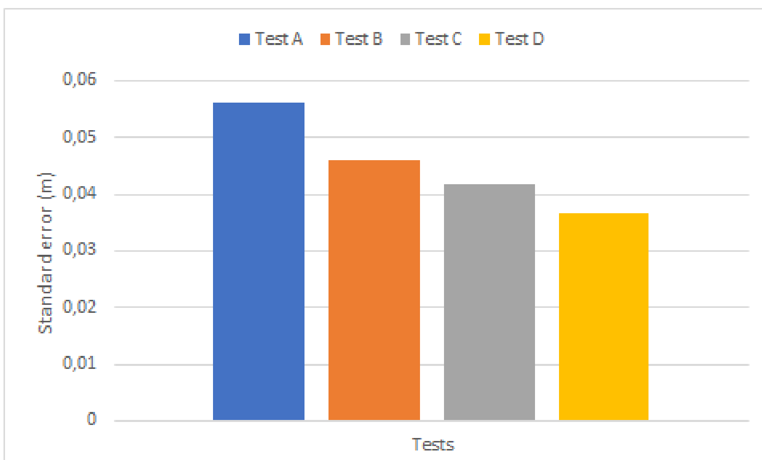

Figure 5.8: Average systematic error presented from the beacon at 2 m from the phone.

beacon and the phone was constant, as they were in a fixed position. And even in that conditions the level of noise and inconsistency of the resulting distance cannot be ignored as they are too high, and as the aim of this investigation is to increase safety for the visually impaired pedestrians.

It is worth pointing out that with a bigger window size in the moving average filter the less the resulting data will be influenced by the variation due to noise, meaning a better precision. But this comes at the cost of a slower reaction time for the system to display and communicate with the user its actual location, as this filter does not have the ability

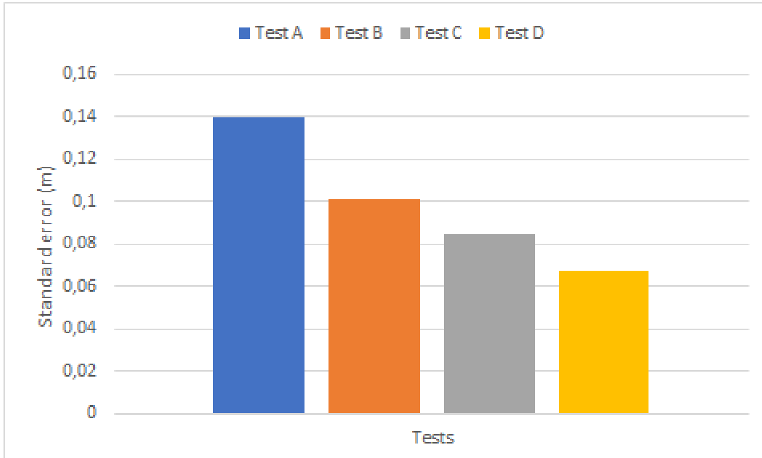

Figure 5.9: Average systematic error presented from the beacon at 5 m from the phone.

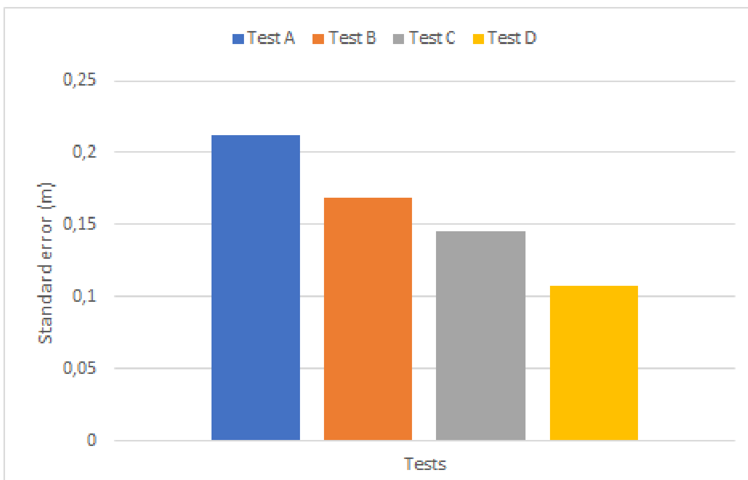

Figure 5.10: Average systematic error presented from the beacon at 10 m from the phone.

to distinguish between noise and data variations caused by the fact that the pedestrian is walking down the street.

Chapter 6

Conclusions and Future Work

The rapid urbanization and technological advancements of the last century have led to the emergence of *smart cities*, which aim to enhance sustainability, efficiency, and the quality of life for citizens. The promotion of life quality of all persons should not be a question of numbers, and it should include all people. In this context, efforts must be made to use the technology to further improve the urban mobility and accessibility of persons with disabilities. It was framed on this idea that the VALLPASS project explored a solution with the inclusion of active security measures targeting visually impaired persons embedded into a smart pedestrian crosswalk.

This approach have been done using BLE beacons as a reference point for the location of the crosswalk, and an application that exchanges information with the beacons and communicates with the user.

As has been discussed in Chapter 5, the performance of the system is not the ideal as even in a controlled environment there were signal noise that caused errors at the moment to calculate the distance between the beacon and the phone. But it is worth pointing out that even if the distance calculation where not precise, they should provide an estimation of the user location (relative to the crosswalk) and with more development able to determine when the pedestrian is near a crosswalk and then communicate relevant information to them such as traffic status and if it is safe to cross; this data can be gathered by the systems already installed into the VALLPASS units.

Future Work

After those preliminary results, it pointed out some further investigation that could be done in the future.

As it was mentioned before, the integration of the system into the VALLPASS solution is a first step to keep this work advancing. With this come the ability of the application to query the system to get information on when to cross the road and communicate with the user.

If the need to have a precise location of the pedestrian is a priority it should be explored another form of RSSI processing such as the use of a machine learning algorithm that could be able to predict the real location of the user based on trained data.

Also to increase the accessibility of the system it should be worth exploring other communication method besides the implemented text-to-speech, such as haptic approaches using the smartphone touchscreen and vibration actuator.

Bibliography

[1] H. Ritchie and M. Roser, "Urbanization," *Our World in Data*, 2018. [Online]. Available: `https://ourworldindata.org/urbanization`.

[2] ISO, *Cyber-physical systems*. [Online]. Available: `https://www.iso.org/foresight/cyber-physical-systems.html`.

[3] R. Hammons and J. Myers, "Smart cities," *IEEE Internet of Things Magazine*, vol. 2, no. 2, pp. 8–9, 2019. DOI: `10.1109/MIOT.2019.8892761`.

[4] Instituto Nacional de Estatística (Statistics Portugal), *Resident population with at least one difficulty aged 5 years or over according to the type and degree of difficulty experienced, by number of difficulties per person, age group and sex.* 2011. [Online]. Available: `https://www.ine.pt/investigadores/Quadros/Q622A.zip`.

[5] S. Gomes, J. Cardoso, C. Roque, E. Arsénio, and J. Ferreira, "Documento normativo para aplicação a arruamentos urbanos," IMT - Portuguese Institute of Mobility and Transportations, Tech. Rep., 2020.

[6] F. van Diggelen and P. Enge, "The world's first gps mooc and worldwide laboratory using smartphones," *ION GNSS+*, pp. 361–369, 2015.

[7] C.-F. Liao, "Mobile accessible pedestrian signals (maps) for people who are blind," *18th ITS World Congress, Orlando, Florida*, 2011.

[8] Z. Zhong and J. Lee, "Virtual guide dog: Next-generation pedestrian signal for the visually impaired," *Advances in Mechanical Engineering*, 2020. DOI: `10.1177/1687814019883096`.

[9] V. Murali and J. Coughlan, "Smartphone-based crosswalk detection and localization for visually impaired pedestrians," *IEEE Int Conf Multimed Expo Workshops*, 2013.

[10] B. Bhargava, P. Angin, and L. Duan, "A mobile-cloud pedestrian crossing guide for the blind,"

[11] R. Cheng, K. Wang, K. Yang, N. Long, and H. Weijian, "Crosswalk navigation for people with visual impairments on a wearable device," *Journal of Electronic Imaging*, vol. 26, p. 1, Oct. 2017. DOI: `10.1117/1.JEI.26.5.053025`.

[12] R. Cheng, K. Wang, J. Bai, and Z. Xu, "Unifying visual localization and scene recognition for people with visual impairment," *IEEE Access*, vol. 8, pp. 64 284–64 296, 2020. DOI: `10.1109/ACCESS.2020.2984718`.

[13] KrVision, *Intoer: Auxiliary glasses for people with visual impairments*, 2017. [Online]. Available: `http://www.krvision.cn/cpjs/`.

[14] H. Son and J. Weiland, "Wearable system to guide crosswalk navigation for people with visual impairment," *Frontiers in Electronics*, vol. 2, 2022, ISSN: 2673-5857. DOI: `10.3389/felec.2021.790081`. [Online]. Available: `https://www.frontiersin.org/articles/10.3389/felec.2021.790081`.

[15] M. S. Al-Akash, "Enhancing intersection safety for the blind and visually impaired (bvi) pedestrian using device-to-infrastructure communication," 2017. [Online]. Available: `https://scholarworks.wmich.edu/masters_theses/2009`.

[16] K. Shin, R. McConville, and O. Metatla, "Outdoor localization using ble rssi and accessible pedestrian signals for the visually impaired at intersections," *Sensor*, 2022. DOI: `10.3390/s22010371`.

[17] Sinowatcher, *Sinowatcher technology co.* [Online]. Available: `https://www.trafficsolution.cn/collection/pedestrian-crossing`.

[18] Euroasfalt, *Przedsiębiorstwo usługowo produkcyjne euroasfalt sp. z o.o.* [Online]. Available: `https://euroasfalt.pl/en/category-product/active-pedestrian-crossing/`.

[19] P. McDermott-Wells, “What is bluetooth?” *IEEE Potentials*, vol. 23, no. 5, pp. 33–35, 2005. DOI: 10.1109/MP.2005.1368913.

[20] B. S. Martin Woolley, “The bluetooth low energy primer,” version 1.10, 2023. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/the-bluetooth-low-energy-primer/?utm_source=internal&utm_medium=blog&utm_campaign=technical&utm_content=the-bluetooth-low-energy-primer.

[21] Ellisys, *Bluetooth beacons*. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-beacons/.

[22] *Understanding bluetooth range*. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/.

[23] P. Spachos and K. Plataniotis, “Ble beacons in the smart city: Applications, challenges, and research opportunities,” *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 14–18, 2020. DOI: 10.1109/IOTM.0001.1900073.

[24] *Intro to bluetooth generic attribute profile (gatt)*. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-gap-gatt/.

[25] *Intro to bluetooth generic access profile (gap)*. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-generic-access-profile-gap/.

[26] *Ibeacon*. [Online]. Available: https://developer.apple.com/ibeacon/.

[27] *Get started with beacons*. [Online]. Available: https://developers.google.com/nearby/notifications/get-started.

[28] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, “Ble beacons for internet of things applications: Survey, challenges, and opportunities,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 811–828, 2018. DOI: 10.1109/JIOT.2017.2788449.

[29] T. S. Rappaport, *Wireless communications - principles and practice*. Prentice Hall, 1996, ISBN: 978-0-13-375536-7.

[30] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, "A location estimation algorithm based on rssi vector similarity degree," *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371 350, 2014. DOI: 10.1155/2014/371350.

[31] S. Sadowski and P. Spachos, "Optimization of ble beacon density for rssi-based indoor localization," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6. DOI: 10.1109/ICCW.2019.8756989.

[32] *Android's kotlin-first approach*. [Online]. Available: https://developer.android.com/kotlin/first.

[33] *Application fundamentals*. [Online]. Available: https://developer.android.com/guide/components/fundamentals?hl=en.

[34] *The activity lifecycle*. [Online]. Available: https://developer.android.com/guide/components/activities/activity-lifecycle.

[35] *Introduction to activities*. [Online]. Available: https://developer.android.com/guide/components/activities/intro-activities.

[36] Espressif, *Esp32 series, version 4.2*. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[37] Espressif, *Arduino core for the esp32*. [Online]. Available: https://github.com/espressif/arduino-esp32.

[38] N. Kolban, *Ble ibeacon*. [Online]. Available: https://github.com/nkolban/esp32-snippets/blob/master/cpp_utils/tests/BLETests/Arduino/BLE_iBeacon/BLE_iBeacon.ino.

Appendix A

Publications

Paper Submitted and Accepted

Facundo M. Bustos, José Gonçalves and João Paulo Coelho, "Improving pedestrian's crosswalk accessibility through digital fencing", *2nd Symposium of Applied Science for Young Researchers: Proceedings*, pp. 131-138, 2022. http://hdl.handle.net/10198/25933.

Appendix B

Summary of the Developed Codes

This appendix provides the link for the GitHub, where the user can find the documents developed. The files are divided as follows:

It is divided in two repositories:

1. The *BLE_Scanner* repository contains all the files for the Android application. It is written in the Kotlin programming language. `https://github.com/FacuBC97/BLE_Scanner`

2. In the *ESP32_Beacon* repository, it is found the code developed using Arduino IDE in C programming language for the ESP32. `https://github.com/FacuBC97/ESP32_Beacon`

Appendix C

Screen record of the application running

This appendix provides the link for the screen capture of the application runnning:

- https://mega.nz/file/HNICDSII#E6hfWCQziNd16nUjXeGxwR7ak6Py4-oaRYiZpLUqx_k